Universidad de Alcalá

Campus Universitario
Dpto. de Teoría de la Señal y Comunicaciones
Ctra. Madrid-Barcelona, Km. 36,6
28805 Alcalá de Henares (Madrid)
Telf: +34 91 885 88 99
Fax: +34 91 885 66 99

D. SATURNINO MALDONADO BASCÓN, Catedrático de Escuela Universitaria del Área de Conocimiento de Teoría de la Señal y Comunicaciones de la Universidad de Alcalá,

CERTIFICA

Que la tesis "**Técnicas de Clasificación, Optimización y Procesado de Señal Aplicadas a Sistemas Basados en Sensores de Gases y Líquidos**", presentada por D. Francisco Javier Acevedo Rodríguez, realizada en el Departamento de Teoría de la Señal y Comunicaciones bajo mi dirección, reúne méritos suficientes para optar al grado de Doctor, por lo que puede procederse a su depósito y lectura.

Alcalá de Henares, 11 de mayo de 2009.

Fdo: Dr. D. Saturnino Maldonado Bascón

Universidad de Alcalá

Campus Universitario
Dpto. de Teoría de la Señal y Comunicaciones
Ctra. Madrid-Barcelona, Km. 36,6
28805 Alcalá de Henares (Madrid)
Telf: +34 91 885 88 99
Fax: +34 91 885 66 99

D. Francisco Javier Acevedo Rodríguez ha realizado en el Departamento de Teoría de la Señal y Comunicaciones y bajo la dirección del Doctor D. Saturnino Maldonado Bascón, la tesis doctoral titulada "**Técnicas de Clasificación, Optimización y Procesado de Señal Aplicadas a Sistemas Basados en Sensores de Gases y Líquidos**", cumpliéndose todos los requisitos para la tramitación que conduce a su posterior lectura.

Alcalá de Henares, 11 de mayo de 2009.

EL DIRECTOR DEL DEPARTAMENTO

Fdo: Dr. D. Manuel Rosa Zurera.

Universidad de Alcalá

Escuela Politécnica Superior

Tesis Doctoral

Técnicas de Clasificación, Optimización y Procesado de Señal Aplicadas a Sistemas Basados en Sensores de Gases y Líquidos.

Autor: F. Javier Acevedo Rodríguez
Director: Dr. Saturnino Maldonado Bascón

Año 2009

Agradecimientos

Me gustaría expresar mi más profundo agradecimiento a mi director de tesis, Saturnino Maldonado, por su trabajo a lo largo de estos años en los que ha sabido guiarme, sugerir ideas y hacer crítica constructiva. Pero si debo destacar algo sobre todas sus múltiples cualidades, es su lado humano que le convierte en un verdadero ejemplo para los demás. También quiero agradecer a mis compañeros de departamento, especialmente a los miembros del GRAM, sus ideas y apoyo durante la elaboración de esta tesis. No puedo dejar pasar la ocasión de agradecer a Manolo y a Paco que me animaran a volver a la Universidad y que me hayan animado a realizar este trabajo.

Para mí ha sido un honor trabajar al lado de Arantxa, Javier y Elena, del Departamento de Química Analítica de la Universidad de Alcalá. He aprendido mucho con ellos y estoy seguro que la colaboración va a seguir dando buenos frutos. Personalmente me siento orgulloso de poder participar en proyectos en los que disciplinas aparentemente alejadas de la ciencia se unen para poder dar respuesta a nuevos retos.

Esta tesis no hubiera sido posible sin el apoyo y cariño de toda mi familia, incluyendo a la de mi mujer. Estoy especialmente agradecido a mis padres y a mi hermano. Me siento realmente afortunado de tenerlos siempre a mi lado. Gracias por todo lo que habéis hecho por mí.

A Ana. No puedo reunir las palabras para agradecer lo feliz que me hace estar a tu lado. Y finalmente a ti, Elena, que te soñamos tantas veces cuando empecé esta tesis y que tu alegría me ha enseñado que no hay mejor dedicación que quererte.

Resumen

Esta tesis se encuadra en el procesado de señales y métodos de reconocimiento de patrones aplicados sobre los sistemas conocidos como nariz y lengua electrónicas. Estos sistemas surgen como técnicas analíticas que tratan de imitar los sentidos del olfato y gusto humanos mediante una matriz de sensores de gases o líquidos más una etapa de procesado y clasificación de la información obtenida. Para desarrollar este tipo de de sistemas surgió la red de excelencia europea General Olfaction and Sensing Projects on a European Level (GOSPEL) que establecía como líneas de actuación la mejora de las tecnologías de los sensores, la implementación en tiempo real de los sistemas y la mejora de las técnicas de procesado estadístico de las señales, siendo éste el punto sobre el que se centran las aportaciones de esta tesis.

En una primera parte se analiza el estado del arte, tanto de las tecnologías de los sensores como de las técnicas aplicadas a los mismos, en las que se constata la importancia que tiene la información proporcionada de forma dinámica. También se revisan los métodos en el estado del arte sobre procesado de señales aplicados sobre estos sistemas así como los métodos de clasificación. A partir de esta revisión bibliográfica surgen las necesidades de proponer nuevos métodos para la extracción de la información dinámica proporcionada por los sensores, así como establecer una comparativa entre los métodos de clasificación que se han venido utilizando. El objetivo final de realizar dicha comparativa es profundizar en los métodos que mejor resultado proporcionen para proponer mejoras adaptadas a los problemas bajo estudio.

En la parte de extracción de la información dinámica se propone la ampliación de la transformada wavelet y se adapta un método de regresión para parametrizar las señales obtenidas de los sensores y utilizar estos parámetros como información discriminante. Los métodos propuestos se han probado sobre una serie de conjuntos de datos procedentes de diferentes tecnologías de sensores buscando en todo momento que los métodos puedan ser aplicados a la mayor variedad de sensores y señales posibles.

En la parte de clasificación se propone una metodología de comparación de los diferentes algoritmos de clasificación encontrados en el estado del arte, además de la propuesta de nuevos métodos kernel que han sido aplicados con éxito a otros campos de investigación. En el marco de esta tesis se ha desarrollado un nuevo método de

aprendizaje incremental de gran utilidad para los sistemas considerados ya que facilita la obtención de un nuevo modelo de clasificación ante un ensayo nuevo cuando se está en el proceso de aprendizaje. La parte de clasificación se cierra con la propuesta de un nuevo algoritmo de selección de características que permite relacionar la información seleccionada con los principios físicos que originan la separación entre clases.

La conclusión más importante de la comparativa establecida entre clasificadores es que los métodos kernel proporcionan un grado de flexibilidad muy adecuado para trabajar con este tipo de sistemas y en especial las máquinas de vectores soporte, cuando se ajustan bien sus parámetros, aparecen como un método de clasificación que por sus características consiguen buenos niveles de tasa de acierto. A partir de esta conclusión, la última parte de la tesis se dedica a la propuesta de mejora de este tipo de clasificadores con los objetivos de mejorar la tasa de acierto, realizar la extensión a los problemas multiclase y reducir el número de operaciones necesarias para evaluar futuras muestras.

Summary

This thesis is focused on signal processing and pattern recognition methods applied to the systems known as electronic noses and electronic tongues. These kind of systems appear as analytical techniques that try to mimic the smell and taste senses by means of a matrix of gas or liquid sensors plus a stage that previoulsy preprocess and classify the obtained information. The excellence network General Olfaction and Sensing Projects on a European Level (GOSPEL) was created to develop these systems. Its main research fields are divided into the improvement in the sensor's technologies, real time implementation and the improvement in the statistical preprocessing techniques, being the last mentioned the central point of this thesis.

At a first stage, the state of the art is analyzed ,either in sensor's technologies or in the techniques applied, where is important to highlight the relevance the dynamic information has in the recent applied techniques. There is also a review of the main signal processing methods applied to these systems as well as the main classification methods are studied. From this bibliographic review, new methods are proposed to extract the dynamic information provided by the sensors and a comparative methodology between the different classification methods is established. The main target of this comparative study is to go into those methods in depth and to propose improvements adapted to the problems under study.

At a second stage the dynamic information extraction is studied. An extension of the wavelet transform is proposed and a regression algorithm is adapted to model the signals obtained from the sensors and to use those parameters as discriminating information. The proposed methods are tested on a variety of data sets obtained from different sensor technologies trying to get the proposed methods applied to the most possible number of the sensor's technologies and techniques.

At the classification stage we have proposed a new comparative methodology among the different classification algorithms found in the state of the art and new kernel classification methods, with a high level of success in other fields, are suggested. In the thesis framework, an incremental learning method is developed, being useful to the considered systems, since it makes easier obtain a new classification model incorporating a new assay in the learning process. This stage is closed with the proposal of a

block feature selection method allowing to find zones and to relate this information with the physical phenomena that produce the discriminating information.

The main conclusion of the comparative study among classifiers is that kernel methods give a high and adequate level of flexibility to work with electronic noses and electronic tongues. Especially when their parameters are adjusted, support vector machines appear as a classification method that achieve high levels of accuracy. As a result of this conclusion, at the last stage of this thesis several ideas are proposed to improve overall accuracy, to extend support vector machines to multiclass problems and to reduce the number of operations needed to evaluate future samples.

Índice general

Lista de figuras

Lista de tablas

Capítulo 1

Introducción

1.1. Sistemas de sensores de gases y líquidos

Las técnicas analíticas son imprescindibles actualmente en un gran número de disciplinas como la medicina, la industria agroalimentaria, la cosmética o la automoción. Debido a esta necesidad se ha desarrollado instrumentación muy precisa y fiable, como pueden ser las cromatografías de gases y líquidos de capa fina. Sin embargo, el equipamiento relacionado con estas técnicas es muy costoso, requiere operarios especializados y es difícil implementar la instrumentación de forma portátil para tomar medidas fuera de un entorno controlado de laboratorio. Debido a estos inconvenientes, en las dos últimas décadas ha surgido con fuerza la aparición de técnicas analíticas con un coste de producción mucho más bajo debido a los sensores y biosensores que emplean. Las respuestas obtenidas por medio de estas técnicas no son tan precisas como las tradicionales, debido a los procesos físico-químicos en los que se basan dichos sensores. Por otro lado, la mayoría de los sensores presentan derivas relacionadas con los parámetros externos, como son la temperatura ambiente, la humedad, y la vida de los sensores. Estos factores, unidos a las fuentes de ruido externo, hacen que el reconocimiento de las señales procedentes de los nuevos sistemas no sea una tarea trivial. Para compensar esta falta de exactitud se hace necesario añadir etapas de procesado y técnicas estadísticas entre las que destaca el reconocimiento de patrones. Es importante destacar que con estos nuevos sistemas se produce un cambio de filosofía respecto a las técnicas tradicionales. Mientras en estas últimas se intenta identificar y cuantificar todos los componentes de una sustancia, en las nuevas técnicas se añade una etapa de reconocimiento que busca la identificación de la muestra bajo análisis en su conjunto. Dentro de este tipo de sistemas destacan aquéllos que tratan de imitar de alguna forma los sentidos humanos del olfato y del gusto. Dichos sistemas son conocidos por la comunidad científica como nariz electrónica en el primer caso y lengua electrónica en el segundo.

El sistema de olfato humano, como se muestra en la Figura (1.1) se basa en la

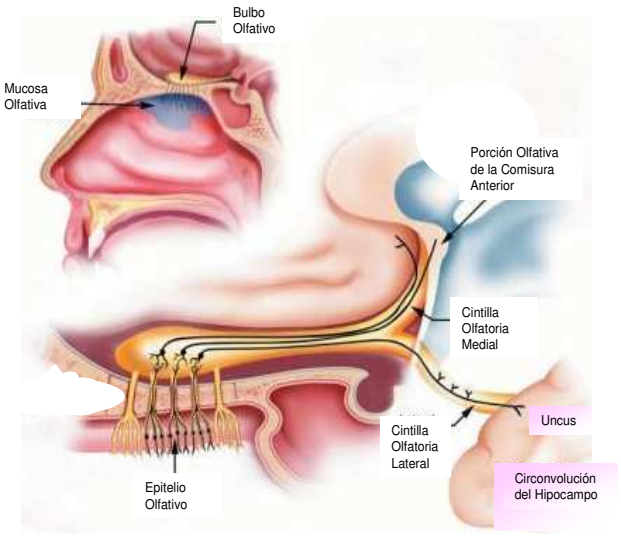


Figura 1.1: El Sistema del Olfato Humano.

recepción de componentes volátiles de alguna sustancia, que son capturadas por la mucosa olfativa. Una vez atrapadas y disueltas, estas moléculas son detectadas por los receptores específicos, denominados glomérulos, ubicados en el epitelio olfativo, situado en la parte superior de la cavidad nasal. Estos receptores son sensibles a determinados componentes volátiles y generan señales en presencia de los mismos. Dichas señales son transmitidas al bulbo olfativo donde se procesan. Desde allí se envía la información, a través de las cintillas medial y lateral, al cerebro donde se presentan las señales como un patrón, de forma que pueden ser identificadas y clasificadas. La idea de sistemas artificiales que pudieran llevar a cabo un modelo similar al sistema de olfato humano comienza en la década de los sesenta [Wilkens64], aunque no es hasta la aparición de [Persaud82] cuando se consigue discriminar entre varias sustancias. A partir de ese momento surge un gran interés en los sistemas de nariz electrónica y se establece su definición [Gardner94] como un *instrumento que consiste en una matriz de sensores con sensibilidades parciales y solapadas y un sistema adecuado de reconocimiento de patrones capaz de reconocer aromas simples o complejos*. Los paralelismos entre el sistema humano y el artificial se muestran en la Figura (1.2), donde los receptores del epiltelio tienen su equivalente en un array de sensores acondicionados para dar una señal eléctrica, la parte del bulbo olfativo se llevaría a cabo en la etapa de preprocesado y las tareas del cerebro son realizadas mediante el sistema de reconocimiento de patrones que incluye la etapa de selección/extracción de características y la parte de clasificación.

El interés de los sistemas de nariz electrónica como nuevas técnicas analíticas ha hecho que se desarrollen multitud de trabajos aplicados a diferentes disciplinas. Así, en

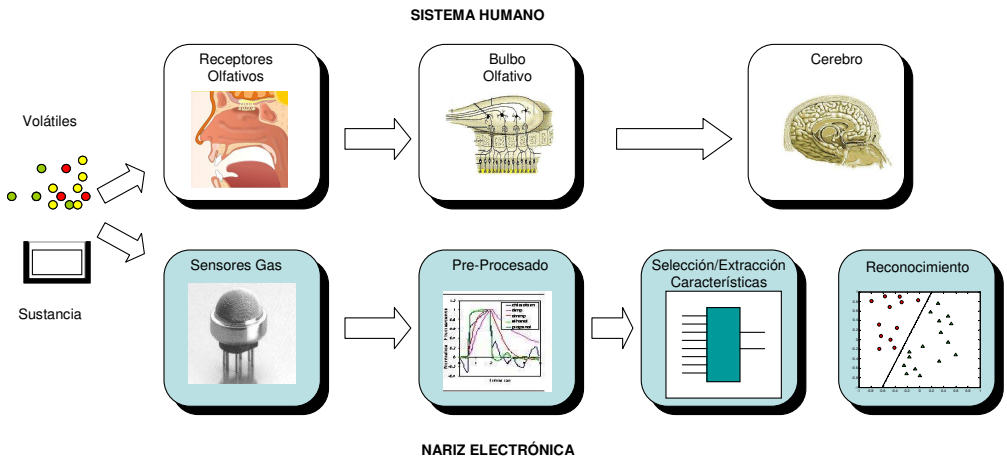


Figura 1.2: Comparativa entre el Sistema Humano y la Nariz Electrónica.

la industria alimentaria podemos destacar su aplicación para la discriminación del aroma del café [Shilbayeh04], [Gardner92], la calidad del salmón ahumado [Olafdottir05], el aceite de oliva [Garcia04], la discriminación de vinos por su aroma [Garcia06], [Lozano05] o el control del nivel de tostado de las avellanas [Correig05] por citar algunos ejemplos. También en el campo de la medicina se han realizado interesantes trabajos como los descritos en [Natale03], [Chen05] con el objetivo de detectar el cáncer de pulmón, o el publicado en [Gardner98] para detectar la presencia de la bacteria E-Coli. En el campo de la automoción también se han desarrollado interesantes aplicaciones, como la detección de gases contaminantes [Huyberechts97] o la monitorización de la calidad de los combustibles [Brudzewski06]. Recientemente, se ha fijado como gran reto el desarrollo de estos sistemas para la detección de explosivos [Gardner04]. Aunque la mayoría de las aplicaciones descritas han sido realizadas mediante narices electrónicas experimentales, existen varias compañías que han comercializado este tipo de dispositivos como la serie Fox de la compañía AlphaMOS, el modelo cyranose 320 desarrollado por Cyrano Sciences, o el dispositivo e-nose 500 de Marconi Applied Technologies. Con el objetivo de fomentar toda la investigación en este campo, unir a las empresas con las instituciones de investigación y facilitar la cooperación de los miembros de la comunidad científica dedicados al estudio de las diversas facetas de la nariz electrónica surgieron los proyectos europeos NOSE, y NOSEII. Una vez finalizados los mismos se creó la red de excelencia europea General Olfaction and Sensing Projects on a European Level (GOSPEL).

Los sistemas de lengua electrónica son mucho más recientes y aparecieron por similitud con los sistemas de nariz electrónica. La definición para un sistema de lengua electrónica se establece como [Hauptmann00] *un instrumento con sensores con especificidad limitada más un sistema de reconocimiento de patrones que puede clasificar*

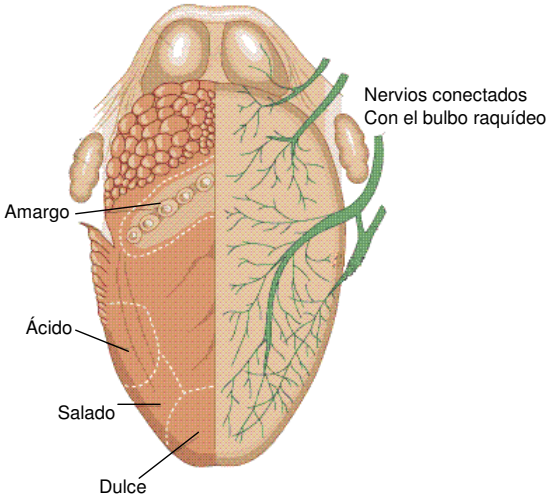


Figura 1.3: El Sistema del Gusto Humano.

muestras líquidas o extraer la información química asociada. El sentido del gusto humano se basa en la lengua como órgano donde están ubicados los receptores gustativos. Éstos se dividen en cinco tipos, capaces de identificar los sabores dulce, salado, ácido y amargo, tal como se puede ver en la figura (1.3). Recientemente se ha detectado un quinto tipo de sabor denominado unami, presente sobre todo al probar las algas marinas y basado en receptores del glutamato sódico. El descubrimiento de este tipo de sabor ha permitido entender mejor cómo funciona el sentido del gusto, donde los receptores gustativos tienen una especificidad relativa sólo a su sabor y es en el cerebro, a partir de las combinaciones de los impulsos enviados por cada tipo de receptor, donde identificamos el sabor de la sustancia más compleja, exactamente como el esquema propuesto para la lengua electrónica. Hay que hacer notar que aunque los nervios gustativos llegan al bulbo raquídeo, desde allí se transmiten hacia una zona del cerebro muy próxima al sentido del olfato, lo que hace que ambos sentidos vayan parejos. Este efecto se puede comprobar si tratamos de comer un alimento con la nariz tapada, dándonos cuenta de que no somos capaces de diferenciar el sabor de la sustancia, lo que hace que en muchas ocasiones se estudie de forma conjunta sistemas de nariz y lengua electrónica. Pese a ser más recientes los sistemas de lengua electrónica tienen unas posibilidades de aplicación tan amplias como las de la nariz electrónica. Como ejemplo de algunas aplicaciones, se han utilizado como técnicas de análisis de aguas residuales [Breijo02], análisis de medicamentos para determinar su sabor [Legin01] o determinar cómo enmascarar el sabor amargo de muchos de ellos [Takagi01]. Al igual que en el caso de la nariz electrónica, en el sector agroalimentario es donde se encuentra un mayor número de aplicaciones como el análisis de aguas minerales [Moreno06], la detección de acidez en la leche para indicar que no está fresca [Winquist00], el análisis

del zumo de naranja artificial para intentar que sea más similar al natural [Martina07] o el análisis de vinos [Parra06]. Al contrario que en el caso de la nariz electrónica, no se ha producido la comercialización de este tipo de dispositivos, al menos de una forma amplia. Esto se debe a que su implementación, como se verá más adelante, resulta más compleja que en el caso de la nariz electrónica, basándose todas las aplicaciones descritas en prototipos experimentales de laboratorio. Dada la similitud que existe con la nariz electrónica, los proyectos europeos antes mencionados y la red de excelencia GOSPEL han sido los ámbitos de referencia también para este tipo de sistemas.

1.2. Temas abiertos en la nariz y lengua electrónicas

Las posibilidades de aplicación de los sistemas de nariz y lengua electrónica hacen extraordinariamente interesante la investigación sobre la mejora de los mismos. En las redes de excelencia europeas se puede diferenciar tres ámbitos de actuación diferenciados:

Mejora de las tecnologías de sensores Es sin duda alguna el campo de mayor área de investigación. Existe una desproporción actual entre el número de receptores que tiene un sistema humano y el número de sensores de un sistema artificial, lo que hace que los esfuerzos se estén centrando en la fabricación de microsistemas electromecánicos (MEMS) y nanosistemas (NEMS) [Gardner01]. La utilización de polímeros en el caso de la nariz electrónica y de microarrays en el caso de lengua electrónica puede hacer que la integración de miles de receptores en un único chip sea posible.

Otra de las grandes líneas de actuación es la producción a gran escala de los sensores, dado que hasta ahora se han desarrollado en laboratorios de investigación pero pocos se han visto comercializados, debiendo exigir a las tecnologías de sensores que éstas sean fácilmente reproducibles de un sensor a otro. Esta línea de actuación también comprende mejoras en la estabilidad, tanto en la deriva como en la sensibilidad de los sensores y en la vida útil de los mismos, especialmente en el caso de los biosensores. Por otro lado, es cada vez más importante desarrollar sensores enfocados ca la aplicación en la que serán utilizados. Las nuevas tendencias van encaminadas hacia sensores que tengan una determinada especificidad para las componentes gaseosas o líquidas de las sustancias bajo análisis.

Aunque no es objetivo de este trabajo aportar mejoras en el campo de la tecnología de sensores, sí es necesario entender los principios de funcionamiento de los mismos.

Técnicas de preprocesado y reconocimiento de patrones Esta parte, que se conoce como la línea de actuación software, es el marco donde se encuadra esta tesis doctoral. Estas técnicas son esenciales para el desarrollo de un sistema de nariz o lengua electrónica. Frente a los sistemas iniciales, donde los sensores daban una medida estática, cada vez tiene más importancia la información proporcionada de forma dinámica relacionada con la cinética de la sustancia bajo análisis. Este cambio hace que las señales procedentes de los sensores tengan que considerarse a lo largo de periodos de tiempo, con lo que la extracción de la información de los arrays de sensores precise de técnicas de preprocesado que contemplen este tipo de medidas.

Uno de los aspectos que tienen mayor relevancia en el campo de los sistemas de nariz y lengua electrónica es la selección y la extracción de características [Distante02], [Leone05]. La información proporcionada por la selección de características hace que se pueda conocer la utilidad de la presencia de un determinado sensor o medida tomada en la etapa de preprocesado. Esta información es crucial cuando se está diseñando una nueva aplicación, ya que se comenzará por una matriz de sensores extensa, de los cuáles no todos serán útiles y solo aportaran ruido al sistema. La extracción de las características nos ayudará a evitar información correlada y dar a cada sensor o medidas asociadas la importancia que deben tener en el sistema de reconocimiento de patrones.

Las técnicas de reconocimiento de patrones completan este área de actuación. El manejo de este tipo de técnicas ha estado ligado al comienzo de los sistemas de nariz y lengua electrónica, pero cada grupo de investigación ha venido utilizando el método de reconocimiento con el que estaba más familiarizado, no estableciendo un marco comparativo entre los mismos con una cierta rigurosidad. También es importante resaltar en este punto que, dada la naturaleza de estos sistemas, el número de muestras disponibles para cada aplicación es muy reducido, ya que cada experimento es costoso de obtener. Así, frente a otros campos de aplicación de las máquinas de aprendizaje, en este tipo de sistemas nos encontramos con conjuntos de entrenamiento muy reducidos. Dada esta condición y el hecho de que los sistemas deban ser implementados en instrumentación de tiempo real se ha propuesto en recientes trabajos la utilización de máquinas de vectores soporte (SVM) [Vapnik98] cuya principal ventaja reside en la capacidad de generalización en el aprendizaje de un problema. Entre las críticas recibidas figura el hecho de que las SVM fueron diseñadas para problemas de dos clases, quedando la extensión al caso multiclase abierto. Por otro lado, al igual que prácticamente todos los métodos de aprendizaje, su rendimiento depende de la selección de

diversos parámetros a priori, lo que es conocido como el *problema de la selección del modelo.*

Por último, hay que destacar el interés que existe en el diseño de las técnicas de preprocesado y de reconocimiento de patrones inspirados en el comportamiento de los sitemas de olfato y gusto humano.

Implementación hardware Esta es la parte encargada de unir los esfuerzos de investigación anteriores e implementarlos. La investigación de esta parte tiene un foco puesto en el desarrollo de equipos portátiles para que puedan tomarse medidas de campo. Así, es necesario rediseñar todos los prototipos de laboratorio y adaptarlos para que puedan ser desarrollados como equipos autónomos. La responsabilidad de esta parte se localiza más en las empresas involucradas en las redes de excelencia.

Es importante tener en cuenta que, aunque las áreas de actuación están claramente diferenciadas, es imprescindible que todo esfuerzo investigador piense en el resto de las partes, especialmente en el área donde está centrada esta tesis doctoral. Los métodos de preprocesado y reconocimiento de patrones deben tener muy presente la tecnología de los sensores para poder obtener la información adecuada y hacer diseños realistas sobre el número de muestras a conseguir. La selección y extracción de características debe dar respuesta al diseño de arrays de sensores en una aplicación determinada. Finalmente, los métodos utilizados deben tener en cuenta la integración en sistemas de tiempo real y que puedan ser portátiles, para lo que es necesario que se minimice en la medida de lo posible el número de operaciones a ejecutar una vez que el sistema ha sido entrenado.

1.3. Objetivos del trabajo

En la sección anterior se han situado las áreas de actuación en los sistemas basados en sensores de gases y líquidos y se ha localizado el ámbito de la tesis en el preprocesado de las señales y en el reconocimiento de patrones. Los objetivos fundamentales de este trabajo son:

1. *Proponer nuevos métodos de obtención de parámetros de las señales.* Cada vez es más importante la información dinámica proporcionada por los sensores, por lo que se hace necesario disponer de métodos que puedan obtener dicha información estimando una serie de parámetros de forma automática. Estos parámetros nos servirán como entradas a nuestro sistema de reconocimiento de patrones. Para cubrir este objetivo se aplicarán métodos que sean válidos para un amplio rango de señales procedentes de sistemas de nariz y lengua electrónica.

2. *Realizar una comparación entre los distintos métodos de clasificación utilizados por los grupos de investigación de sistemas de nariz y lengua electrónica.* Con este objetivo se quiere dar respuesta a la necesidad de establecer las ventajas e inconvenientes de los diferentes métodos utilizados. Para poder lograr este objetivo se deberá hacer un estudio profundo de estos métodos, con el propósito de implementarlos y ajustarlos. Es necesario también, para cumplir este objetivo, revisar los métodos de estimación del error.

3. *Aplicar nuevos clasificadores en el estado del arte.* Además de realizar el estudio, implementación y comparación de todos los clasificadores utilizados en este tipo de sistemas, se aplicarán métodos de reconocimiento de patrones que están dando buenos resultados en otras disciplinas y que no han sido aplicados a los sistemas de nariz y lengua electrónica.

4. *Estudiar y proponer nuevos métodos de selección y extracción de características.* Para cumplir con este objetivo se deberá desarrollar una herramienta que permita seleccionar entre los diferentes métodos y también el tipo de clasificador. A partir de los resultados obtenidos se deberá entender la forma en que los distintos algoritmos consiguen seleccionar las características y se propondrán nuevos métodos.

5. *Dar respuesta a los puntos no cerrados en la clasificación usando SVM.* Como se ha mencionado, el uso de SVM parece adecuado en los tipos de sistemas en los que no disponemos de gran cantidad de datos, pero existen puntos abiertos como el problema de la selección del modelo, la extensión al caso multiclase o el planteamiento de un sistema coherente entre la selección de características y el clasificador a utilizar. Para poder cumplir este objetivo debemos entender cómo obtener medidas de generalización del error de forma rápida y que se aproximen lo más posible a la teoría de las SVM. Además, se debe optimizar, en la medida de lo posible, el tiempo de cómputo necesario para evaluar futuras muestras con este tipo de clasificadores.

6. *Abrir nuevas líneas de trabajo y aplicar a otras líneas de investigación las aportaciones propuestas.* Dado que esta tesis es la primera que se realiza en esta línea de investigación dentro del departamento de Teoría de la Señal y Comunicaciones de la Universidad de Alcalá, uno de los grandes objetivos de la misma es que sirva como punto de partida para futuros trabajos de investigación. A su vez, las técnicas estudiadas podrán ser exportadas a otras líneas de investigación.

1.4. Organización de la memoria

La memoria de esta tesis se ha estructurado en seis capítulos.

En este **primer capítulo** se ha situado el marco de la investigación con la definición de los sistemas conocidos como nariz y lengua electrónica. Se han introducido las similitudes con los sentidos del gusto y del olfato en los humanos y se han descrito algunas de las más importantes aplicaciones actuales de estos sistemas. También se han descrito las áreas de actuación abiertas, centrando nuestra investigación en el preprocesado y el reconocimiento de patrones. Una vez centrada la investigación se han planteado los objetivos a conseguir con este trabajo.

En el **segundo capítulo** se hace un estudio del estado del arte en todos los campos que serán de aplicación en este trabajo. Aunque no es objetivo del trabajo mejorar las tecnologías de los sensores, en una primera parte se describen los principios físicos en los que se basan los sensores utilizados en el campo de la nariz y lengua electrónicas. También es de gran importancia entender cómo se obtienen las respuestas dinámicas de los sensores, bien mediante sistemas de flujo variable o modulando la temperatura de trabajo o el voltaje. Además, en este capítulo se hace una revisión bibliográfica de los métodos de selección/extracción de características y de reconocimiento de patrones.

En el **tercer capítulo** se estudian métodos de procesado de señales para extraer la información dinámica de los sensores. A partir de la revisión bibliográfica se detecta la necesidad de estudiar nuevos métodos válidos para un amplio rango de señales, por lo que se realiza una extensión en el estudio de la transformada wavelet y se propone un método kernel para la parametrización de las señales obtenidas.

En el **cuarto capítulo** se realiza una comparación de los métodos de clasificación encontrados en el estado del arte, junto con un estudio de la estimación del rendimiento de los mismos. Para poder realizar esta comparación se tuvieron que implementar y ajustar los clasificadores descritos en la revisión bibliográfica, por lo que se detallan en este capítulo aquellos aspectos necesarios para su implementación. Además se proponen como nuevos métodos de clasificación el uso de las *máquinas de vectores relevantes* (RVM) y las *máquinas de vectores soporte por mínimos cuadrados* (LS-SVM). Como resultado de este capítulo se verá que, por las características de la información con que trabajamos, los métodos que mejor se adaptan son los métodos *kernel*. Además, para las LS-SVM se ha desarrollado un método de aprendizaje incremental, que resulta de gran utilidad cuando se trata de verificar cómo afecta un nuevo ensayo. También se propone un nuevo algoritmo de selección de características por bloques que proporciona información muy útil para relacionar la selección de características con los fenómenos físicos subyacentes.

En el **quinto capítulo** se hace una serie de aportaciones para optimizar las SVM aplicadas a los sistemas bajo estudio y se plantea el problema de la selección de hi-

perparámetros. Para poder resolver este problema se aborda el estudio de los métodos de estimación del error propios de las SVM, adicionales a los descritos en el capítulo anterior y se propone en este trabajo diversos algoritmos metaheurísticos para el ajuste de los hiperparámetros como son los algoritmos genéticos, simmulated annealing, particle swarm optimization o una adaptación del método de la colonia de hormigas. Posteriormente, se hace un estudio de las estrategias para extender las SVM al caso multiclase y se proponen modificaciones a métodos en el estado del arte para realizar esta extensión. Para terminar este capítulo de aportaciones sobre las SVM se plantean dos métodos para reducir el número de operaciones en la fase de test como son la reducción de vectores soporte cercanos y el establecimiento de cotas en la función de decisión.

El **sexto capítulo** se dedica a las conclusiones generales obtenidas, las aportaciones originales de esta tesis y las futuras líneas que quedan abiertas con el desarrollo de esta tesis doctoral.

Capítulo 2

Antecedentes y Estado del Arte.

Una vez situado el contexto de la investigación, en este capítulo se describe el estado del arte de los sistemas de nariz y lengua electrónica. El estudio comienza con una descripción de las tecnologías de sensores de gases y líquidos junto con las formas de obtener la información de los mismos, tanto de forma estática como dinámica. Aunque, como se ha mencionado en el capítulo anterior, no es objeto de esta tesis mejorar las tecnologías de los sensores, es muy importante entender el funcionamiento y el significado de la información que proporcionan los mismos. A continuación, se hace un estudio de las técnicas de preprocesado de las señales, considerando como tales a las técnicas destinadas a transformar y obtener los parámetros de entrada al sistema de reconocimiento, con énfasis en aquellos métodos que luego serán objeto de mejora en este trabajo. El capítulo se completa con una descripción de los diferentes métodos de reconocimiento de patrones utilizados en el campo de la nariz y la lengua electrónica, que incluye un estudio de los algoritmos de selección de características y de los sistemas de clasificación.

2.1. Tecnologías en sensores de gas.

Las propiedades deseables de un array de sensores para una nariz electrónica incluyen buena sensibilidad, alta estabilidad, respuesta rápida, que puedan ser fabricados a gran escala y un tamaño pequeño para poder ser integrados en instrumentos portátiles [Barlett92]. En esta sección se expone un repaso de aquellos principios físicos y técnicas de medida utilizados para tratar de alcanzar estos objetivos.

2.1.1. Tipos de sensores.

Existen múltiples tecnologías de sensores que proporcionan una señal en presencia de un determinado tipo o rango de gases. En esta sección solo describiremos aquellas

técnicas que se están utilizando en la actualidad en las narices electrónicas, bien en diseños experimentales o en sistemas comerciales. La clasificación de los tipos de sensores puede realizarse de distintas formas: de acuerdo a los gases que detectan, según la magnitud eléctrica que modifican o atendiendo a su principio físico. En este caso seguiremos la clasificación que se establece en [Hierlemann05], atendiendo a los principios físicos en los que se basan:

- Químico mecánicos o másicos.
- Electroquímicos.
- Ópticos.
- Térmicos.

Aunque el tipo de sensores térmicos aparece en diversas clasificaciones, en la última década no se han registrado prácticamente trabajos con este tipo de sensores. Por el contrario, los sensores ópticos aparecen como la apuesta futura en el campo de la nariz electrónica [Röck08] y en particular los microespectrómetros. Su estabilidad, selectividad y velocidad de respuesta los convierten en unos buenos candidatos, si bien hasta ahora, no han sido ampliamente utilizados en el campo de la nariz electrónica debido a su elevado coste. Los nuevos tipos de sensores ópticos están basados en capas de polímeros colorimétricos [Kenneth04] o por espectrometría de infrarrojos [Laborante07]. Sin embargo, los sensores más ampliamente utilizados en el campo de la nariz electrónica han sido los másicos y sobre todo los electroquímicos. A continuación se describen estos tipos de sensores, deteniéndonos en los de tipo MOS que han sido los utilizados en esta tesis.

2.1.1.1. Sensores másicos.

Se definen así los sensores que cambian sus propiedades de masa en presencia de gases a los cuáles son sensibles. Existen tipos experimentales como los descritos en [Adams03], pero los más extendidos son aquéllos en los que un cambio en su masa, por la presencia un de un gas, se refleja mediante un cambio en la frecuencia de oscilación del sensor.

Sensores QMB Los sensores de microbalanzas de cuarzo (QMB) son también conocidos como sensores BAW, siglas de Bulk Acoustic Wave. La sensibilidad de estos sensores se consigue utilizando polímeros no conductivos, seleccionados conforme a los gases objetivo. Consisten en capas finas de cristal de cuarzo cubiertas con electrodos de oro en los lados. Dichos electrodos son ligeramente cubiertos con

los polímeros seleccionados, por lo que en presencia de los gases objetivo se producirá un incremento en la masa de los polímeros, afectando a la frecuencia de oscilación del cristal.

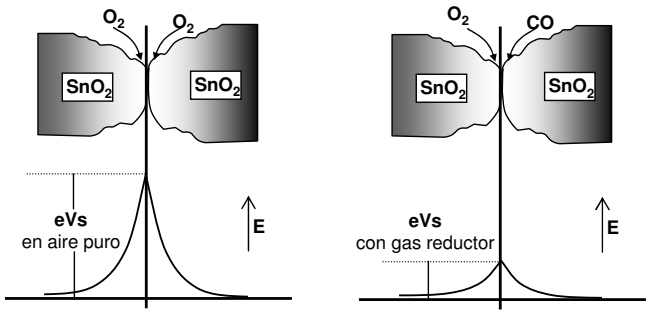
Su uso en el campo de la nariz electrónica es extenso [Saevels04], [Montag01], [Casalinuovo06] y forman uno de los módulos de sensores del prototipo MOSE-SII [Mitrovics98], uno de los prototipos de nariz electrónica más populares. Las ventajas que presentan este tipo de sensores son su estabilidad, facilidad para la fabricación a gran escala y que operan a temperatura ambiente. Las desventajas son su elevado coste, gran tamaño, son solamente sensibles para concentraciones de analito altas y el circuito de acondicionamiento es más complejo que con otro tipo de sensores, al tratarse de un cambio en frecuencia.

Sensores SAW Los sensores basados en onda acústica de superficie tienen un principio de funcionamiento muy similar a los sensores QMB, pues en presencia de los gases objetivos se produce un cambio en la masa que se registra como un cambio en frecuencia. El principio de funcionamiento consiste en realizar una guíaonda acústica, mediante polímeros no conductivos, conectada a un emisor y a un receptor. En presencia de los gases objetivo, la onda transmitida viajará por una superficie con propiedades diferentes, con lo que se registran cambios en frecuencia del receptor cuando dichos gases están presentes. Su uso en el campo de la nariz electrónica es creciente [Gan05]. Como ventajas, al igual que los sensores QMB, presentan una elevada estabilidad y fácil reproductibilidad, además de ser sensibles a concentraciones de gas analito mucho menores que en el caso de los QMB. Sin embargo presentan la desventaja de ser caros en su producción y necesitar equipamiento específico para transmitir y recibir las señales, lo que encarece todavía más el sistema.

2.1.1.2. Sensores electroquímicos

Son, sin duda alguna, los más ampliamente utilizados en el campo de la nariz electrónica. Ante la presencia de un gas al que son sensibles proporcionan un cambio en su conductividad, por lo que el circuito de acondicionamiento es mucho más sencillo que el requerido para otros tipos de sensores.

Sensores MOS La mayoría de los trabajos publicados en el campo de la nariz electrónica están basados en los sensores con base de dióxido de estaño (SnO_2) como se describe en [Gardner99], aunque experimentalmente se han probado otros semiconductores como el WO_3. El uso extensivo de los sensores tipo MOS basados en SnO_2 se debe en gran medida a que se encuentran disponibles de forma comercial. Su principio de funcionamiento puede verse en la figura (2.1). A temperaturas



(a) *En presencia de aire.* **(b)** *En presencia de gas reductor.*

Figura 2.1: Esquema de cambio de conductividad en la barrera de potencial del semiconductor.

eleveadas (200-500 ^{o}C), ante la falta de oxígeno, los electrones libres fluyen fácilmente entre las fronteras granuladas del semiconductor. En aire puro, el oxígeno atrapa electrones debido a su afinidad electrónica, quedando adsorbido en la superficie de SnO_2, con lo que se crea una barrera de potencial en las fronteras granulares. Esta barrera dificulta la libre circulación de los electrones aumentando la resistencia eléctrica de la capa activa (figura 2.1(a)). Ante una atmósfera rica en gases reductores (como pueden ser los gases combustibles), la superficie de SnO_2 adsorbe estas moléculas gaseosas provocando su oxidación (figura 2.1(b)). Este proceso disminuye la barrera de potencial, facilitando la circulación de electrones libres, lo que reduce la resistencia del sensor. Para aumentar la sensibilidad ante determinados gases, se añaden impurezas con metales catalíticos, típicamente con paladio o platino.

Para poder alcanzar las temperaturas de funcionamiento, se incorpora al substrato una resistencia calefactora o *"heater"*, unida a un *"micro-hot plate"* que hace que el calor se distribuya de forma uniforme por la capa de dioxido de estaño, tal como se representa en la figura (2.2). Es importante destacar que existen

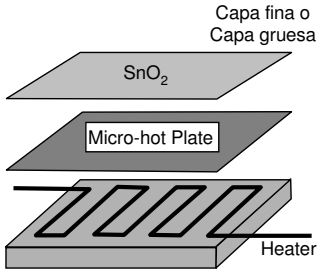
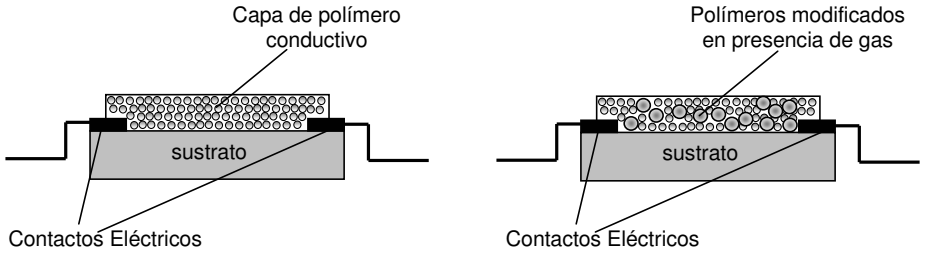


Figura 2.2: Esquema de un sensor de SnO_2.

dos tecnologías fundamentales en la fabricación de sensores de SnO_2: De capa

gruesa y capa fina. Los sensores de capa gruesa (Thick film) fueron los primeros en utilizarse en los sistemas de nariz electrónica. Tienen una mayor sensibilidad, al presentar más superficie en contacto con los gases objetivo, pero tienen como desventajas su gran consumo, tamaño y el número de gases al que responden. Por otro lado, los sensores de capa fina (Thin Film), pese a tener menos estabilidad y sensibilidad, presentan un consumo mucho menor, y dado su tamaño pueden integrarse fácilmente en arrays [Díaz-Delgado02]. En ambos casos, la sensibilidad a determinados gases es fuertemente dependiente de la temperatura de trabajo, lo que da origen a la técnica de termomodulación, como se explicará más adelante.

Polímeros Conductivos El desarrollo de sensores de gas basados en polímeros conductivos, tales como la polianilina, el polipirrol o el poliacetileno y sus derivados, está emergiendo como una de las grandes apuestas dentro de las narices electrónicas [Dai07], [Li05]. Su principio de funcionamiento se basa en depositar una capa de polímeros conductivos sobre un sustrato, al que se unen unos electrodos. En ausencia de gases objetivo, la estructura presentará una resistencia R_i, como se muestra en la figura (2.3(a)). En presencia de un gas objetivo, éste es absorbido por la estructura principal del polímero, causando una modificación de la estructura del mismo. En esta situación, la capa de polímeros cambia su composición como se muestra en la figura (2.3(b)), presentando una nueva resistencia R_n. Entre las ventajas que podemos destacar de este tipo de sensores, se encuentra la variedad de gases a detectar, pues variando la composición principal del polímero se puede hacer selectivo a diferentes gases. Otra de las grandes ventajas es que estos sensores operan a temperatura ambiente, además de su tamaño que permite fabricar arrays integrados con un gran número de sensores. Entre sus desventajas se encuentra la gran dependencia que tienen con la humedad, la estabilidad y que no se encuentren comercializados.



(a) *Sin presencia de gas.*

(b) *Los gases intereaccionan con la capa de polímeros.*

Figura 2.3: Esquema de funcionamiento de un sensor con polímeros conductivos.

En la Tabla (2.1.1.2) se muestra una comparativa tomada de [Pearce06] entre las diferentes tecnologías de sensores electroquímicos expuestos.

Propiedades	**Polímeros**	**SnO_2 (capa gruesa)**	**SnO_2 (capa fina)**
Fabricación	Crecimiento Electroquímico	Pasta	Gel
Tipo de Materiales	Amplio	Reducido	Reducido
T^a Funcionamiento	10-110 ^{o}C	250-600 ^{o}C	250-600 ^{o}C
Rango de moléculas a detectar	Amplio	Vapores Combustibles	Vapores Combustibles
Rango Detección	<20 ppm	10-1000 ppm	1-100 ppm
Tiempo Respuesta	60 s	20 s	20 s
Tamaño	<1 mm^2	1 x 3 mm	<1 mm^2
Consumo	<10 mW	800 mW	80 mW
Integrado en Array	Si	No	Si
Estabilidad	Moderado	Relativamente Pobre	Relativamente Pobre

Fuente: [Pearce06]

Tabla 2.1: Comparación entre las propiedades de los sensores electroquímicos

2.1.2. Medidas Dinámicas

En esta sección nos centraremos en los sensores electroquímicos vistos en el apartado anterior. En la aproximación clásica de la nariz electrónica se miden los cambios eléctricos, generalmente cambios en la conductividad de los sensores respecto al aire, ante una sustancia que está presente suficiente tiempo. Cada uno de los sensores que componen la matriz tendrá un valor estático, por lo que ante una determinada sustancia presentará un patrón determinado. En la figura (2.4) se puede ver la representación polar de los valores de una matriz de doce sensores ante varias muestras de dos sustancias, apareciendo dos patrones claramente diferenciados.

El problema con este tipo de medidas reside en la falta de estabilidad de los sensores, descrita en el apartado anterior, junto a la fuerte influencia de factores como el nivel de humedad o la temperatura de trabajo [Moseley91], lo que ocasiona que las medidas estáticas sean cada vez menos usadas cuando se trabaja con sensores electroquímicos. Para evitar estos defectos, en la literatura se proponen principalmente dos técnicas consistentes en tomar la información de forma dinámica, bien por variación del flujo o por variación de la temperatura de trabajo, lo que también se conoce como modulación en temperatura o termomodulación.

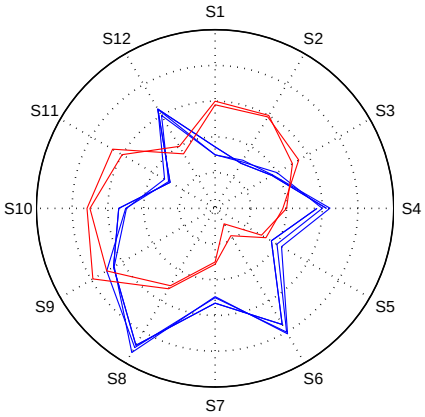


Figura 2.4: Representación polar para matriz de 12 sensores ante dos sustancias

2.1.2.1. Modulación del Flujo

Esta estrategia se basa en los procesos dinámicos que tienen lugar cuando las moléculas de la sustancia bajo estudio interactúan con el sensor. Así, se procede a pasar, de forma controlada, vapor de la sustancia bajo análisis durante un tiempo. A continuación se corta el flujo para dejar paso a un gas portador, siendo el aire puro el más utilizado. Al hacer pasar durante suficiente tiempo el gas portador, en nuestro caso aire puro, se obtendrá una medida de conductancia de referencia G_0. Al utilizar la estrategia de modulación en flujo, la señal producida por el sensor en el tiempo tiene una información valiosa sobre los procesos de absorción y desorción que tienen lugar al contacto con el sensor. Esta estrategia, puede apreciarse en la figura (2.5) donde se aprecia la respuesta de un sensor cuando se permite el paso de volátiles de la sustancia sólo durante cierto tiempo. En esta gráfica están marcados los parámetros más importantes, como son la variación de conductancia máxima alcanzada $\left(\triangle\frac{G}{G_0}\right)$, el tiempo que tarda en alcanzar este valor desde que permitimos el paso de sustancia de flujo (τ_r) y el tiempo que tarda en volver a la conductancia en aire (τ_d). En [Llobet97] podemos encontrar una descripción más detallada de esta técnica.

2.1.2.2. Modulación de Temperatura

La idea de trabajar con un mismo sensor de SnO_2 a diferentes temperaturas fue introducida en [Clifford83], haciendo notar que la selectividad de este tipo de sensores hacia determinados gases variaba dependiendo de la temperatura de trabajo. Aunque el objetivo de dicho trabajo era eliminar la interferencia con otros gases y medir únicamente CO, aumentando la selectividad del sensor, a partir de estos experimentos se

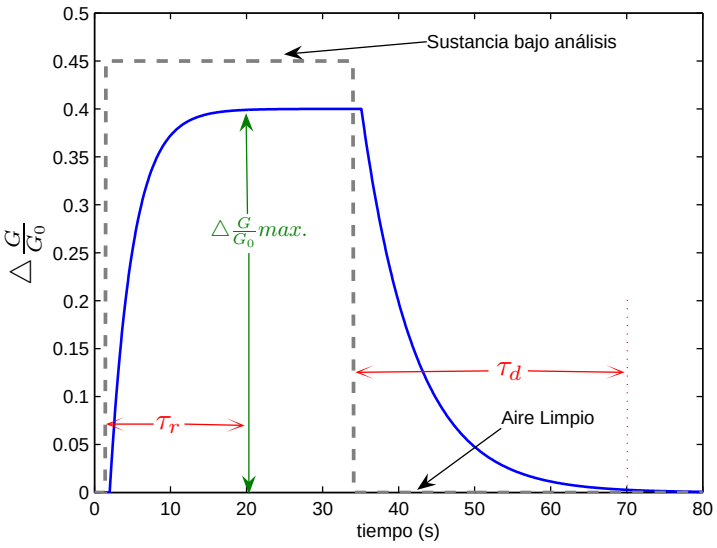


Figura 2.5: Respuesta típica de un sensor de $Sn0_2$ en modulación de flujo

pudo deducir que la variación de la conductancia con la temperatura era diferente para cada gas, como muestra la figura (2.6). Estos trabajos de investigación fueron revisados con la tecnología de fabricación micro-hot plate, de forma que controlando la tensión aplicada al heater se puede ir variando la sensibilidad del sensor a diferentes gases [Cavicchi95], [Heilig97]. De esta forma, para cada temperatura el comportamiento del sensor es diferente, lo que da origen a diferentes señales temporales ante diferentes sustancias [Vergara05]. Aparece así el concepto de pseudo-sensor, de forma que un único sensor se convierte en una matriz de pseudo-sensores, cada uno trabajando a una temperatura diferente. Al incrementar el número de sensores disponibles nos aproximamos más al sistema del olfato humano.

2.2. Tecnologías en Sensores de Líquidos

2.2.1. Tipos de Sensores

Como se comentó en el primer capítulo, los sistemas de lengua electrónica, como tales, son mucho más recientes que los sistemas de nariz electrónica. Sin embargo, existe una gran cantidad de literatura científica respecto a sensores para el análisis en líquidos, que son aplicables a las lenguas electrónicas. Podemos establecer la siguiente división de sensores para líquidos:

- Sensores Másicos.

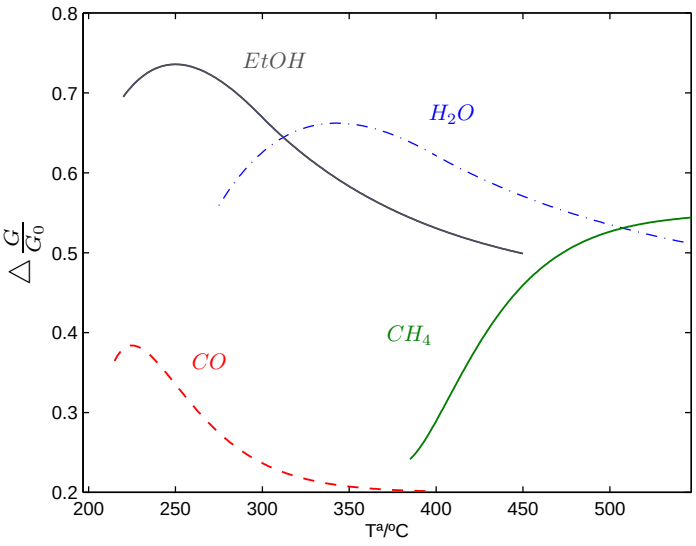


Figura 2.6: Variación de la sensibilidad con la temperatura distintos gases

- Sensores Potenciométricos.
- Sensores Amperométricos o Voltamétricos.
- Sensores Ópticos.

Los sensores másicos utilizados son los de tipo SAW, con idéntico principio de funcionamiento al descrito en el caso de la nariz electrónica, por lo que no se volverán a describir en este apartado. Los sensores potenciométricos y los amperométricos deberían ser unidos en un grupo denominado electroquímicos. Sin embargo, es tal su importancia que a menudo nos encontramos muchas clasificaciones de sensores de lengua electrónica que sólo contemplan estos dos tipos de sensores. En esta revisión nos detendremos en el desarrollo de los sensores amperométricos, por ser éstos los sensores electroquímicos sobre los que se ha investigado en esta tesis. Por último describiremos los sensores ópticos, que también han sido utilizados en esta tesis.

2.2.1.1. Sensores Potenciométricos

Los sensores potenciométricos fueron los primeros en ser utilizados en la lengua electrónica [Vlasov97]. El principio de funcionamiento es una reacción redox que tiene lugar en los electrodos de una celda electroquímica. Normalmente, se considera un electrodo activo o de trabajo y otro electrodo de referencia, en el que la reacción redox es completamente reversible, por lo que no afecta la cantidad de iones presentes en el electrolito. El electrodo de referencia suele estar compuesto por un hilo de plata

sumergido en una solución de cloruro sódico, mientras que el electrodo de referencia tiene una cápsula con una membrana porosa, que permitirá el paso del líquido de análisis. Así, aparece una fuerza electromotriz entre el sensor de trabajo y el sensor de referencia que viene determinada por la ecuación de Nernst (2.1):

$$E = E^r + K\ ln\left[\frac{\alpha_{ox}}{\alpha_{red}}\right] \tag{2.1}$$

donde E y E^r son los potenciales de los electrodos de trabajo y referencia respectivamente, K es una constante proporcional a la temperatura y α_{ox}, α_{red} son las concentraciones oxidante y reductora respectivamente. Sin embargo, este primer tipo de sensores potenciométricos respondería a cualquier posible reacción, originando mucho ruido. Por ese motivo, los electrodos de trabajo potenciométricos suelen modificarse de forma que son selectivos ante determinados iones (ISE, Ion Selectivity Electrode). Esto se consigue mediante membranas de PVC que sólo dejan pasar los iones objetivo. Así la ecuación de Nernst la podemos expresar según:

$$\triangle E = K\ ln[a] \tag{2.2}$$

siendo a la concentración de iones bajo análisis. Ejemplos de este tipo de sensores aplicados en la lengua electrónica los encontramos en [Riul03], [Gallardo04] o [Gallardo05].

2.2.1.2. Sensores Amperométricos

Este tipo de sensores también son referidos en la literatura como sensores voltamétricos, en referencia a que se puede aplicar sobre los mismos la técnica analítica denominada voltametría [Lee06]. Al igual que en el caso de los sensores potenciométricos, su principio básico se basa en una celda electroquímica, pero para este tipo de sensores suele aplicarse una configuración de tres electrodos, donde además del electrodo de trabajo y el electrodo de referencia aparece el electrodo auxiliar o *counter* destinado como retorno de la corriente producida por los efectos de reducción y oxidación que se dan en la celda cuando existe un analito objetivo [Reddy00]. Su principio de funcionamiento consiste en aplicar un potencial entre el electrodo de referencia y el de trabajo, midiendo la corriente que circula entre el electrodo de trabajo y el *counter*. Entre los sensores amperométricos, son de especial interés aquéllos en los que el electrodo de trabajo es fabricado con pasta de carbón (CPE, Carbon Paste Electrodes), debido a su bajo coste. Sin embargo, este tipo de sensores no son específicos y su respuesta es lenta. Para solucionar estos problemas se pueden crear electrodos de pasta de carbón modificados (MCPE, Modified Carbon Paste Electrodes). Entre las técnicas disponibles para lograr este objetivo, cabe destacar la inmovilización de enzimas en el electrodo, de forma que se cataliza la reacción [Gorton95]. Hay que notar que al aplicar

este tipo de técnicas pasamos al campo de los biosensores. En esta tesis se han empleado biosensores basados en oxidasas desarrollados en el departamento de Química Analítica de la Universidad de Alcalá.

2.2.1.3. Sensores Ópticos

Los sensores ópticos tienen la ventaja frente a los electroquímicos de no ser invasivos con el analito bajo estudio y sobre todo, de no sufrir el denominado efecto de envenenamiento de los sensores. Este efecto se da cuando ponemos en contacto sensores electroquímicos durante mucho tiempo con determinados líquidos, de forma que los sensores pierden su capacidad de reducción/oxidacción. La estabilidad y precisión de los sensores ópticos es superior a la de los otros tipos de sensores expuestos, aunque su problema fundamental consiste en su tamaño y en el precio [Kuswandi07]. Los sensores ópticos se pueden clasificar en tres tipos fundamentales, en función de su principio de funcionamiento [Baldini06]:

- Espectrometría
- Fluorescencia
- Quimioluminiscencia

La espectrometría es una técnica ampliamente estudiada consistente en hacer un barrido con diferentes longitudes de onda y medir la absorbancia obtenida para cada una de ellas. En aplicaciones de lengua electrónica el rango de longitudes de onda más utilizado es el ultra violeta visible (UV-VIS) (200-800 nm). Respecto a los sensores basados en fluorescencia y quimioluminiscencia, no son abordados aquí por su escasa implantación actual en el campo de la lengua electrónica.

2.2.2. Técnicas de medidas dinámicas

En el caso de la lengua electrónica, desde los primeros trabajos, se aborda su estudio mediante técnicas dinámicas. Además de obtener información sobre la cinética de las reacciones de la sustancia bajo análisis, la ventaja de obtener medidas con sistemas dinámicos es que permiten automatizar el proceso, de forma que se puede tomar una gran cantidad de muestras, lo que será muy útil para las posteriores etapas de reconocimiento de patrones.

2.2.2.1. Técnicas en Flujo

La primera técnica estudiada para los sistemas de lengua electrónica fue el análisis por inyección del flujo (FIA, *Flow Injection Analysis*) [Trojanowicz00]. El principio

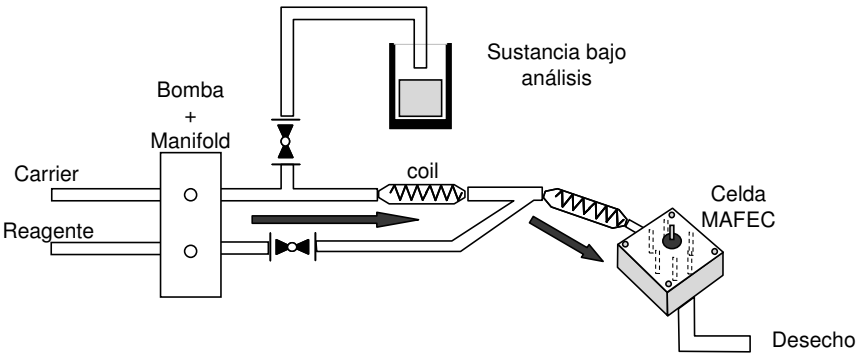


Figura 2.7: Esquema de un sistema FIA.

de esta técnica puede observarse en la figura (2.7). Los elementos necesarios son una bomba de inyección, un sistema manifold de válvulas para mezclar con la sustancia o sustancias bajo análisis y una solución tampón o carrier, que tiene la misma finalidad que el aire limpio en los sistemas de nariz electrónica. Dependiendo de la configuración del manifold se pueden considerar diferentes tipos de sistemas FIA. En este tipo de análisis es extremadamente importante controlar que el flujo que pasa por los sensores es constante para todos ellos, por lo que los sensores deben estar repartidos en una celda de flujo constante. Las medidas tomadas en esta tesis han sido realizadas utilizando una célula MAFEC (*Multianalyte Flow Electrochemical Cell*) descrita en [Maestre05]. Por el circuito del sistema circula la solución tampón, hasta que se mezcla con un bolo de solución bajo análisis, que puede ser introducida en combinación de reagentes, cuyo papel es de catalizar la reacción. En la celda de flujo están conectados los sensores, que registrarán este cambio de sustancia bajo análisis. Una vez que ha pasado por la celda de flujo se desecha la solución por el orificio de desagüe. En la figura (2.8) podemos ver la respuesta típica de un sensor de los utilizados. Los picos grandes de la figura corresponden a la introducción de una solución de control, a la que se conoce que el sensor responde con gran sensibilidad, mientras que los picos más pequeños corresponden con la inyección de la solución de análisis.

Además de las técnicas FIA, recientemente están teniendo auge las técnicas de análisis por inyección secuencial (SIA, *Sequential Injection Analysis*). La gran diferencia con la técnica FIA es que la bomba utilizada es bidireccional, de forma que se pueden preparar sustancias con diferentes tipos de concentración de forma automática. Podemos encontrar ejemplos de este tipo de sistemas utilizados con lenguas electrónicas en [Durán05a] o [Calvo07].

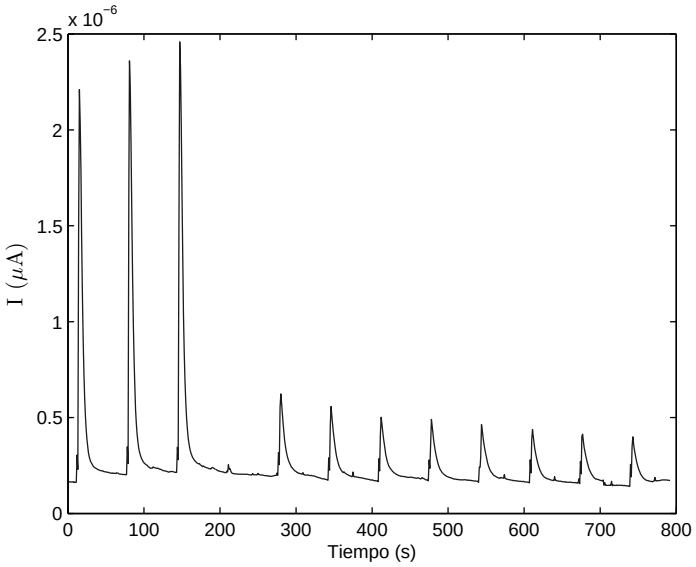


Figura 2.8: Respuesta de un sensor amperométrico de pasta de carbón en un sistema FIA.

2.2.2.2. Ciclovoltiamperometría

Los sistemas basados en flujo, proporcionan información valiosa sobre la cinética de la reacción de la sustancia bajo análisis y sirven tanto para los sensores potenciométricos como los amperométricos. Además, como se ve en la figura (2.8) se pueden introducir sustancias de control para verificar que las propiedades de los sensores se están manteniendo dentro de un rango razonable. Sin embargo, este tipo de sistemas, incluyendo los SIA, son difíciles de controlar y presentan mucho ruido en los circuitos, fundamentalmente debido a la actuación de las bombas y válvulas. En el caso de los sensores amperométricos se puede utilizar otra técnica en la que la sustancia bajo análisis no tiene por qué requerir un flujo constante. Esta técnica consiste en hacer barridos de potencial entre el electrodo de trabajo y el de referencia y estudiar la curva de respuesta de la corriente, denominándose esta técnica como ciclovoltiamperometría. Normalmente, la señal utilizada es una señal triangular de baja frecuencia, con lo que podemos representar una curva I-V como la mostrada en la figura (2.9).

Se puede observar cómo se produce un fenómeno de histéresis acusado, que dependerá de la sustancia bajo análisis. Además de este tipo de técnica, que ha sido utilizada en esta tesis, se encuentra la voltametría pulsada [Kounaves97], en la que se somete a los electrodos a pulsos de diferente polaridad, alternando entre positiva y negativa, lo que producirá reducción y oxidación. Esta técnica ha sido utilizada en el reconocimiento de vinos [Apetrei07].

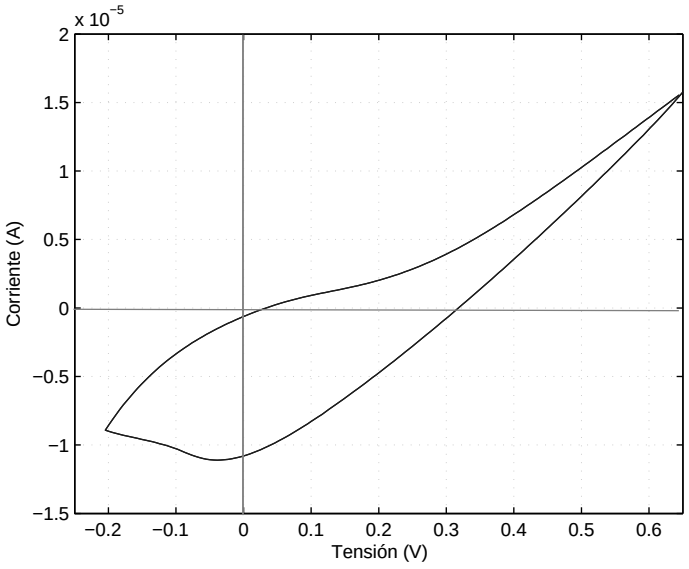


Figura 2.9: Ejemplo de ciclovoltiamperograma.

2.3. Extracción de información dinámica

Las técnicas dinámicas expuestas, tanto para sensores de gas como de líquidos, dan como resultado una señal dinámica por cada sensor. En muchas ocasiones es necesario realizar transformaciones o métodos de preprocesado de las señales de entrada antes de pasar a la etapa de reconocimiento de patrones. El objetivo de esta etapa será la de extraer la información útil de la señal, de forma que se pueda reducir el número de características en el sistema de reconocimiento de patrones, realizando el clasificador más sencillo. Hay que indicar que esta etapa es opcional dentro del sistema de reconocimiento, pudiendo trabajar con las señales directamente como vectores de entrada del sistema de reconocimiento de patrones, denominándose en este caso señales en formato RAW. Sin embargo, además de posibilitar un clasificador más sencillo, la extracción de información de esta etapa posibilita una mejor comprensión del proceso.

En esta sección solo se referirá la variable tiempo como variable independiente, pero pueden ser aplicadas a cualquier tipo de variable como la temperatura del sensor, la tensión entre electrodos en los ciclovoltiamperogramas o la longitud de onda en el caso de espectrogramas.

2.3.1. Transformada discreta wavelet

La transformada discreta wavelet (DWT) fue desarrollada como una alternativa al análisis localizado de Fourier (STFT), para poder tener precisión en el tiempo y la frecuencia. El procedimiento para extraer la información a partir de la transformada

wavelet es realizar la transformación de los datos de entrada para seleccionar aquellos coeficientes que tengan la máxima información, de forma que dichos coeficientes serán las componentes del vector que se formará para el sistema de reconocimiento de patrones.

Su implementación mediante bancos de filtros queda descrita en [Mallat99], donde se describe esta transformada como un método multi-resolución. Esta descomposición se muestra en la figura (2.10), donde se aprecia que la señal $x[n]$ se descompone en dos señales, mediante los filtros de análisis $h_a[n]$ y $g_a[n]$ que serán paso bajo y paso alto respectivamente. Las salidas de los filtros se diezman convirtiéndose en las señales $c_1[n]$ y $d_1[n]$ respectivamente. En el campo de la compresión se conoce a la señal $c_1[n]$ como coeficientes de aproximación y detalles de la señal. El esquema se puede iterar descomponiendo nuevamente la señal $c_1[n]$ incrementando la profundidad de la descomposición, pudiendo realizar la descomposición varias veces.

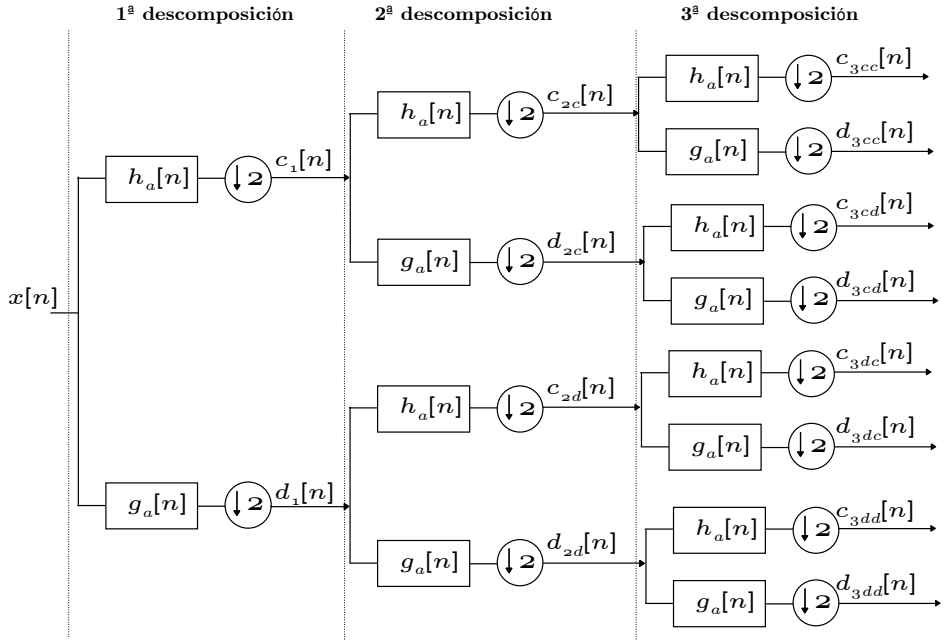


Figura 2.10: Esquema de descomposición wavelet de 2 niveles.

A partir de los coeficientes de aproximación y detalles de un nivel se pueden recuperar los coeficientes de aproximación de un nivel superior utilizando los filtros de síntesis, $h_s[n]$ y $g_s[n]$, previa interpolación de los coeficientes. La reconstrucción de un nivel de descomposición puede verse en la figura (2.11). Con este esquema, es inmediato darse cuenta de que la transformada wavelet es un esquema de descomposición sin pérdidas.

En la figura (2.12) la señal obtenida por un sensor se ha descompuesto en dos señales, cuya longitud es la mitad de la señal original, con las componentes de baja frecuencia en el primer caso y de alta frecuencia en el segundo. En la figura (2.12(b))

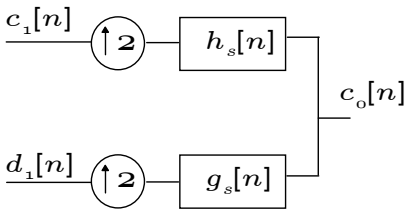


Figura 2.11: Esquema de reconstrucción wavelet.

se muestra la señal recuperada anulando la parte de alta frecuencia o detalles de la señal. Podemos apreciar cómo el resultado es una señal paso bajo de la señal del sensor original que contiene gran parte de la información de la misma. En la figura (2.12(c)) se ha realizado un proceso similar, pero anulando en este caso la señal paso bajo o aproximación de la señal para recuperar los detalles de la misma. La figura (2.12(d)) muestra la recuperación de la señal obtenida a partir de los coeficientes de la segunda aproximación, eliminando los detalles tanto de la segunda como de la primera descomposición. En esta figura se puede apreciar cómo sigue conservándose gran parte de la información de la señal original.

Este tipo de transformación ha sido aplicado en el campo de la nariz y la lengua electrónica en los trabajos descritos en [Distante02], [Phaisangittisagul07], [Ionescu02], [Llobet02] o [Al-Khalifa03]. En todos los casos se aplican filtros ortogonales de la familia Daubechies [Daubechies92].

2.3.2. Aproximación por exponenciales. Padé-Z y Ridge Regression.

Esta aproximación, consiste en estimar las señales mediante el cálculo de coeficientes de diferentes curvas que pueden aproximarse a la señal dinámica de un sensor. En el caso de las señales moduladas en concentración, la señal de la figura (2.5) puede aproximarse por una expresión matemática del tipo:

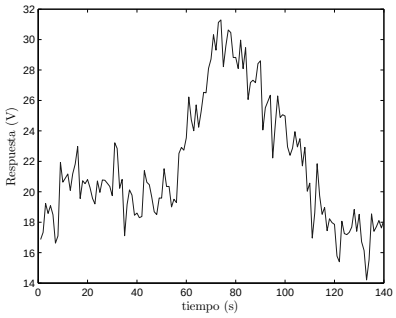
$$x(t) = E_0\left(1 - \exp^{-\frac{t}{\tau}}\right) \tag{2.3}$$

siendo E_0 y τ los parámetros a ajustar mediante cualquier método de optimización con las restricciones $E_0, \tau \geq 0$. En [Eklov97] se propone aproximar la señal mediante cinco parámetros, de forma que dicha señal está parametrizada por:

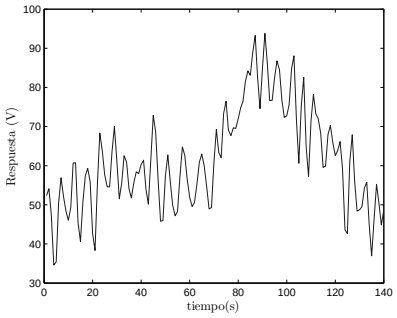
$$x(t) = E_0 + E_1\left(1 - \exp^{-\frac{t}{\tau_1}}\right) + E_2\left(1 - \exp^{-\frac{t}{\tau_2}}\right) \tag{2.4}$$

Este trabajo se puede extender a una forma más general mediante:

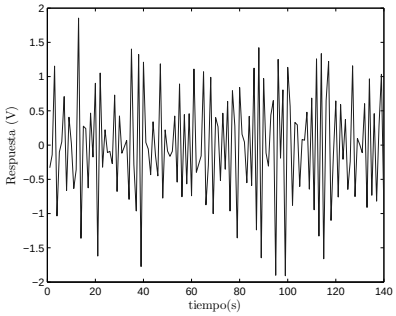
$$x(t) = \sum_{i=1}^{M} E_m \exp^{-\frac{t}{\tau_m}} \tag{2.5}$$



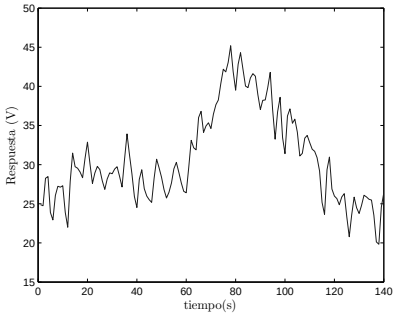
(a) *Señal original*



(b) *Aproximación de la señal tras pasar por filtro wavelet paso bajo.*



(c) *Detalles de la señal tras pasar por filtro paso alto.*



(d) *Aproximación de segundo nivel de la señal.*

Figura 2.12: Aplicación de la transformada wavelet con varios niveles de profundidad.

Sin embargo, en este trabajo el autor describe cómo, aunque esta segunda aproximación parece ajustarse mejor en la mayoría de los casos a las señales descritas por los sensores, el ajuste de curvas por éste proceso es un problema no exacto, ya que las exponenciales reales no forman una base ortogonal, lo que significa que en un ajuste por mínimos cuadrados no dará una solución única a los coeficientes.

2.3.2.1. Padé-Z

Profundizando más en el ajuste de M exponenciales y el problema de la optimización de los parámetros de las exponenciales, en [Gutierrez-Osuna99a] se propone resolver este problema mediante las técnicas conocidas como Padé-Laplace y Padé-Z. En este último caso, se trata de aproximar la transformada Z de la secuencia discreta $x[n]$ con N muestras. Discretizando la ecuación (2.5), una vez fijado el número de exponenciales M que componen la señal, se puede calcular su transformada como:

$$X(z) = \sum_{i=1}^{M} E_i \frac{z}{z - \exp^{-\frac{T}{\tau_m}}} \tag{2.6}$$

Es posible obtener una aproximación $\tilde{X}(z)$ mediante el desarrollo en serie de Taylor de orden k:

$$\begin{aligned} \tilde{X}(z) &= \sum_{n=0}^{k} \tfrac{1}{n!} X^{(n}(z)|_{z=z_0} (z - z_0)^n \\ X^{(n}(z) &= \sum_{i=0}^{N-1} x[i] (-1)^n \tfrac{(i+n-1)!}{(i-1)!} z^{-i-n} \end{aligned} \tag{2.7}$$

Para poder resolver de una forma sencilla los coeficientes de la ecuación (2.6), se obtienen los aproximadores Padé $[M/M]$ de la ecuación (2.7) y se resuelve el sistema de ecuaciones. Sin embargo, el autor de este trabajo describe cómo es importante ajustar el valor de z_0 para obtener unos resultados adecuados.

2.3.2.2. Ridge Regression

En [Gutierrez-Osuna03] se introduce el concepto de ridge regression (RR) tomado del campo del reconocimiento de patrones para determinar los parámetros E_m de la ecuación (2.5). La idea es fijar un número M de exponenciales y sus constantes de tiempo τ_m asociadas. En el caso del mencionado trabajo se eligió una escala logarítmica $\boldsymbol{\tau} = [0.01, 0.02, .., 0.9, 1, 2, .,9]$. Una vez fijadas las constantes de tiempo, podemos encontrar la solución a los parámetros E_m como

$$\begin{pmatrix} e^{-\frac{1}{\tau_1}} & e^{-\frac{1}{\tau_2}} & \cdots & e^{-\frac{1}{\tau_m}} \\ e^{-\frac{2}{\tau_1}} & e^{-\frac{2}{\tau_2}} & \cdots & e^{-\frac{2}{\tau_m}} \\ \vdots & & \ddots & \vdots \\ e^{-\frac{N}{\tau_1}} & e^{-\frac{N}{\tau_2}} & \cdots & e^{-\frac{N}{\tau_m}} \end{pmatrix} \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_m \end{pmatrix} = \begin{pmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{pmatrix} \tag{2.8}$$

En forma matricial se puede expresar como $\boldsymbol{AE} = \boldsymbol{X}$, siendo la matriz $\boldsymbol{A}$ la matriz con las combinaciones de exponenciales. La forma inmediata de obtener los coeficientes es despejar la matriz $\boldsymbol{E}$, para lo que hay que calcular la pseudo-inversa de $\boldsymbol{A}$:

$$\boldsymbol{E} = \left(\boldsymbol{A}\boldsymbol{A}^T\right)\boldsymbol{A}^T\boldsymbol{X} = \boldsymbol{A}^{\dagger}\boldsymbol{X} \tag{2.9}$$

Sin embargo, al existir multicolinealidad en la matriz $\boldsymbol{A}$ la solución de la anterior ecuación no dará resultados correctos. El método de regresión conocido como ridge regression establece un parámetro ρ de regularización:

$$\boldsymbol{E} = \left((1-\rho) + \rho\frac{tr\left(\boldsymbol{A}\boldsymbol{A}^T\right)}{M}\right)\boldsymbol{A}^T\boldsymbol{X} \tag{2.10}$$

siendo $tr\left(\boldsymbol{A}\boldsymbol{A}^T\right)$ la traza de la matriz $\boldsymbol{A}\boldsymbol{A}^T$. La ventaja de este método es que se puede cambiar la matriz $\boldsymbol{A}$ para poder adaptarla a nuevas señales.

2.3.3. Modelos ARMA

Los modelos ARMA (*Auto Regressive Moving Average*) están muy extendidos en el campo de estimación de series en el tiempo, fundamentalmente en el campo de la economía. Estos métodos tratan de realizar una estimación lineal de la señal mediante:

$$ARMA(p,q): \quad x[n] = \sum_{i=1}^{p} \alpha_i x[n-1] + \sum_{i=1}^{q} \beta_i e[n-1] \tag{2.11}$$

Fijando p o número de polos del sistema y q como número de ceros, encontramos los parámetros que caracterizarán la señal. Su aplicación en los sensores de gas la podemos encontrar en [Nakamura94], [Llobet06].

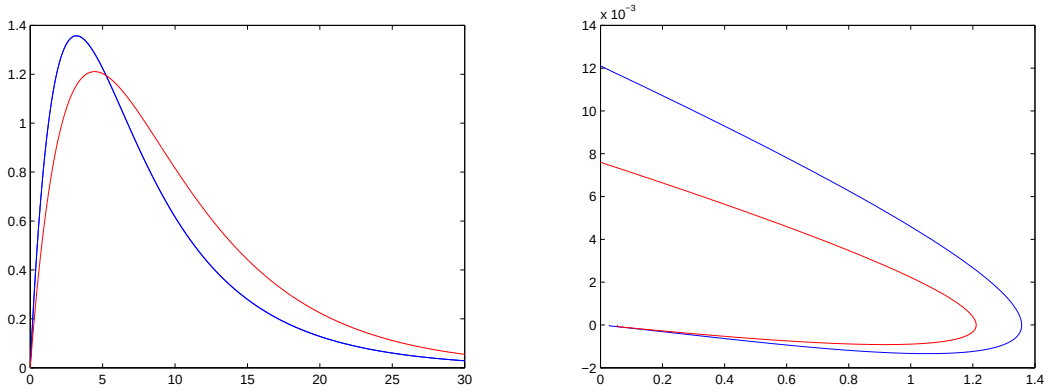
2.3.4. El espacio de fase

Tomando conceptos de los sistemas dinámicos, en [Martinelli04] se introdujo la extracción de parámetros mediante el espacio de fase. Dada la señal discreta $x[n]$, las coordenadas del nuevo espacio son $[x[n], x[n-n_0], ..., x[n-(k+1)n_0]]$ formando una trayectoria que identifica su señal. Tomando solo dos términos de la trayectoria y $n_0 = 1$ tendremos que en el nuevo espacio los puntos del espacio de fase son representados por la señal y su derivada en cada momento, como se puede apreciar en la figura (2.13) para el caso de un sensor de gas modulado en concentración. En el trabajo referido, se proponían varias medidas geométricas como parámetros para identificar la forma de la curva, mientras que en [Vergara07] se propone el uso de cinco momentos dinámicos para identificar la señal a partir del espacio de fase.

2.3.5. Window time scale

Este método fue introducido en [Gutierrez-Osuna99b] y posteriormente fue aplicado en [Gutierrez-Osuna03]. Se basa en la misma idea de representar la señal de entrada mediante una serie de funciones *kernel* desplazadas y escaladas en el tiempo. La señal puede ser descompuesta como suma de N_s funciones iguales, siendo en estos trabajos el kernel utilizado:

$$\kappa_i[n] = \frac{1}{1 + \left(\frac{|n-c_i|}{a_i}\right)^{2b_i}} \tag{2.12}$$



(a) *Señales en el tiempo.* **(b)** *Espacio de fase para ambas señales.*

Figura 2.13: Ejemplo sobre el espacio de fase.

donde los parámetros a determinar de cada kernel son $[a, b, c]$. Tanto el autor como la referencia citada no explican cómo obtener dichos coeficientes, dejando entrever que se ajustan de forma determinista, observando las señales obtenidas.

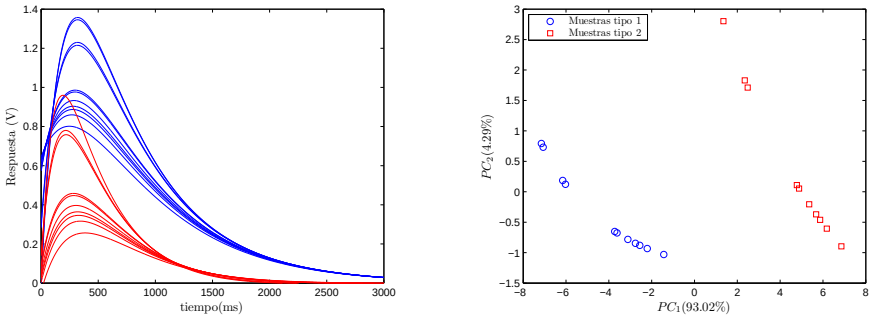
2.3.6. Análisis de componentes principales (PCA)

El análisis por componentes principales (PCA), también denominado como transformada discreta de Karhunen-Loève en el campo de las telecomunicaciones, es probablemente la herramienta más utilizada en procesos de quimiometría [Wise06]. Considerando una señal dinámica procedente de un sensor como un vector columna de m muestras, formaremos la matriz $\boldsymbol{X}$ con n señales, obtenidas cada una de ellas en distintos experimentos. Para nuestros propósitos consideraremos que $n \geq m$ y sin pérdida de generalidad consideraremos que la media de las filas de la matriz $\boldsymbol{X}$ es nula. El análisis de componentes principales se basa en la transformación lineal:

$$\boldsymbol{Y} = \boldsymbol{W}^T \boldsymbol{X} \tag{2.13}$$

donde $\boldsymbol{W}$ es una matriz de autovectores de $\sum_x$, definiendo como tal a la matriz de covarianza de $\boldsymbol{X}$, ordenados según el valor de sus autovalores asociados. Al hacer esta transformación, estamos proyectando los vectores de $\boldsymbol{X}$ hacia las direcciones de máxima varianza. La transformación descrita en la ecuación (2.13) es bilineal, pudiendo expresar también:

$$\boldsymbol{X} = \boldsymbol{W}\boldsymbol{Y} \tag{2.14}$$



(a) *Señales originales.*

(b) *Representación de las dos primeras componentes principales para los dos tipos de señal.*

Figura 2.14: PCA plot para diferentes señales.

Generalmente, se trunca el número de vectores de la matriz $\boldsymbol{W}$, quedándonos con los k primeros vectores de ésta asociados con los k autovalores mayores que explican las principales direcciones de variación. Al proyectar de esta forma la matriz $\boldsymbol{X}$, la matriz transformada $\boldsymbol{Y}$ tendrá dimensiones $[k \times n]$, con lo que se hace una reducción del número de características respecto a la matriz $\boldsymbol{X}$, pues $k \leq m$. A su vez, podemos reconstruir la señal a partir de las proyecciones de los k autovectores mediante:

$$\boldsymbol{X}^{rec} = \boldsymbol{W}_k^T \boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E} \tag{2.15}$$

donde $\boldsymbol{W}_k^T$ es la matriz de los k vectores asociados a los autovalores de mayor valor y $\boldsymbol{E}$ es la matriz de error. Entre las posibles transformaciones lineales hacia un espacio de menor dimensión, el método PCA asegura un error cuadrático medio mínimo.

En quimiometría, se suele denominar a la matriz $\boldsymbol{W}$ como *matriz de loadings*, representada por $\boldsymbol{P}$, mientras que al vector proyección $\boldsymbol{y}_i$ de un vector $\boldsymbol{x}_i$ se suele denominar como *score*. Es muy común encontrarse con el denominado *PCA plot*, que representa las proyecciones del conjunto de datos del primer autovector o primera componente principal frente a la segunda componente, como se muestra en la figura (2.14), donde las señales de la figura (2.14(a)) se someten a una transformación con 2 componentes principales. En la figura (2.14(b)) se muestra el *PCA plot* y la cantidad de información que contiene cada autovector, calculada a partir de sus autovalores λ_i mediante:

$$I_i\,(\,\%) = \frac{\lambda_i}{\sum_{j=1}^{m} \lambda_j} \cdot 100 \tag{2.16}$$

Si bien el análisis PCA hace una transformación con un error cuadrático mínimo, es preciso constatar que no tiene en cuenta las clases de las muestras que transforma, por lo que diversos autores lo han considerado como un método de clasificación no supervisado.

Es importante destacar dos parámetros estadísticos del análisis PCA, pues algún método de clasificación que se verá más adelante se basa en ellos. Por un lado el estadístico Q_i es una medida de la suma del error cuadrático entre el vector $\boldsymbol{x_i}$ y su versión reconstruida $\boldsymbol{x_i}^{rec}$. Por tanto se calcula como:

$$Q_i = \boldsymbol{e}_i^T \boldsymbol{e}_i \tag{2.17}$$

donde $\boldsymbol{e}_i$ es el vector i de la matriz $\boldsymbol{E}$. Por otro lado, el estadístico Hoeltelling's T_i^2 se calcula como:

$$T_i^2 = \boldsymbol{y}_i^T \boldsymbol{\Lambda}_k^{-1} \boldsymbol{y}_i \tag{2.18}$$

donde $\boldsymbol{\Lambda}_k$ es una matriz diagonal que contiene los autovalores $\lambda_i, i = 1, 2, .., k$. El estadístico Hoetellings T_i^2 es una medida de la distancia entre la muestra proyectada y el origen de coordenadas del nuevo subespacio donde se proyectan las muestras.

2.3.7. Análisis discriminante lineal (LDA)

El análisis discriminante lineal (LDA), al igual que en el caso de PCA, se considera en muchos casos como un método de clasificación en sí mismo. En este caso se verá su utilidad como método para conseguir una extracción de características. El método, al igual que PCA, se basa en una transformación lineal como la descrita en la ecuación (2.13), pero a diferencia de éste, la matriz $\boldsymbol{W}$ se construye teniendo en cuenta las clases a las que pertenecen los datos, por lo que se considera a LDA como una transformación supervisada.

Si en PCA el criterio para construir la matriz de transformación era maximizar la varianza de los nuevos elementos proyectados, en el caso de LDA se trata de proyectar los datos de forma que se maximice la dispersión interclase y se minimice la dispersión intraclase. Para definir mejor estos términos, supongamos que, como en el caso de PCA, tenemos un conjunto de datos $\boldsymbol{X}$ compuesta por n vectores columna, cada uno de ellos de m muestras. Además, supondremos que cada vector $\boldsymbol{x}_i$ tiene asignada una etiqueta y_i que identifica la clase a la que pertenece, donde $i = 1, 2, ..., M$. Así, podemos considerar la matriz $\boldsymbol{X}_i$ como una matriz que contiene los n_i vectores pertenecientes a la clase $\mathcal{C}_i$. Las matrices de dispersión interclase $\boldsymbol{S}_B$ e intraclase $\boldsymbol{S}_I$ quedan definidas según:

$$\begin{aligned} \boldsymbol{S}_I &= \sum_{i=1}^{c} \frac{1}{n_i} \left(\boldsymbol{X}_i - \boldsymbol{\mu_i}\mathbf{1}^T\right)\left(\boldsymbol{X}_i - \boldsymbol{\mu_i}\mathbf{1}^T\right)^T \\ \boldsymbol{S}_B &= \sum_{i=1}^{c} n_i \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)^T \end{aligned} \tag{2.19}$$

donde $\boldsymbol{\mu}_i$ es un vector de m componentes con los valores medios de los vectores que forman la matriz $\boldsymbol{X}_i$, mientras que $\boldsymbol{\mu}$ es un vector de m componentes con las medias de todo el conjunto, independientemente de la clase a la que pertenezcan los vectores. El vector $\mathbf{1}$ es un vector de unos de n elementos. Por tanto, la matriz de transformación se calcula mediante los autovectores de la matriz $\boldsymbol{S}_I^{-1}\boldsymbol{S}_B$, ordenados según sus autovalores correspondientes. La reducción se efectúa proyectando la información del vector $\boldsymbol{x}_i$ sobre los k autovectores obtenidos.

2.3.8. Consideraciones sobre los métodos propuestos

Entre los métodos expuestos de procesado de señal existe un grupo de ellos centrado en identificar los parámetros teniendo en cuenta que la señal se puede representar como un conjunto de exponenciales decrecientes, ya que en los sistemas de nariz electrónica modulados por flujo aparecen este tipo de señales. Pertenecen a este tipo de métodos la búsqueda por exponenciales, la extracción por Padé-Z o la técnica que se ha descrito como ridge regression. Sin embargo, como también se ha comentado en la descripción de las técnicas de medidas dinámicas, cada vez se utiliza más la termomodulación en el caso de la nariz electrónica, lo que originará tipos de señales diferentes que no pueden ser caracterizadas por la aproximación de exponenciales decrecientes. En el caso de la lengua electrónica, especialmente cuando la técnica empleada es la ciclovoltiamperometría o la voltametría pulsada, los tipos de señales también varían de la suma de exponenciales decrecientes en el tiempo.

Respecto a los modelos ARMA, su uso para la caracterización y reconocimiento de señales pueden aplicarse cuando tenemos los siguientes criterios:

- Procesos Estocásticos. Esto significa que la señal que obtenemos es la realización de un proceso aleatorio que mantiene constantes la media, la varianza y la autocorrelación.

- Caracterización por la densidad espectral. Los coeficientes de los modelos ARMA se estiman a partir de la información espectral. Por tanto, identifican señales que tienen la misma información espectral aunque tengan formas de onda diferentes, lo que los hace atractivos para determinados campos como el reconocimiento de voz.

La desventaja de utilizar este tipo de modelos con las señales discretas es que el proceso que caracteriza la generación de señales, aún habiendo compensado la deri-

va, no es necesariamente estocástico, ya que la potencia de ruido puede variar entre la adquisición de unas señales a otras. Por otro lado, la información que se pretende capturar de las señales dinámicas de nariz y lengua electrónicas está relacionada con la forma de onda en el tiempo. Pudiera darse el caso de obtener dos formas de onda claramente diferenciadas, procedentes de sustancias completamente distintas que tuvieran una densidad espectral similar y por tanto, los coeficientes del modelo ARMA de ambas sustancias serían parecidos.

Como se destaca en [Hierlemann08] los métodos basados en el espacio de fase son muy sensibles a pequeños cambios en la información proporcionada por los sensores. Esta misma conclusión se reconocía en los trabajos que habían utilizado este tipo de métodos. Por este motivo dicha técnica, aún siendo válida para todo tipo de señales, no es un método adecuado para resolver el problema de la extracción de parámetros que caractericen la señal bajo estudio.

Tanto PCA como LDA son métodos muy extendidos en el campo del reconocimiento de patrones, con excelentes resultados. Sin embargo, en [Röck08] se advierte que en muchas ocasiones se presentan resultados, después de la aplicación de estos métodos, que solamente están relacionados con la varianza debida a la propia naturaleza de tomar señales en momentos diferentes. Además, su uso puede ser complementario al de otras técnicas de extracción, consiguiendo una mejor decorrelación o separación de las mismas.

Respecto a la transformada wavelet, como se ha expuesto, sus características son muy adecuadas para el análisis de forma de onda, pues consigue localización tanto en tiempo como en frecuencia. El principal problema del uso de la transformada wavelet radica en identificar en qué coeficientes está contenida la información de la señal para ser clasificada. Por tanto, para la identificación de un tipo de sustancia tendremos que considerar:

- Un número fijo de coeficientes. Dado que la inmensa mayoría de los clasificadores trabaja con vectores del mismo tamaño para identificar una clase concreta, para cada clase deberemos considerar un número fijo de coeficientes.

- Los coeficientes que maximicen la separación entre clases. No basta con identificar un número fijo de coeficientes que representen adecuadamente la señal. Así, si al clasificador se le pasa un vector $\boldsymbol{v} = [v_1, v_2, ..., v_m]$, la componente v_1 siempre debe hacer referencia al mismo coeficiente wavelet.

En los trabajos descritos en la bibliografía, el problema de determinar los coeficientes wavelet relevantes ha sido resuelto mediante prueba y error, considerando cada vez diversos coeficientes. Este método, si bien es válido para una fase de estudio preliminar sobre la aplicación de la transformada wavelet al análisis de señales de sensores de gases y líquidos, no es viable para una aplicación práctica.

2.4. Métodos de selección de características

Una vez que se han expuesto los principios físicos de los sensores, las técnicas para obtener la información dinámica de los mismos y los métodos de extracción de esa información, se analizará la parte de reconocimiento de patrones. A partir de este momento, la información procedente de los sensores, bien en formato RAW o bien los parámetros que definen las señales de los mismos, se tratará como vectores con una serie de variables o características cada unos de ellos. Antes de pasar a la etapa de clasificación, hay que tener en cuenta que el uso de todas las variables disponibles por cada una de las muestras no suele ser una buena estrategia, ya que alguna de las características (componentes del vector) no serán relevantes para la clasificación, añadiendo únicamente ruido al clasificador. Además, cuanto mayor es el número de características de los patrones de entrada, más complejo se vuelve el problema a discriminar y por tanto, resultará más complicado tener una tasa de error baja. Estos fenómenos son conocidos como *la maldición de la dimensión* en el campo del aprendizaje [Bishop06].

Para definir formalmente la selección de características, supongamos que tenemos un conjunto $\boldsymbol{X}$ de n muestras, teniendo cada una de ellas inicialmente m características. El problema de la selección de características consiste en encontrar un subconjunto D de características óptimas, de forma que los nuevos vectores tendrán d características con $d < m$. La selección de características se realiza mediante una función de criterio o función objetivo $J(D)$, donde D es una de las posibles combinaciones con d elementos seleccionados de las m características totales. Existen dos grandes grupos de funciones objetivo:

Funciones de filtrado En este tipo de funciones la selección se realiza por el contenido de la información de las características, típicamente la distancia interclase o la dependencia estadística. Las funciones de filtrado no tienen en cuenta el tipo de clasificador que existe posteriormente, por lo que su criterio es aplicable a todo tipo de clasificadores. Normalmente, los criterios de filtrado establecen un ranking de cada una de las características y la selección de las mismas se realiza quedándonos con las d que mejor comportamiento presenten. Entre estos criterios, los más populares son los coeficientes de Fisher, donde cada característica r se evalúa como:

$$J(r) = \frac{\sum_{i=1}^{c} n_i (\mu_i - \mu)^2}{\sum_{i=1}^{c} n_i \sigma_i^2} \tag{2.20}$$

donde μ_i y σ_i^2 son la media y dispersión de los valores de la característica r considerando solo los vectores que pertenecen a la clase $\mathcal{C}_i$.

Otros criterios utilizados son los coeficientes de Pearson [Weston00] o el criterio de máxima-relevancia, mínima-redundancia [Peng05].

Funciones por envolvente (wrapper) Los criterios de selección por envolvente toman como función de criterio la salida del clasificador considerado o una estimación de su rendimiento. Son por tanto dependientes del clasificador a utilizar, siendo su criterio de búsqueda más coherente con el objetivo de seleccionar el mejor subconjunto que proporcione los mejores resultados en la clasificación. Los métodos por envolvente proporcionan mejores resultados que los métodos de filtrado, pero la complejidad y tiempo necesario en la fase de obtención del subconjunto de características seleccionadas es mucho mayor. Además, al probar diferentes combinaciones evitan el denominado *efecto de nido*, que se produce al considerar las características de forma aislada, no teniendo en cuenta las combinaciones de las mismas.

En el campo de la nariz y la lengua electrónica, se encuentran varios trabajos de selección de características como [Llobet07], [Gualdron06], [Durán05b] o [Perera02], basándose todos estos trabajos en criterios por envolvente. Entre los métodos presentados, podemos realizar una clasificación según el criterio de búsqueda del conjunto de características a seleccionar:

Algoritmos exponenciales. Evalúan un número de subconjuntos que crece exponencialmente con el número de características totales m del problema. Entre los algoritmos utilizados podemos destacar

- Búsqueda exhaustiva.
- Beam search.

Algoritmos secuenciales. Se evalúa la aportación de añadir o quitar una característica sobre un conjunto dado, siendo su tiempo de cómputo muy inferior a los algoritmos exponenciales. Sin embargo, suelen caer en mínimos locales de la solución. Entre los algoritmos utilizados se encuentran

- Sequential fordward search (SFS).
- Sequential backward search (SBS).
- Plus-l-Minus-R search.
- Sequential floating search (SFFS y SFBS).

Algoritmos aleatorios. La solución que proponen se basa en una búsqueda aleatoria guiada. Tienen la ventaja de ser computacionalmente menos costosos que los algoritmos exponenciales y no tienen tanta tendencia a caer en un mínimo local como los secuenciales. Los algoritmos utilizados de este grupo son:

- Algoritmos Genéticos.
- Simulated Annealing.

Todos estos métodos son guiados por una función criterio $J(D)$ obtenida a la salida del clasificador cuando se entrena y se realiza el test considerando las características del subconjunto D. A continuación se describen los diferentes métodos enumerados y que han sido implementados en esta tesis doctoral con el objetivo de realizar una comparativa entre los mismos y proponer mejoras en algunos de ellos.

2.4.1. Búsqueda exhaustiva

El método de búsqueda exhaustiva, también denominado algoritmo por fuerza bruta, trata de evaluar todas las posibles combinaciones del total de características. Si el número de características a fijar d está determinado a priori tendrá que evaluar $\begin{pmatrix} m \\ d \end{pmatrix}$ combinaciones, mientras que si el número de características a fijar debe ser ajustado, el número de combinaciones a evaluar deberá ser de 2^m combinaciones. Este método es computacionalmente prohibitivo si el número de características inicial m es elevado.

Mediante este algoritmo se llega a la solución óptima, pero no es de aplicación en la mayoría de las señales procedentes de nariz y lengua electrónica, especialmente cuando se analizan señales dinámicas de los sensores. El resto de soluciones siguen una estrategia de búsqueda que encuentra una combinación sub-óptima, pero tienen más aplicación que la búsqueda exhaustiva.

2.4.2. Sequential Search (SFS) y (SBS)

Estos algoritmos son probablemente los más utilizados en la selección de características, debido a su sencillez y velocidad. El algoritmo SFS se basa en los siguientes pasos:

1. Denominar D a la combinación ganadora e iniciar como $D = \{\oslash\}$.
2. Seleccionar la característica r_g tal que $r_g = \underset{\text{máx}}{\arg} J(D, r), r \notin D$.
3. Si $J(D, r) < J(D)$, devolver D y detener el algoritmo.
4. Asignar a la combinación ganadora $D = [D, r]$.
5. Si el número de características de la combinación ganadora es el deseado, devolver D y detener el algoritmo.
6. Volver al paso 2.

El algoritmo SFS funciona bien cuando la combinación ganadora tiene un número de características pequeño, ya que a medida que avanza el algoritmo las regiones que explora son mucho menores. El principio del algoritmo es que una vez que se añade una característica, se queda permanentemente en la combinación ganadora, por lo que existe el riesgo de caer en un mínimo local.

El algoritmo SBS es muy similar al anterior, pero la búsqueda se hace quitando características. Así, los pasos a dar son los siguientes:

1. Denominar D a la combinación ganadora e iniciar como el conjunto de todas las características posibles $D = \{r_1, r_2, ..., r_m\}$.

2. Seleccionar la característica r_l tal que $r_l = \arg\min J(D - \{r\}) = \arg\min J(D^{\setminus r}), r \subset D$.

3. Si $J(D^{\setminus r}) < J(D)$, devolver D y detener el algoritmo.

4. Quitar de la combinación ganadora la muestra r, $D = [D^{\setminus r}]$.

5. Si el número de características de la combinación ganadora es el deseado, devolver D y detener el algoritmo.

6. Volver al paso 2.

Al contrario que SFS, el algoritmo SBS funciona bien cuando la combinación ganadora tiene muchas muestras, pues al comienzo se explora mucha parte del espacio de combinaciones a eliminar. Sin embargo, a medida que va avanzando el algoritmo la solución es mucho más rígida, ya que una vez tomada la decisión de quitar de la combinación ganadora una característica, ésta no vuelve a añadirse.

2.4.3. Beam search

El algoritmo beam search se basa en una cola que contiene en cada momento las k mejores soluciones. Los pasos de los que consta el algoritmo son los siguientes:

1. Evaluar la función de criterio $J(r_i)$ para cada una de las posibles características. Guardar en la cola las k características que obtengan mejor resultado. En este momento, las combinaciones de la cola están formadas por una única característica.

2. Evaluar todas las posibles combinaciones resultantes de añadir una característica a las combinaciones de la cola. La característica a añadir no estará repetida dos veces en la misma combinación. Ordenar los resultados y guardar en la cola las k combinaciones que tengan mejor resultado.

3. Si el número de características es el deseado, detener el algoritmo. Si no, volver al paso 2.

Si $k = m$ este algoritmo se convierte en la búsqueda exhaustiva, mientras que si $k = 1$ el algoritmo se convierte en el SFS.

2.4.4. Plus-l-Minus-r (LRS)

Este algoritmo fue propuesto originalmente en [Stearns76] y trata de compensar las desventajas de SFS y SBS, permitiendo que una muestra que es añadida pueda ser eliminada de la combinación ganadora. El funcionamiento del algoritmo, depende de los pasos que demos hacia adelante l y de los pasos establecidos para atrás r. Si $l > r$ los pasos del algoritmo son los siguientes:

1. Denominar D a la combinación ganadora e iniciar como $D = \{\oslash\}$ y k=1, z=1.
2. Seleccionar la característica q_g tal que $q_g = \underset{\text{máx}}{\arg} J(D, q), q \notin D$. Hacer $D = \{D, q_g\}$.
3. Incrementar k=k+1. Si $k = l$, asignar $k = 1$ y continuar al paso 4. Si no, repetir el paso 2.
4. Seleccionar la característica q_l tal que $q_l = \underset{\text{mín}}{\arg} J(D^{\backslash q}), q \subset D$. Hacer $D = D - \{q_l\}$.
5. Incrementar z=z+1. Si $z = r$, asignar $z = 1$ y continuar al paso 6. Si no, repetir el paso 4.
6. Si el número de características de la combinación ganadora es el deseado, detener el algoritmo. Si no, repetir desde el paso 2.

Por tanto, el algoritmo efectúa l pasos del método SFS y posteriormente efectúa r pasos del algoritmo SBS. Si por el contrario, $r > l$ se efectúan en primer lugar r pasos del algoritmo SBS y posteriormente l pasos del algoritmo SFS. De esta forma se evita caer en los mínimos locales de los algoritmos SFS y SBS y se extiende el espacio de búsqueda. Sin embargo, el algoritmo en su forma original requiere de un número d de características deseadas y de los valores l y r.

2.4.5. Sequential Floating Search (SFFS) y (SFBS)

Este algoritmo propuesto en [Pudil94] es la evolución natural del método LRS. En lugar de determinar los valores de l y de r, será el propio algoritmo en su evolución el que vaya fijando estos valores de forma flotante. Existen dos versiones del algoritmo, una es la que parte del conjunto vacío (SFFS) y la otra parte del conjunto con todas las características. Puesto que son análogas se exponen los pasos del algoritmo SFFS:

1. Denominar D a la combinación ganadora e iniciar como $D = \{\oslash\}$, $FF = 1, BF = 1$
2. Seleccionar la característica r_g tal que $r_g = \underset{\text{máx}}{\arg} J(D, r), r \notin D$.
3. Si $J(D) < J(D, r_g)$, actualizar $D = \{D, r_g\}$, $FF = 1$ y saltar al paso 5.
4. Si $BF = 0$, terminar el algoritmo. Si no, poner $FF = 0$.
5. Seleccionar la característica r_l tal que $r_l = \underset{\text{mín}}{\arg} J(D^{\setminus r}), r \subset D$.
6. Si $J(D) < J(D^{\setminus r_l})$, actualizar $D = D - \{r_l\}$, $BF = 1$ y saltar al paso 2.
7. Si $FF = 0$ terminar el algoritmo. Si no, poner $BF = 0$.
8. Si el número de características es el deseado, devolver D. Si no volver al paso 2.

Mediante este algoritmo y su homólogo, los valores de l y de r se ajustan automáticamente en cada iteración. El algoritmo expuesto, sólo permite añadir o eliminar una característica al mismo tiempo, aunque existen otras versiones que permiten evaluar tuplas de O elementos [Somol00].

2.4.6. Algoritmos genéticos

Los algoritmos genéticos [Goldberg89] son procesos aleatorios de búsqueda basados en los principios de la selección y la evolución natural. La adaptación de este tipo de algoritmos para la selección de características para la nariz electrónica fueron introducidos en [Kermani99]. Las posibles combinaciones de características son codificadas en forma de cadenas de bits, de manera que una posible solución vendrá descrita por una sucesión de unos y ceros indicando la presencia o la ausencia, mediante un uno o un cero respectivamente, de cada una de las variables en esa combinación particular. En el caso de los algoritmos genéticos cada combinación, que representa una posible solución, es denominada individuo y la cadena de ceros y unos se denominará cromosoma. Aunque los algoritmos genéticos se basan en soluciones aleatorias, la búsqueda se guía mediante una *función de fitness*, que en nuestro caso será el nivel de acierto a la salida del clasificador con una combinación determinada. Aunque existen diferentes versiones para los algoritmos genéticos, en todas las referencias encontradas en el campo de la nariz y la lengua electrónica se sigue el siguiente algoritmo:

1. La búsqueda se inicia con un conjunto de h posibles soluciones generadas aleatoriamente denominadas población inicial.

2. Se evalúa la población actual mediante la función de fitness y se asigna un valor a cada individuo conforme a su valor de fitness, estableciendo un ranking de individuos.

3. Se permite sobrevivir a la siguiente generación a los n^{best} individuos.

4. Se realiza una selección de $h - n^{best}$ individuos. La selección se hace de forma aleatoria pero la probabilidad de cada individuo de ser seleccionado es proporcional al valor que obtuvo en la función de fitness. De esta manera, los que mejor resultado han demostrado tienen más probabilidad de ser seleccionados.

5. Con los individuos seleccionados se hace una operación de cruce (crossover) por pares de individuos. Normalmente se toman los b primeros bits de uno de los individuos y los $m-b$ bits del otro. Así, dado que cada bit codifica la presencia o ausencia de una característica dada, un nuevo cromosoma formado mediante una operación de cruce es una combinación de dos individuos que han dado buenos resultados.

6. Se realiza una operación de mutación con una probabilidad muy pequeña sobre los individuos. La mutación consistirá en cambiar un cero por un uno o un uno por cero. Al igual que sucede en la naturaleza, la mutación no se debe dar muy frecuentemente.

7. Una vez realizadas las operaciones de cruce y mutación tendremos $h-n^{best}$ nuevos individuos que, junto con los n^{best} de la población anterior constituyen una nueva población. Se repite el algoritmo desde el paso 2.

El algoritmo genético prosigue hasta que iguala o supera un valor de la función de fitness establecido como meta, hasta que exista una convergencia en la población, de manera que un determinado porcentaje de sus miembros acaben siendo idénticos, o hasta que se llegue al número máximo de iteraciones.

Los algoritmos genéticos serán tratados en más profundidad en esta tesis como métodos de optimización. En la selección de características proporcionan la ventaja de explorar una gran parte del espacio de soluciones e ir guiando la búsqueda mediante las operaciones de cruce, mientras que la operación de mutación permite escapar de un mínimo local. Entre las críticas que suelen recibir los algoritmos genéticos para la selección de características, tal como se ha expuesto el algoritmo, se indica que es difícil conocer cuándo parar el algoritmo y que están basados en una solución aleatoria.

2.4.7. Simulated Annealing

El algoritmo Simulated annealing (SA) o del temple simulado es una técnica estocástica inspirada en la termodinámica y en concreto aprovecha la analogía en la forma en la que una aleación de hierro fundido alcanza un estado de mínima energía a medida que se va enfriando la temperatura. Su introducción en el campo de la nariz electrónica se encuentra en [Gualdron06]. Al igual que con los algoritmos genéticos, se establece el procedimiento propuesto en dicho trabajo.

1. Comenzar el algoritmo a una temperatura inicial $T = T_0$, $step = 0$. Obtener aleatoriamente una combinación inicial D_w y calcular $J_w = J(D_w)$.

2. Calcular una solución vecina a D_w denominada D_n. Esta solución vecina se genera cambiando el valor de los bits con una probabilidad independiente $p = KT$, siendo K una constante a ajustar a priori que asegura que $KT_0 \ll 1$. Calcular $J_n = J(D_n)$.

   - Si $J_w > J_n$. Aceptar la nueva solución como solución ganadora
     $J_w \leftarrow J_n$
     $D_w \leftarrow D_n$
   - Si $J_w < J_n$ Aceptar la nueva solución con una probabilidad

   $$P = e^{-\frac{\triangle E}{T}} \tag{2.21}$$

   donde $\triangle E = J_n - J_w$.

3. Si $step \neq step_{max}$, hacer $step = step + 1$ y se enfría la temperatura de trabajo según:

   $$T = T_0 + \frac{step}{step_{max}}(T_f - T_0) \tag{2.22}$$

   volver al punto 2.

4. Si $T = T_f$ detener el algoritmo y devolver D_w. Si no, decrementar la temperatura y repetir desde el paso 2.

La ecuación (2.21) es la aceptación de probabilidad de Boltzmann y trata de escapar de mínimos locales. Cuando la temperatura es muy alta, la probabilidad de aceptar nuevas soluciones es también muy elevada. A medida que va descendiendo o enfriandose la temperatura, las soluciones que han resultado peores tienen una probabilidad muy baja de ser seleccionadas. En el citado trabajo no señalan cómo generar la solución vecina, aunque lo común en este tipo de algoritmos es realizar una mutación de los bits con una baja probabilidad.

La clave de la convergencia del algoritmo simulated annealing está en el esquema de enfriamiento de temperatura [Salomon02]. Con un esquema de enfriamiento suficientemente lento se asegura la convergencia hacia la solución óptima, pero el algoritmo deriva a la búsqueda exhaustiva. Por el contrario, en un esquema muy rápido de enfriamiento, la solución aceptada es prácticamente aleatoria.

2.5. Clasificación de Patrones

La clasificación tiene por objetivo identificar las señales de los sensores, o los parámetros extraídos de las mismas como se ha descrito en este capítulo, y asignar la sustancia bajo análisis a una clase. Así, se parte de un conjunto de vectores de entrenamiento $\boldsymbol{X}$ del que podemos conocer la clase a la que pertenecen y_i, en el caso de clasificación supervisada, o no conocerlo en esta fase de entrenamiento, en el caso de la clasificación no supervisada. En cualquier caso, la meta final de todo método de clasificación es encontrar la forma de relacionar $\boldsymbol{x}_i \rightarrow y_i$ para futuras muestras. Habría que señalar que el término *reconocimiento de patrones* es más amplio que la clasificación, ya que comprende la regresión o medida de una serie de propiedades cuantificables, como puede ser la concentración de una sustancia en un analito más complejo, y las técnicas de *clustering*, cuya misión es aprender las relaciones estructurales entre diferentes señales de sensores. En esta tesis, como se mencionó en el primer capítulo, el trabajo se ha centrado en el campo de la clasificación supervisada, por ser el que más relación tiene con los conceptos definidos de nariz y lengua electrónica.

Probablemente, el método más referenciado en múltiples trabajos sobre sensores de gases y líquidos es el análisis por componentes principales (PCA), especialmente realizando un PCA plot. Aunque en muchas ocasiones permite separar de forma visual una clase sobre otra, muchos autores no lo consideran en sí mismo un método de clasificación, pues no proporciona la expresión matemática que mapea $\boldsymbol{x}_i \rightarrow y_i$.

Uno de los objetivos de esta tesis es establecer una comparación entre los diferentes métodos de clasificación y proponer mejoras de los mismos, por lo que se aborda en esta sección los métodos utilizados en los sistemas de nariz y lengua electrónicas. Los trabajos descritos en [Hines99] y [Gutierrez-Osuna02] son una primera referencia de los métodos de clasificación empleados en el campo de la nariz electrónica, donde se describen los siguientes métodos:

- Fischer linear discriminant (FLD).
- k Nearest neighbor.
- Perceptrón multicapa.
- Fuzzy Artmap.

- Clasificadores de base radial (RBF).

En el campo de la lengua electrónica encontramos otros dos métodos ampliamente utilizados:

- Soft independent modeling of class analogies (SIMCA).
- Partial least squares discriminant analysis (PLS-DA).

Además de estos métodos, en la revisión bibliográfica podemos encontrar otros métodos recientes como son:

- Máquinas de vectores soporte (SVM), [Pardo05], [Distante03].
- Random Forest, [Pardo07].

A continuación se exponen los principios de funcionamiento y principales características de cada uno de estos métodos.

2.5.1. Fischer linear discriminant (FLD)

El método Fischer linear discriminant está íntimamente relacionado con el análisis discriminante lineal (LDA), expuesto en la extracción de características. Si en el caso de LDA el objetivo era proyectar los vectores sobre un nuevo espacio, de forma que las matrices de dispersión intraclase ($\boldsymbol{S}_I$) e interclase ($\boldsymbol{S}_B$), definidas en la ecuación (2.19), fueran minimizadas y maximizadas respectivamente, en este caso se trata de encontrar, para el conjunto de entrenamiento, un plano de separación $[\boldsymbol{\omega}, b]$ que consiga la siguiente función de decisión

$$f(\boldsymbol{x}) = \text{signo}\left(\langle \boldsymbol{\omega} \cdot \boldsymbol{x} \rangle + b\right) \tag{2.23}$$

Para encontrar el plano de separación $\boldsymbol{\omega}$, atendiendo a las matrices de dispersión, se deberá resolver el problema:

$$\boldsymbol{\omega} = \underset{max}{\arg} F(\boldsymbol{\omega}') = \frac{\langle \boldsymbol{\omega}' \cdot \boldsymbol{S}_B \boldsymbol{\omega}' \rangle}{\langle \boldsymbol{\omega}' \cdot \boldsymbol{S}_I \boldsymbol{\omega}' \rangle} \tag{2.24}$$

Supongamos un problema binario de clasificación, en el que las clases asociadas a los vectores $\boldsymbol{x}_i$ quedan determinadas mediante la etiqueta $y_i \in \{+1, -1\}$. La implementación clásica de la solución al problema planteado en la ecuación (2.24) vendría determinada por:

$$\boldsymbol{\omega} = \boldsymbol{S}_I^{-1}\left(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-\right) \tag{2.25}$$

donde $\boldsymbol{\mu}^+$ y $\boldsymbol{\mu}^-$ son las medias de los vectores pertenecientes a cada clase. El desplazamiento b, se puede encontrar manteniendo la siguiente igualdad:

$$\langle \boldsymbol{w} \cdot \boldsymbol{\mu}^+ \rangle + b = -\left(\langle \boldsymbol{w} \cdot \boldsymbol{\mu}^- \rangle + b\right) \tag{2.26}$$

Debe observarse que el método del discriminante lineal de Fischer considera que la función densidad de probabilidad de cada uno de los conjuntos es gaussiana y separable linealmente. Existen implementaciones no lineales de este método, como es el método Kernel Fischer Discrimant, pero en el campo de la nariz y la lengua electrónica se encuentra en pocas referencias, como por ejemplo en [Decoste01].

2.5.2. k nearest neighbor

El algoritmo k nearest neighbor (kNN), o de los k vecinos más cercanos, ha sido ampliamente utilizado en el campo de la nariz y lengua electrónica debido a su potencia para tratar problemas de clasificación no lineales. Este algoritmo guarda los vectores del conjunto de entrenamiento junto con sus clases, de forma que cuando se presenta un nuevo vector $\boldsymbol{x}_t$ se medirá la distancia con todos los vectores de entrenamiento y se seleccionará la clase predominante de los k vectores más próximos a la nueva muestra.

Aunque el planteamiento parece extremadamente sencillo, el algoritmo kNN es una formulación no paramétrica del criterio maximum a posteriori (MAP). De hecho, con un número elevado de muestras de entrenamiento, el algoritmo se aproxima a un clasificador bayesiano [Duda00]. Considerar un algoritmo de clasificación como no paramétrico significa no presuponer ningún tipo de distribución a priori.

Las principales desventajas del método son la cantidad de memoria requerida para almacenar todo el conjunto de entrenamiento y el número de operaciones requeridas en la fase de test debido a que para cada muestra nueva hay que calcular las distancias con todas las muestras del conjunto de entrenamiento.

2.5.3. Perceptrones multicapa

Dentro de las redes neuronales artificiales, el modelo más popular son los denominados perceptrones multicapa, consistentes en una red de unidades sencillas, denominadas neuronas, y constituidas por varias capas. Cada neurona desarrolla una suma promediada por pesos de sus entradas y se activa de acuerdo a una función, habitualmente no lineal. En el caso de los perceptrones multicapa, la información fluye desde la entrada hasta la salida, sin existir ningún tipo de conexión realimentada, por lo que se denominan redes de propagación directa. Aunque el modelo del perceptrón multicapa permite múltiples capas ocultas, en la bibliografía consultada sobre la aplicación de

este tipo de métodos a la nariz y lengua electrónica sólo se utiliza una capa oculta. Una vez entrenada, la salida de la neurona i vendrá dada por la expresión:

$$f_i(\boldsymbol{x}) = f_o\left(\sum_{j=1}^{N} \omega_{oj}^i \left(f_h\left(\langle \boldsymbol{\omega}_h^j \cdot \boldsymbol{x}\rangle + b_{hj}\right)\right)\right) + b_i \tag{2.27}$$

donde $f_h(.)$, $f_o(.)$ son las funciones de activación de la capa oculta y de la capa de salida respectivamente, $\boldsymbol{\omega}_h^j$ es el vector de pesos de la neurona j perteneciente a la capa oculta, $\boldsymbol{\omega}_o^i$ es el vector de pesos de la neurona i de salida, N es el número de neuronas de la capa oculta, mientras que b_{hj} y b_i son los denominados *bias* o constantes de desplazamiento de cada neurona. En [Cybenko89] se demuestra que con este esquema se pueden aproximar casi todas las funciones de decisión, por lo que la flexibilidad de los perceptrones multicapa permite resolver problemas complejos de clasificación.

El número de neuronas de la capa oculta N debe fijarse a priori, mientras que el número de neuronas de la capa de salida viene determinado por el número de clases del problema. Las funciones de activación también deben ser seleccionadas antes del entrenamiento, debiendo cumplir que sean continuas no constantes, monótonas crecientes y estar acotadas.

Durante el entrenamiento de una red neuronal se calcularán los vectores de pesos y los bias de todas las neuronas de la red. El método más común de entrenamiento es el ajuste por retropropagación del error, que tratará de minimizar una función criterio $e(.)$. Aunque existen varias funciones criterio [Duda00], [Bishop06], la más popular es el error cuadrático medio (MSE) definido por:

$$e(\boldsymbol{\omega}) = \frac{1}{2} \parallel \boldsymbol{y} - \boldsymbol{f} \parallel^2 \tag{2.28}$$

donde $\boldsymbol{y}$ es el vector de etiquetas o clases a las que pertenece el conjunto de entrenamiento y $\boldsymbol{f}$ es el vector de salidas estimadas por la red. El método de retropropagación del error requiere, como se ha mencionado, que las funciones de activación sean derivables para poder ajustar el error en la dirección de minimización de $e(.)$. Existen varios métodos de entrenamiento para buscar la minimización de la función criterio, de forma que se vayan actualizando los pesos de las neuronas. Entre los diferentes métodos podemos destacar:

Descenso del gradiente. Definiendo el operador gradiente como:

$$\nabla = \left[\frac{\partial}{\partial \omega_1}, \cdots, \frac{\partial}{\partial \omega_n}\right]^T \tag{2.29}$$

La actualización de pesos en el instante $t+1$ se consigue mediante:

$$\boldsymbol{\omega}^{t+1} = \boldsymbol{\omega}^t - \eta \nabla e(\boldsymbol{\omega}^t) = \boldsymbol{\omega}^t - \eta \boldsymbol{J}^t \tag{2.30}$$

donde $\eta < 1$ es una constante a ajustar a priori. Un valor muy alto de esta constante hace que el vector generado quede muy alejado del vector actual, pudiendo degenerar el algoritmo de aprendizaje. Un valor pequeño de esta constante hace que el entrenamiento sea excesivamente lento, por lo que en esta tesis se ha decidido utilizar algoritmos que adapten este parámetro a medida que transcurre el tiempo, siendo grande al principio para que se pueda avanzar más rápido y un valor pequeño al final que refina la solución. La matriz $\boldsymbol{J}$ es el Jacobiano de la función de error evaluada en el punto $\boldsymbol{\omega}$.

Método de Gauss-Newton. Para funciones de criterio como el error cuadrático medio, es posible emplear los métodos de Gauss-Newton. En determinados problemas de optimización, la convergencia es más rápida si se usa la información de la derivada segunda, lo que equivale a utilizar la matriz Hessiana en lugar del Jacobiano de la ecuación (2.30) fijando además $\eta = 1$. Este tipo de entrenamiento se denomina método de Newton y aunque es mucho más rápido que el método de descenso por el gradiente el cálculo de la Hessiana es computacionalmente costoso. Para funciones de error como el error cuadrático medio, es posible aproximar la Hessiana por:

$$\boldsymbol{H} = \left(\boldsymbol{J}^T \boldsymbol{J}\right)^{-1} \tag{2.31}$$

Así, los métodos Gauss-Newton calculan el error mediante:

$$\boldsymbol{\omega}^{t+1} = \boldsymbol{\omega}^t - \left(\boldsymbol{J}^t \boldsymbol{J} + \alpha \boldsymbol{I}\right)^{-1} \boldsymbol{J}^t \boldsymbol{e}\left(\boldsymbol{\omega}^t\right) \tag{2.32}$$

donde α es un parámetro que debe ser ajustado. En el caso de esta tesis se ha utilizado como método Gauss-Newton el algoritmo de Levenberg-Marquardt que adapta el parámetro α en el tiempo. Este método converge más rápido que el método utilizado por descenso de gradiente, pero los requerimientos de memoria hacen que solo sea viable para redes pequeñas.

Hay que tener en cuenta que en los perceptrones multicapa existen varios parámetros que deben ser ajustados a priori, como son el número de neuronas en la capa oculta, las funciones de activación, los parámetros de aprendizaje y el número de iteraciones que permitimos a los métodos de entrenamiento. Si se permiten pocas iteraciones no se ajustarán los pesos correctamente, mientras que si se permiten muchas iteraciones se tiende al sobreaprendizaje del conjunto de entrenamiento. Además, se debe tener en cuenta la existencia de mínimos locales en la función de error, por lo que se debe comenzar por vectores de pesos aleatorios.

2.5.4. Redes fuzzy artmap

Las redes fuzzy artmap, descritas con profundidad en [Carpenter92], son una evolución de las redes artmap [Carpenter91] de forma que permiten trabajar con patrones reales. Tanto las redes artmap como fuzzy artmap se basan en las redes ART, acrónimo de teoría de resonancia adaptativa, que son un método no supervisado para crear clusters o agrupaciones de muestras sin suponer una distribución de los datos de entrada.

Antes de ver su funcionamiento, describiremos los elementos característicos de una red fuzzy artmap:

Entradas. Se parte de vectores de entrada normalizados $0 < x_i(j) < 1$, $j = 1, 2, ..d$ asociados con una clase $y_i = 1, 2, .., M$. Para cada vector de entrada $\boldsymbol{x}_i$ se crea un nuevo vector $\boldsymbol{v}_i$ de dimensión $2d$, obtenido mediante codificación complementaria, de forma que:

$$\begin{array}{l} v\,(2j-1) = x\,(j) \\ v\,(2j) = 1 - x\,(j) \end{array} \quad j = 1, 2, .., d. \tag{2.33}$$

Red artmap. Está compuesta por una serie de neuronas, cada una con un peso formado por un vector de dimensión igual a la de los patrones que se presentan a la red ($2d$). Cada neurona tendrá asignada una entrada en el mapfield. La red se encuentra vacía al comenzar el entrenamiento.

Mapfield. Se trata de una tabla que asigna a cada neurona la clase con la que está relacionada.

Parámetro de vigilancia máxima (ρ_{max}). Este parámetro controla si un nuevo vector presentado a la red artmap está en el entorno de alguna de las neuronas de dicha red. El parámetro toma valores entre cero y uno, siendo en este último caso necesario que sea idéntica la muestra presentada al peso de la neurona con la que se está comparando. A este parámetro se le asigna un valor inicial, aunque el proceso de entrenamiento irá cambiando su valor.

Operación de vigilancia. Se establece entre el vector de entrada $\boldsymbol{v}$ y cada una de las neuronas como:

$$\rho_j = \frac{\sum_{k=1}^{2d} \min\left(v_k, \omega_k^j\right)}{\sum_{k=1}^{2d} v_k} \tag{2.34}$$

Operación de similitud. La similitud entre una entrada $\boldsymbol{v}$ presentada a una neurona y el peso $\boldsymbol{\omega}_j$ de la misma se mide mediante:

$$d_{i,j} = \frac{\sum_{k=1}^{2d} \min\left(v_i\left(k\right), \omega_j\left(k\right)\right)}{\sum_{k=1}^{2d} v_i\left(k\right)} \tag{2.35}$$

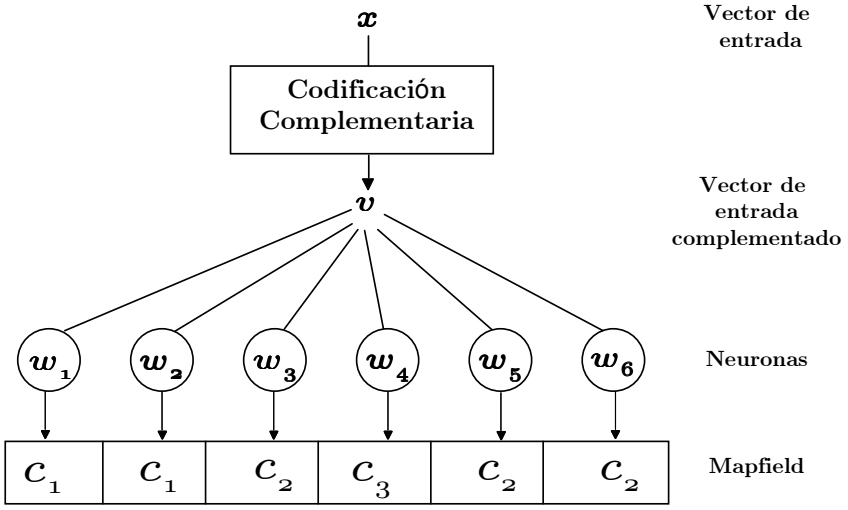


Figura 2.15: Esquema de una red fuzzy artmap.

Una vez entrenada, para una nueva muestra a clasificar, se sigue el esquema de la figura (2.15). Cada muestra a clasificar se complementa siguiendo la ecuación descrita en (2.33) formando el vector $\boldsymbol{v}$. Para cada una de las neuronas de la red se mide similitud según la ecuación (2.35). Adicionalmente, se mide la vigilancia entre la entrada y cada neurona siguiendo la expresión (2.34). Se selecciona la neurona j cuya distancia al vector de entrada haya resultado menor y se comprueba que $\rho_j < \rho_{max}$. Si esta última condición se cumpliera se asigna a la entrada el valor de la clase $\mathcal{C}_j$ del mapfield asociado con la neurona j. Si no se cumpliera la condición de vigilancia, se continua con la siguiente neurona con distancia más pequeña. Si no se cumple la condición de vigilancia para ninguna de ellas, se dice que la entrada no tiene asignada una clase conocida.

Para la fase de entrenamiento, necesitamos además de los elementos característicos de una red fuzzy artmap, definir el parámetro de *velocidad de ajuste* (β). Este parámetro controla la adaptación de los pesos a las nuevas muestras presentadas. Al igual que ocurría en la ecuación (2.30) un valor muy pequeño hará que la red converja muy lentamente, mientras que un valor elevado hará que el aprendizaje se vuelva inestable. Durante la fase de entrenamiento, para cada vector $\boldsymbol{v}_i$, construido con el complemento de los patrones del conjunto de entrenamiento $\boldsymbol{x}_i$, se ejecuta el siguiente algoritmo:

1. Comprobar que la clase del vector $\boldsymbol{v}_i$ se encuentra en alguna de las relaciones del mapfield. Si encontramos que existe una neurona con la misma clase, saltar al paso 3.

2. Se crea una nueva neurona j, asignando como peso el vector de entrada y la clase del mapfield:

$$\begin{aligned} \boldsymbol{\omega}_j &= \boldsymbol{v}_i \\ MF(j) &= y_i \end{aligned} \tag{2.36}$$

   Se marca cambio = 1. Se termina con este vector.

3. Crear una lista L, ordenada de forma creciente según el valor $d_{i,j}$, siendo este término la similitud del vector de entrada a cada una de las neuronas de la red Artmap. Asignar $l = 1$.

4. Considerar como neurona activa $n_a = L(l)$.

5. Si $d_{i,n_a} \leq \rho$ la neurona activa y el vector de entrada no están en resonancia, por lo que se salta al paso 2. Si $d_{i,n_a} > \rho$ la neurona activa ha resonado con el vector de entrada. Se comprueba que $MF(n_a) = y_i$. Si no coinciden las clases saltar al siguiente paso. Si coinciden, se actualiza el peso de la neurona:

$$\boldsymbol{\omega}^{t+1} = (1-\beta)\left(\boldsymbol{v}_i \wedge \boldsymbol{\omega}_{na}^{t}\right) + \beta\boldsymbol{\omega}_{na}^{t} \tag{2.37}$$

   donde el operador $\wedge$ selecciona el mínimo de cada característica. Se marca cambio= 1. Se termina con este vector.

6. El vector ha resonado con una neurona que no es de su misma clase, por lo que se incrementa el parámetro de vigilancia $\rho = \rho + \varepsilon$. Se incrementa el valor $l = l + 1$, se selecciona como neurona activa $n_a = L(l)$ y se repite el paso 5. Si no hay más neuronas en la lista, ir al paso 2.

El algoritmo de entrenamiento se ejecuta para todas las muestras del conjunto de entrenamiento y se repite hasta que no existan cambios o se alcance un número máximo de generaciones. Como se puede apreciar, el algoritmo de entrenamiento de las redes fuzzy artmap es muy sencillo, consiguiendo un error empírico nulo en el conjunto de entrenamiento. Además, las operaciones en las que se basa provienen de la lógica difusa, por lo que no presupone ningún tipo de distribución, lo que convierte a este clasificador en un clasificador no paramétrico. Para la fase de test, en esta tesis se ha implementado el criterio descrito en [Brezmes01], por el que una vez entrenada la red el parámetro de vigilancia queda constante.

2.5.5. Redes de base radial (RBF)

Las redes de base radial se engloban normalmente como redes neuronales de una capa oculta y una capa de salida con conexiones directas. A diferencia de los perceptrones multicapa, la función de la capa oculta, $\phi(.)$, es una función dependiente de la distancia, euclídea o de Mahalanobis dependiendo del tipo de algoritmo, entre el vector de entrada y el vector de pesos de la neurona. Por otro lado, la función de la capa de salida es una función lineal. Así, la función de decisión, una vez entrenada la red, quedará:

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} \omega_j \boldsymbol{\phi}_j(\boldsymbol{x}) + b \tag{2.38}$$

donde N es el número de neuronas de la capa oculta, ω_j el peso de la salida de la neurona j en la neurona de salida y b es una constante de desplazamiento de la neurona de salida. Las funciones de base radial más populares, y utilizadas en esta tesis, son las de tipo gaussiano:

$$\boldsymbol{\phi}_j(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{\omega}_j\|^2}{2\sigma_j^2}\right) \tag{2.39}$$

donde $\boldsymbol{\omega}$ es un vector propio de la neurona y σ^2 es una constante a ajustar a priori. La decisión de la clase a la que pertenece el vector $\boldsymbol{x}$ se hará mediante el signo de la ecuación (2.38) si se trata de un problema binario o mediante un redondeo al entero más cercano en caso de problemas multiclase. La ecuación (2.39) establece una gaussiana con media en el punto definido por $\boldsymbol{\omega}$ y cuya anchura queda determinada por el parámetro σ. Durante el proceso de entrenamiento se ajustan los pesos de las neuronas, tanto de la capa oculta como de la capa de salida. Sin embargo, el método de entrenamiento difiere respecto a lo expuesto en las redes neuronales de tipo perceptrón multicapa. Los métodos clásicos de entrenamiento seleccionan un algoritmo no supervisado para la capa oculta, como el clustering por k-means [Hush93] o el algoritmo EM, acrónimo de Expectation Maximization [Bishop06]. El algoritmo seleccionado para el entrenamiento de redes RBF en esta tesis está basado en el algoritmo *orthogonal least squares* [Chen91] y selecciona como centros de las gaussianas aquellos que explican un mayor incremento en las variables de salida o clases asociadas con los patrones.

Las redes RBF requieren menos parámetros de ajuste que los perceptrones multicapa y consiguen magníficos ratios de precisión en la clasificación si las muestras futuras se encuentran próximas a los patrones de entrenamiento. Sin embargo, el número de neuronas de la capa oculta suele ser mucho mayor que en el caso de los perceptrones multicapa.

2.5.6. Soft independent modelling of class analogies (SIMCA)

En secciones anteriores se describió la utilidad del análisis de componentes principales (PCA) para obtener un nuevo conjunto de características reducidas que contuvieran parte de la información de los datos originales. Sin embargo, PCA no utiliza la información de las clases para realizar esta transformación, sino que intenta describir la variación total de las clases. El método soft independent modeling of class analogies (SIMCA), cuya descripción ampliada puede encontrarse en [Wold77], trata de aprovechar las ventajas de PCA incorporando la información de las clases. Este método se basa en realizar un análisis PCA por cada una de las clases, utilizando para construir el modelo PCA_i sólo las muestras dentro de la clase $\mathcal{C}_i$.

Estudiando los estadísticos descritos en el análisis PCA, descritos en las ecuaciones (2.17) y (2.18),con un número suficiente de componentes principales las muestras pertenecientes a una clase deberían ser similares a las reconstruidas, por lo que las muestras asociadas con esa clase tendrán un valor del estadístico Q pequeño. Por otro lado, si las muestras están agrupadas, al proyectarlas sobre el nuevo espacio reducido, estarán próximas entre sí, por lo que el valor del estadístico Hotellings T^2 debería ser pequeño para las muestras utilizadas en ese modelo. Para poder hacer comparaciones entre modelos, y definir de un modo más riguroso qué es un valor del estadístico grande o pequeño, hablamos de los estadísticos normalizados Q_n y T_n^2 como:

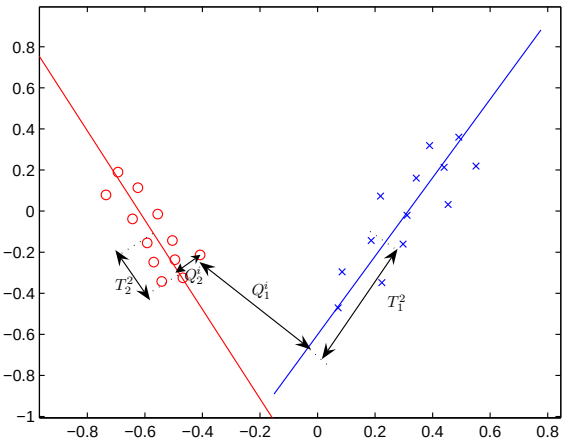
$$Q_{ni} = \frac{Q_i}{Q_{0,95}} \quad T_{ni}^2 = \frac{T_i^2}{T_{0,95}^2} \tag{2.40}$$

donde $Q_{0,95}$ y $T_{0,95}^2$ son los límites de confianza de los estadísticos con un 95 %. En la figura (2.16(a)) se puede ver un conjunto de datos de dos clases. Para cada grupo de muestras de una misma clase en esta figura se hace un PCA, tomando en este caso una única componente para cada modelo. Se puede apreciar cómo una muestra del conjunto rojo tendrá valores Q y T^2 altos respecto del modelo construido con las muestras que no son de su clase, mientras que los mismos estadísticos respecto al modelo construido con las muestras de su misma clase presenta unos valores de Q y T^2 pequeños. Esta situación puede verse mejor en las figuras (2.16(b)) y (2.16(c)) en la que se representan los estadísticos normalizados en ambos modelos para todas las muestras.

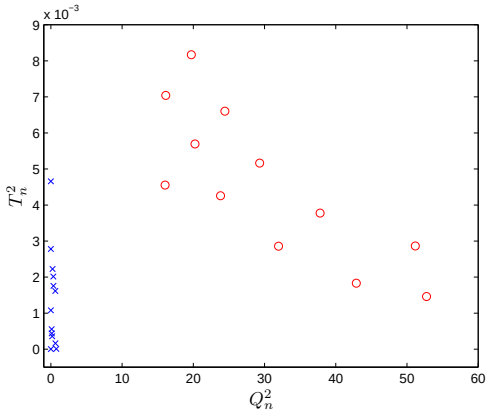
Por tanto, en la fase de test, las nuevas muestras medirán sus estadísticos respecto a los modelos construidos en el entrenamiento con las diferentes clases. Para una nueva muestra $\boldsymbol{x}_i$, se define el concepto de distancia al modelo j como:

$$d_{i,j}^2 = \left(\frac{Q_{i,j}}{Q_{0,95}^j}\right)^2 + \left(\frac{T_{i,j}^2}{T_{0,95}^{2j}}\right)^2 \tag{2.41}$$

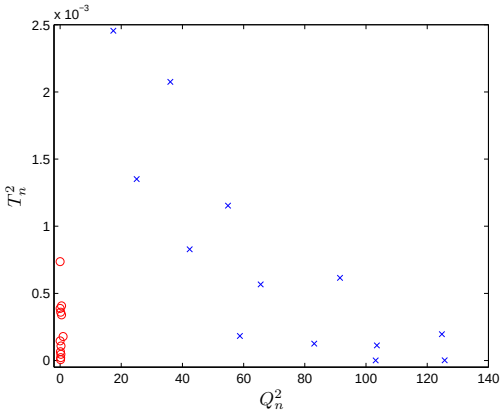
A esta nueva muestra se le asignará la clase con menos distancia o puede asignarse un umbral indicando que la nueva muestra no se encuentra estadísticamente próxima



(a) *Conjunto de datos y componentes principales de cada clase.*



(b) *Representación de los estadísticos normalizados usando el modelo 1*



(c) *Representación de los estadísticos normalizados usando el modelo 2*

Figura 2.16: Utilización de estadísticos normalizados de los modelos PCA.

a ningún modelo. Este método es ampliamente utilizado en los trabajos de lengua electrónica.

2.5.7. Partial Least Squares - Discriminant Analysis (PLS-DA)

Este método está basado en el conocido método de regresión lineal Partial Least Squares (PLS), descrito en profundidad en [Geladi86]. Se basa en suponer el problema de clasificación como un problema de regresión, de forma que a partir de unos datos de entrenamiento $\boldsymbol{X}$ podamos determinar la clase $\mathcal{C}_i$ a la que pertenece el vector $\boldsymbol{x}_i$. Para problemas de clasificación binarios, se puede asignar a una clase el valor cero y

a la otra clase el valor uno, de forma que el modelo, una vez entrenado, proporcione salidas que estarán próximas a cero para una clase y próximas a uno para la otra. En problemas de varias clases, se podría pensar en una estrategia de asignar un valor diferente para cada clase, de forma que proporcione una salida próxima a uno cuando $\mathcal{C}_i = 1$, una salida próxima a dos si $\mathcal{C}_i = 2$ y así con el resto de las clases. Sin embargo, esta estrategia no suele aplicarse en este tipo de métodos, ya que la clase dos no tiene por qué tener una relación intermedia entre la clase uno y la clase 3, sino que son asignaciones independientes. Por este motivo, se forma una matriz $\boldsymbol{Y}$ cuyos vectores columna $\boldsymbol{y}_j$ para cada clase j se formarán según:

$$\begin{aligned} y_{ij} &= 1 \quad \mathcal{C}_i = j \\ y_{ij} &= 0 \quad \mathcal{C}_i \neq j \end{aligned} \tag{2.42}$$

El método PLS tratará de buscar los factores que maximicen la covarianza entre la matriz $\boldsymbol{X}$ y la matriz $\boldsymbol{Y}$. Estos factores se denominan variables latentes (LV) y buscan maximizar la varianza de cada matriz y la correlación entre ambas matrices. Sin pérdida de generalidad, supondremos una matriz $\boldsymbol{X}$ de media nula. El método para encontrar las variables latentes LV_k es secuencial, siendo los pasos necesarios los siguientes:

1. Seleccionar el vector $\boldsymbol{u}$ como el vector de mayor varianza de la matriz $\boldsymbol{Y}$. Se calcula el vector de pesos
$$\boldsymbol{w}_k^s = \frac{\boldsymbol{X} \cdot \boldsymbol{u}}{\| \boldsymbol{X} \cdot \boldsymbol{u} \|} \tag{2.43}$$

2. Calcular las proyecciones de los vectores $\boldsymbol{x}_i$ sobre el vector $\boldsymbol{w}_k^s$, formando un vector $\boldsymbol{t}_k$ o *vector de scores*.
$$\boldsymbol{t}_k = \boldsymbol{X}^T \cdot \boldsymbol{w}_k^s \tag{2.44}$$

3. Con la información anterior, se calcula el vector $\boldsymbol{q}_k = \boldsymbol{Y} \cdot \boldsymbol{t}_k$.

4. Ahora se puede calcular la proyección de $\boldsymbol{Y}$ sobre el vector $\boldsymbol{q}_k$. Esta proyección formará un vector que trata de capturar la mayor varianza en $\boldsymbol{Y}$ correlada con $\boldsymbol{X}$. Se calcula
$$\boldsymbol{u}_k = \boldsymbol{Y}^T \boldsymbol{q}_k \qquad \boldsymbol{w}_k^f = \frac{\boldsymbol{X} \cdot \boldsymbol{u}_k}{\| \boldsymbol{X} \cdot \boldsymbol{u}_k \|} \tag{2.45}$$

5. Se comprueba la convergencia del algoritmo. Si $\| \boldsymbol{w}_k^f - \boldsymbol{w}_k^s \| < \varepsilon$ el vector $\boldsymbol{w}_k$ ha encontrado su final. Si no, se iguala $\boldsymbol{w}_k^s = \boldsymbol{w}_k^f$ y se repite desde el paso 2.

6. Se calcula el vector
$$\boldsymbol{p}_k = \frac{\boldsymbol{X} \cdot \boldsymbol{t}_k}{\| \boldsymbol{t}_k^T \boldsymbol{t}_k \|} \tag{2.46}$$

7. Si se quieren calcular más variables latentes deberá hacerse sobre las matrices sin la información detectada por la variable k. Para ello se hace:

$$\boldsymbol{X} = \boldsymbol{X} - \boldsymbol{p}\boldsymbol{t}^T \qquad \boldsymbol{Y} = \boldsymbol{Y} - \frac{\left(\boldsymbol{t}(\cdot\boldsymbol{Y}\cdot\boldsymbol{t})^T\right)^T}{\|\boldsymbol{t}_k^T\boldsymbol{t}_k\|} \tag{2.47}$$

Una vez formadas las variables latentes, se puede construir el modelo de regresión mediante la matriz $\boldsymbol{B}$, definida como:

$$\boldsymbol{B} = \boldsymbol{W}\left(\boldsymbol{P}^T\boldsymbol{W}\right)^{-1}\boldsymbol{Q}^T \tag{2.48}$$

donde $\boldsymbol{W}, \boldsymbol{Q}$ y $\boldsymbol{P}$ son las matrices obtenidas a partir de los vectores $\boldsymbol{w}_k$, $\boldsymbol{q}_k$ y $\boldsymbol{p}_k$ respectivamente. Una vez encontrado esta matriz de transformación, para conocer la clase correspondiente a una nueva muestra se proyectará como:

$$\boldsymbol{y}_i = \boldsymbol{B}^T\boldsymbol{x}_i \tag{2.49}$$

de forma que el vector $\boldsymbol{y}_i$ tendrá M componentes correspondientes a los valores para cada una de las clases. Como clase se asignará aquella que ha alcanzado un valor más alto. También se puede establecer un umbral para determinar si la muestra bajo análisis se determina como ruido. Debe notarse que, mediante la ecuación (2.49), se están estableciendo fronteras de clasificación lineales. De hecho los resultados obtenidos con este método son muy parecidos a los obtenidos con el método FLD como se demuestra en [Barker03].

2.5.8. Máquinas de Vectores Soporte

Como se mencionó en el primer capítulo, las máquinas de vectores soporte (SVM) [Vapnik98] tienen unas características que hacen que sean un método especialmente adecuado para aplicaciones de nariz y lengua electrónica. Entre estas características podemos destacar [Burges98]:

Generalización. Aunque el conjunto de entrenamiento disponga de pocas muestras, las máquinas de vectores soporte tratan de generalizar el problema, de forma que se puedan clasificar correctamente las muestras futuras. Esta característica es muy adecuada si se tiene en cuenta que los experimentos en el campo de la nariz y la lengua electrónica son costosos y por tanto, el número de muestras de los conjuntos de entrenamiento suele ser reducido.

Reducción. Una de las características de las SVM es que, una vez entrenadas, sólo es necesario guardar parte de la información del conjunto de entrenamiento, tratando de reducir tanto los requerimientos de memoria como el número de operaciones

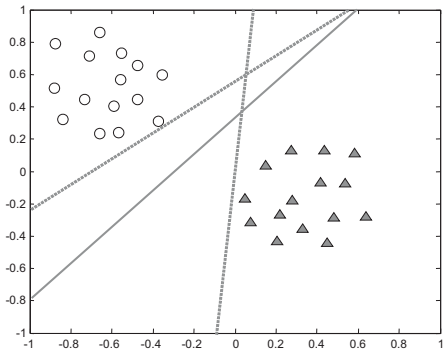


Figura 2.17: Posibles planos de separación de un problema de clasificación binario.

para la fase de test. Esta característica resulta muy atractiva cuando se pretende hacer sistemas en tiempo real. Esta propiedad, denominada en el campo de reconocimiento de patrones como *sparse* o baja densidad, se cumple en las SVM como se demuestra en [Bishop06].

Control de la complejidad. Los clasificadores descritos en esta sección pueden ser divididos en dos grandes grupos: aquellos que crean una frontera lineal y los basados en medidas no lineales. Los primeros responden muy bien a problemas sencillos de clasificación, creando fronteras lineales que serán las más adecuadas al estudiar problemas linealmente separables, mientras que los segundos pueden resolver problemas de fronteras mucho más complejas. Las SVM pueden pertenecer a ambos tipos, dependiendo del *kernel* seleccionado.

Dada la importancia que tienen para el campo de la nariz y la lengua electrónica, se profundizará en algunos aspectos de las SVM en capítulos posteriores, donde se proponen mejoras ajustadas a los campos de aplicación. En esta sección se describe parte de su base teórica para poder explicar mejor la comparación con otros métodos de clasificación.

Las SVM son sistemas de clasificación binarios basados en la teoría estadística del aprendizaje (SLT). Para poder exponer mejor su principio de funcionamiento nos centraremos en un problema de separación como el descrito en la figura (2.17). Existen diversos planos posibles de separación lineal para dicho conjunto. Entre ellos, los que formarían los clasificadores lineales anteriormente definidos en este capítulo. La pregunta que trata de resolver la SLT es cuál es el mejor plano de separación para futuras muestras sin presuponer ningún tipo de distribución de los datos de entrada. En este caso, el mejor plano será el que maximice el margen de separación. Así, en la figura (2.17) el mejor plano se ha dibujado con trazo continuo.

La idea reflejada anteriormente puede ser expresada de forma matemática suponiendo un conjunto de entrenamiento $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_l\}$ cuyas muestras tienen una

etiqueta binaria $y_i \in \{+1, -1\}$. La referida maximización del margen puede formularse como:

$$\begin{aligned} &\text{mín } \tfrac{1}{2}\langle \boldsymbol{\omega}, \boldsymbol{\omega} \rangle \\ &\text{sujeto a:} \\ &\quad y_i \left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b \right) \geq 1 \end{aligned} \tag{2.50}$$

Para poder resolver este problema, debemos acudir a la teoría de optimización con restricciones. En el caso de que las restricciones sean mayores o iguales que una cierta cantidad, las condiciones de Karush-Kuhn-Tucker (KKT) establecen que el problema puede ser resuelto mediante multiplicadores de Lagrange α_i y además, en nuestro caso, para cada una de las l condiciones impuestas se cumplirá

$$\alpha_i \left(y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b \right) - 1 \right) = 0 \tag{2.51}$$

De esta forma, podemos construir una función de optimización en su forma primaria

$$L\left(\boldsymbol{\omega}, b, \boldsymbol{\alpha}\right) = \frac{1}{2} \langle \boldsymbol{\omega}, \boldsymbol{\omega} \rangle - \sum_{i=1}^{l} \alpha_i \left(y_i \left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b \right) - 1 \right) + \sum_{i=1}^{l} \alpha_i \tag{2.52}$$

Los problemas de optimización con condiciones KKT son resueltos de forma más fácil mediante su problema dual, para lo que se obtiene las derivadas parciales de la ecuación (2.52) dando lugar a

$$\begin{aligned} &\frac{\partial L}{\partial \boldsymbol{\omega}} = 0 \rightarrow \boldsymbol{\omega} = \sum_{i=1}^{l} y_i \boldsymbol{\alpha}_i \boldsymbol{x}_i \\ &\frac{\partial L}{\partial \boldsymbol{\omega}} = 0 \rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0 \end{aligned} \tag{2.53}$$

Sustituyendo los valores de las derivadas en (2.52) obtenemos el problema dual

$$\begin{aligned} &\text{mín } W\left(\boldsymbol{\alpha}\right) = \tfrac{1}{2} \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha} \\ &\text{sujeto a:} \\ &\quad \alpha_i \geq 0 \\ &\quad \mathbf{y}^T \boldsymbol{\alpha} = 0 \\ &\quad y_i \left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b \right) \geq 1 \\ &\alpha_i \left(y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b \right) - 1 \right) = 0 \end{aligned} \tag{2.54}$$

donde $\mathbf{1}$ es un vector de unos y $\boldsymbol{Q}$ es una matriz simétrica $l \times l$, cuyos elementos son $Q_{i,j} = y_i y_j \langle x_i, x_j \rangle$. Es importante notar que la Hessiana de $W\left(\boldsymbol{\alpha}\right)$ es $\boldsymbol{Q}$ y en este caso se trata de una matriz definida positiva, por lo que la solución del problema de

optimización planteado en (2.54) es única. Volviendo a las condiciones KKT planteadas en (2.51), una vez encontrado el vector $\boldsymbol{\alpha}^*$ solución al problema anterior tendremos

$$\begin{aligned} \alpha_i^* &= 0, \quad \text{si} \quad y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b\right) > 1 \\ \alpha_i^* &> 0, \quad \text{si} \quad y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b\right) = 1 \end{aligned} \tag{2.55}$$

Combinando esta ecuación con (2.53) se constata que el hiperplano óptimo de separación se puede obtener como una combinación lineal dependiente únicamente de aquellos puntos $\boldsymbol{x}_i$ que se encuentran sobre las líneas de margen. Estos puntos de entrenamiento para los cuales los valores de los coeficientes de optimización son no nulos son los denominados vectores soporte.

En el planteamiento del problema anterior, se ha partido de que el conjunto de entrenamiento es linealmente separable. Sin embargo, existen múltiples ocasiones en la clasificación de señales procedentes de sensores de gases y líquidos en las que el conjunto de entrenamiento no es linealmente separable, debido fundamentalmente a la presencia de muestras aisladas de los grupos de su clase, definidas como *outliers*. Gráficamente esta situación puede observarse para un problema de dos dimensiones en la figura (2.18).

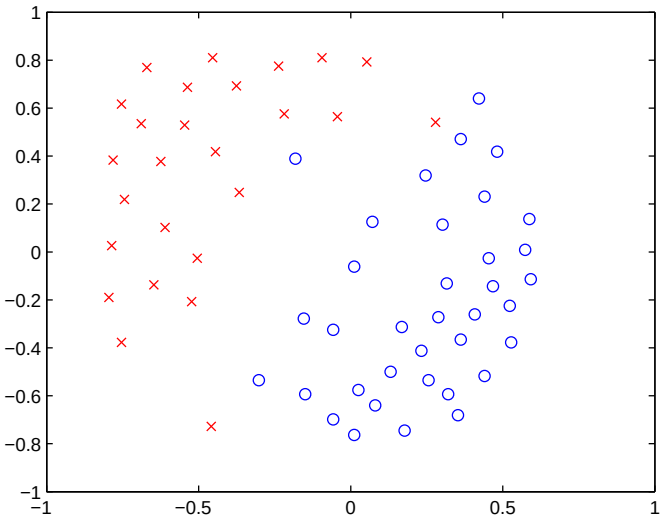


Figura 2.18: Problema en dos dimensiones no separable

Para poder resolver este tipo de problemas se introducen las denominadas variables de pérdidas ξ_i, replanteando las condiciones expuestas en (2.50) de la siguiente forma

$$y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b\right) \geq 1 - \xi_i \tag{2.56}$$

En la figura 2.19 se muestra el sentido físico de las variables de pérdidas. Claramente, permitiendo que el valor de los diferentes ξ_i sea suficientemente grande, cualquier hiperplano de separación cumplirá cada una de las l ecuaciones planteadas en 2.56. Sin embargo, esto nos separaría del objetivo de maximización del margen, por lo que debemos adoptar una solución de compromiso con las variables de pérdidas. La solución planteada en [Vapnik98] pasa por formar un vector con estas variables e introducir el mencionado vector como otro parámetro a optimizar en el problema primario descrito en la ecuación (2.52). La forma en la que se considera este vector de variables de pérdidas en el problema de optimización da lugar a diferentes formulaciones para resolver las SVM. La forma más extendida es mediante la regularización de la norma 1 del vector variables de pérdidas $\boldsymbol{\xi}$, dando origen a las denominadas L1-SVM, aunque generalmente son referidas como SVM. El problema de optimización para este tipo de máquinas resulta:

$$\begin{aligned} &\tfrac{1}{2}\langle \boldsymbol{\omega}, \boldsymbol{\omega}\rangle + C\sum_{i=1}^{l}\xi_i \\ &\text{sujeto a:} \\ &\qquad y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i\rangle + b\right) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0 \end{aligned} \tag{2.57}$$

donde C en una constante de regularización del problema fijada a priori. Así, formamos el nuevo Lagrangiano incluyendo las nuevas condiciones:

$$\begin{aligned} &L\left(\boldsymbol{\omega}, b, \boldsymbol{\alpha}\right) = \tfrac{1}{2}\langle \boldsymbol{\omega}, \boldsymbol{\omega}\rangle + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i\left(y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i\rangle + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{l}\mu_i\xi_i \\ &\text{sujeto a:} \quad y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i\rangle + b\right) \geq 1 - \xi_i \\ &\qquad \alpha_i\left(y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i\rangle + b\right) - 1 + \xi_i\right) = 0 \\ &\qquad \xi_i \geq 0 \\ &\qquad \alpha_i \geq 0 \\ &\qquad \mu_i \geq 0 \\ &\qquad \mu_i\xi_i = 0 \end{aligned} \tag{2.58}$$

donde los multiplicadores de Lagrange μ_i se introducen debido a la última condición impuesta en la expresión (2.57). El Lagrangiano de la expresión (2.58) puede derivarse para obtener el problema dual, al igual que se hizo en el caso separable. Las derivadas parciales respecto al vector $\boldsymbol{\omega}$ y b dan unos resultados muy similares a los planteados en la ecuación (2.53). Además, debemos considerar la derivada parcial respecto al vector de pérdidas:

$$\frac{\partial L}{\partial \boldsymbol{\xi}} = 0 \;\rightarrow C - \alpha_i - \mu_i = 0 \tag{2.59}$$

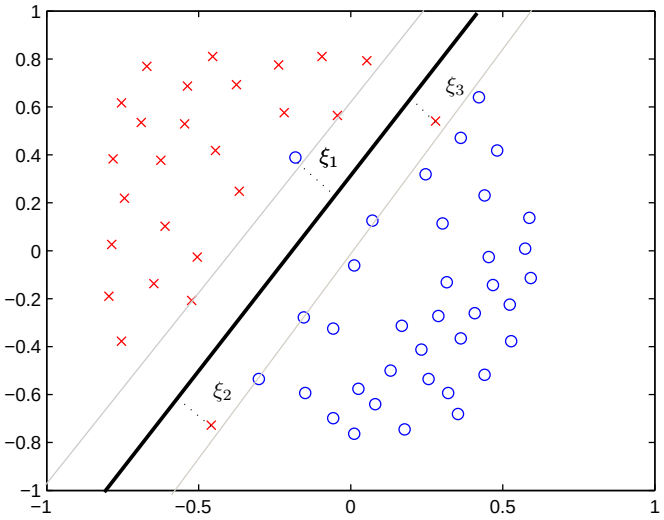


Figura 2.19: Variables de pérdidas en un problema no separable

Por tanto, si unimos el resultado de esta expresión con la última condición KKT planteada en (2.58) obtendremos que cuando las variables de pérdidas sean nulas, $\alpha_i = C - \mu_i < C$. Solo en el caso de que una muestra tenga variables de pérdidas no nulas tendremos $\mu_i = 0$ y por tanto $\alpha_i = C$. Sustituyendo de nuevo las expresiones obtenidas de las derivadas, se encuentra en forma matricial el siguiente problema de optimización

$$\begin{aligned} &\min W(\boldsymbol{\alpha}) = \tfrac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q}\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha} \\ &\text{sujeto a:} \\ &\quad 0 \le \alpha_i \le C \\ &\quad \boldsymbol{y}^T\boldsymbol{\alpha} = 0 \end{aligned} \tag{2.60}$$

La solución del vector $\boldsymbol{\alpha}^*$ de la expresión (2.60) clasifica los patrones de entrada en tres tipos diferentes:

$$\begin{aligned} &\alpha_i^* = 0, \quad \text{si} \quad y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i\rangle + b\right) > 1 \\ &0 < \alpha_i^* < C, \quad \text{si} \quad y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i\rangle + b\right) = 1 \\ &\alpha_i^* = C, \quad \text{si} \quad y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i\rangle + b\right) < 1 \end{aligned} \tag{2.61}$$

En el primer caso, los patrones de entrenamiento son correctamente clasificados $\alpha_i = 0$ y el resultado del entrenamiento hubiera sido el mismo si no se hubieran tenido en cuenta. Para el caso de los patrones que caen justo en las fronteras del margen, se tendrá $0 < \alpha_i < C$ y son los denominados vectores soporte de tipo 1. Por último, aquellos patrones que caen dentro del margen o quedan mal clasificados tendrán un

multiplicador de Lagrange $\alpha_i = C$ y se denominarán vectores soporte de tipo 2. Al conjunto de muestras de entrenamiento cuyo $\alpha_i > 0$ se les denomina de forma genérica vectores soporte.

Como se ha expuesto, la regularización del vector de variables de pérdidas $\boldsymbol{\xi}$ en la expresión (2.54), puede dar lugar a otro tipo de planteamiento en las SVM. Así, si se considera la norma 2 el problema se reformula como:

$$\begin{aligned} &\tfrac{1}{2}\langle \boldsymbol{\omega}, \boldsymbol{\omega} \rangle + \tfrac{1}{2} C \sum_{i=1}^{l} {\xi_i}^2 \\ &\text{sujeto a:} \\ &\qquad y_i \left(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b \right) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0 \end{aligned} \tag{2.62}$$

dando lugar a las denominadas L2-SVM, que proporcionan resultados similares a las L1-SVM pero con un número de vectores soporte superior, lo que se traducirá en un mayor tiempo de cómputo [Abe05]. En esta tesis se estudiará en capítulos posteriores una modificación de estas expresiones para obtener otro tipo de clasificadores.

Así, una vez realizado el entrenamiento se encontrarán los valores de los multiplicadores de Lagrange α_i y del parámetro de desplazamiento b. Una vez obtenidos estos valores, se contará con una función de decisión para una nueva muestra $\boldsymbol{x}_i$ como:

$$f(\boldsymbol{x}_i) = \operatorname{sign}\left(\sum_{j=1}^{N_{sv}} \alpha_j y_j \langle \boldsymbol{s}_j, \boldsymbol{x}_i \rangle + b \right) \tag{2.63}$$

donde N_{sv} es el número de vectores soporte o muestras de entrenamiento cuyos $\alpha_i > 0$.

La teoría de las SVM de separación lineal se extiende a casos no lineales, suponiendo una transformación de los datos hacia un espacio donde podrán ser separados más fácilmente. Dicha transformación no es conocida usualmente, siendo necesario tan sólo conocer el producto escalar en el espacio transformado. Se define una función *kernel* como una función en el espacio de entrada capaz de calcular el producto escalar en un espacio de Hilbert transformado sin necesidad de transformar los datos de entrada:

$$\begin{aligned} &\kappa(\boldsymbol{x}, \boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle \\ &\kappa(\boldsymbol{x}, \boldsymbol{y}) = \kappa(\boldsymbol{y}, \boldsymbol{x}) \rightarrow \mathbb{R} \end{aligned} \tag{2.64}$$

La expresión anterior es conocida a menudo como truco del kernel, ya que no es necesario conocer cómo se transforman los datos al nuevo espacio si la única información que se desea conocer es el producto escalar.

Así, la función de decisión quedará como:

$$f(\boldsymbol{x}_i) = \operatorname{sign}\left(\sum_{j=1}^{l} \alpha_j y_j \kappa(\boldsymbol{s}_j, \boldsymbol{x}_i) + b \right) \tag{2.65}$$

La ecuación (2.65) coincide con la ecuación (2.63) cuando se supone un kernel lineal según la función:

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle \tag{2.66}$$

Entre los kernels no lineales más populares se encuentra el gaussiano, también denominado algunas veces como RBF por ser similar al descrito en las redes de base radial:

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\gamma \parallel \boldsymbol{x} - \boldsymbol{y} \parallel\right)^2 \tag{2.67}$$

El problema de optimización utilizando los kernels es el mismo que el expuesto en (2.60) teniendo presente que la matriz $\boldsymbol{Q}$ tiene elementos $Q_{i,j} = y_i y_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Si el kernel seleccionado cumple el teorema de Mercer:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0 \ \forall c_i, c_j \in \mathbb{R} \tag{2.68}$$

la matriz $\boldsymbol{Q}$ sigue siendo definida positiva por lo que el problema de optimización es también de solución única. De esta forma, se está procediendo a realizar la maximización regulada del margen en el espacio de Hilbert transformado, del que solo es necesario conocer su producto interno a través de la función kernel.

2.5.9. Random forest

En [Pardo07] se describe por primera vez el uso del método de clasificación conocido como *random forest* aplicado a la nariz electrónica. Este método, cuyo tutorial puede encontrarse en [Breiman01], se basa en la construcción de un grupo de árboles de clasificación de tipo CART (*Classification and Regression Tree*) [Duda00]. Una vez construidos los árboles, ante una nueva muestra, cada uno de ellos votará por una clase como clase candidata a la que pertenece la nueva muestra. Finalmente, se recuenta el número de votos y la clase que haya obtenido una mayoría es considerada como clase a asignar a la nueva muestra. Por tanto, random forest es una técnica de tipo *boosting*, la cuál se basa en construir clasificadores débiles (weak classifiers) de forma que por votación se consigue un clasificador robusto. La construcción de cada árbol se basa en tres premisas fundamentales:

- Cada árbol se forma con un subconjunto bootstrap del conjunto de entrenamiento.
- Si existen d características del conjunto de entrenamiento, sólo se consideran $mty << d$ para realizar la mejor división de cada nodo. Estas mty características son seleccionadas al azar.

- Cada árbol se hace crecer hasta el final, sin que exista algoritmo de poda alguno.

Una de las características claves del método son los datos *out-of-bag* o datos que no han sido seleccionados en ningún subconjunto bootstrap. Una vez que se han construido todos los árboles, se testea con las muestras out-of-bag de forma que se obtiene un estimador insesgado del error. En este proceso se contará el número de votos correctos que ha obtenido cada muestra. Una vez realizado este proceso, para cada muestra se cambia aleatoriamente su valor y se obtiene el porcentaje de votos correctos de los datos con la permuta realizada. A cada característica se le asigna un ranking entre el número de aciertos con los datos out-of-bag originales y los aciertos con los mismos datos pero con esa característica permutada. De esta forma se obtiene una relación de características importante. Después de esta primera selección se puede volver a construir el bosque con las características más importantes. Esto hace que el método random forest tenga su propia selección de características.

Entre las características que hacen atractivo este método se encuentra el número de parámetros a ajustar, solo el valor *mty* y el número de árboles del bosque, no siendo críticos entre ciertos niveles. Además, al utilizar subconjuntos bootstrap, no sufre el problema de sobreaprendizaje, aprovechando todos los datos disponibles para proporcionar tanto el modelo entrenado como la estimación del error y un ranking de características.

Capítulo 3

Extracción de Parámetros en Señales Dinámicas

Las señales dinámicas proporcionadas por los sensores de gases y líquidos pueden ser utilizadas directamente para su clasificación, pero esto implica trabajar con clasificadores con una elevada dimensión. La estrategia de la etapa de preprocesado, consiste en obtener información de las señales dinámicas mediante una serie de procedimientos con un doble objetivo. Por un lado es una forma de realizar clasificadores menos complejos, al trabajar con un menor número de características, mientras que el segundo objetivo trata de proporcionar un mejor entendimiento de las señales bajo estudio. Al describir en el capítulo anterior los métodos encontrados en la bibliografía que tratan de cubrir estos objetivos, se incluyó una pequeña discusión sobre las ventajas e inconvenientes de cada uno de ellos. En el campo de la nariz y la lengua electrónica, además de los tradicionales métodos empleados en el campo del reconocimiento de patrones como son PCA y LDA, el uso de la transformada wavelet destaca entre los métodos más empleados en la bibliografía como método de extracción de parámetros característicos de la señal. Sin embargo, queda abierto cómo seleccionar los coeficientes wavelet que formarán la entrada del clasificador y si el uso de las wavelet ortogonales son la mejor elección para todas las aplicaciones basadas en sensores de gases y líquidos.

El uso de wavelet packets es bien conocido en el campo de compresión de señales, por lo que será explorado en este capítulo. Además de los interrogantes que quedaban abiertos para el uso de la transformada wavelet, se debe encontrar cuál es la mejor descomposición común para todas las señales, por lo que en este capítulo se plantean algoritmos de selección de la descomposición y los coeficientes.

Además de profundizar en la línea anterior, en este capítulo se plantea como aportación de esta tesis un nuevo método basado en la regresión realizada en espacios transformados por medio de kernels para obtener los parámetros más importantes de la señal. Este método se expondrá posteriormente como *Kernel Fixed Regression*. El

capítulo se completa con la aplicación de los métodos propuestos sobre un conjunto variado de datos procedentes de sistemas de nariz y lengua electrónica. Como resultado de esta aplicación se obtienen interesantes conclusiones sobre los métodos propuestos. Finalmente, ambos métodos son comparados con varios de los métodos expuestos en el capítulo anterior.

3.1. Ampliación del uso de la transformada wavelet

En esta parte se expondrá cómo aplicar algunas de las técnicas de la transformada wavelet, que son ampliamente conocidas en el campo de la compresión, al problema de la extracción de características.

3.1.1. La extensión periódica y simétrica

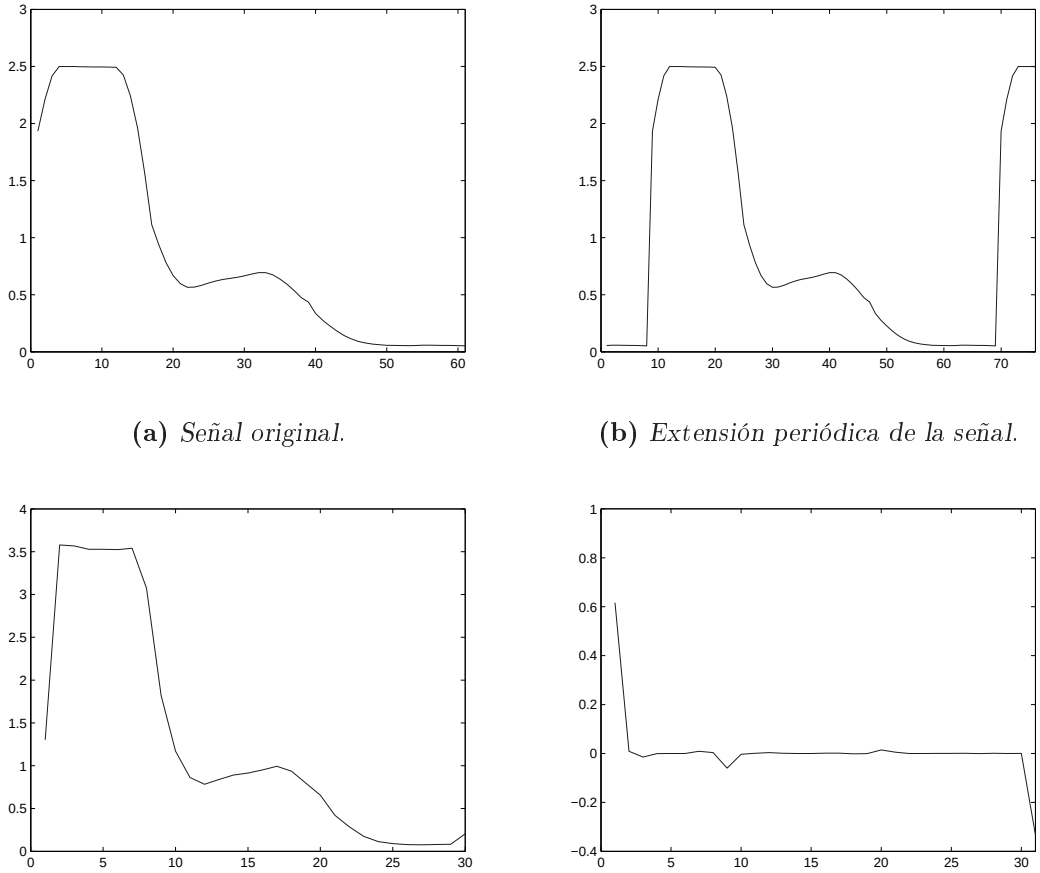
Según se expuso en el capítulo anterior, la transformada wavelet puede ser iterada varias veces para conseguir que la mayor energía se concentre en unos pocos coeficientes. Sin embargo, partiendo de una señal de coeficientes $c_0[n]$ de longitud N y de filtros wavelet con unas longitudes L y G, correspondientes al paso alto y paso bajo, en cada descomposición tendremos las siguientes señales:

$$\begin{array}{ll} c_1[n] & n = 0,1,2,\ldots (L+N-1)/2 \\ d_1[n] & n = 0,1,2,\ldots (L+G-1)/2 \end{array} \tag{3.1}$$

A medida que crece el orden de los filtros empleados y la profundidad de descomposición, el número total de coeficientes wavelet se incrementa de forma significativa. Parte de esos coeficientes solo están relacionados con la convolución de la señal cuando ésta ya se ha anulado, por lo que no proporcionan información tan útil si el objetivo que pretendemos es extraer información para la clasificación. Además, complican en exceso los algoritmos al tener que considerar las distintas longitudes de las señales que van quedando. Una forma habitual de solventar este problema es realizar la extensión periódica de la señal [Proakis96]. Este método tiene en cuenta que si a la entrada de un sistema tenemos una señal periódica de periodo P la salida del sistema filtrado también será periódica con periodo P. Para nuestros propósitos, basta extender la señal tanto al comienzo como al final en $(L-1)/2$ muestras o extender únicamente al principio la señal con $(L-1)$ muestras.

Sin embargo, como se destaca en [Strang96], en aquellas señales que no finalicen en valores muy próximos a los que se empieza, la extensión simétrica generará componentes importantes de alta frecuencia. Este hecho, se puede observar en la figura (3.1), donde se parte de una señal obtenida por espectrofotometría UV-VIS, representada en la figura (3.2(a)), obtenida a partir de un vino tinto. A dicha señal se le ha aplicado

la extensión periódica necesaria que se muestra en la figura (3.1(b)). En las figuras (3.1(c)) y (3.1(d)) se representa la descomposición wavelet en su aproximación y los detalles respectivamente. Se puede apreciar cómo al realizar la extensión periódica se



(a) *Señal original.*

(b) *Extensión periódica de la señal.*

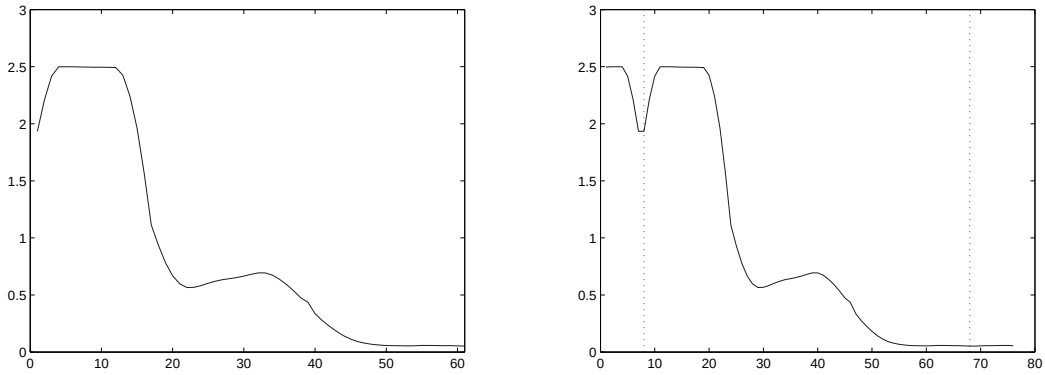
(c) *Coeficientes de la aproximación.*

(d) *Coeficientes de los detalles.*

Figura 3.1: Efecto de la extensión periódica sobre señal obtenida por espectrometría UV-VIS.

ha introducido una componente de alta frecuencia que no forma parte de la información característica de la señal. Este tipo de situaciones sucede en las señales objeto de interés de esta tesis cuando se produce una cierta histéresis en las señales proporcionadas de los sensores y el tiempo de volver a la posición inicial es excesivamente grande. También se produce este fenómeno en el análisis de señales procedentes de espectrofotometría UV-VIS, como el mostrado en la figura (3.1), pues la absorbancia de las regiones de longitudes de onda inicial y final no tienen por qué coincidir. La forma de abordar este problema también ha sido ampliamente estudiada en el campo de la compresión mediante una extensión periódica y simétrica. Esta forma de extender la señal puede

ser apreciada en la figura (3.2). Si bien se reducen las componentes de alta frecuencia, la extensión realizada sobre una señal de longitud m tendrá un periodo $2m$. Sin embargo, si el filtro utilizado es de fase mínima y además tiene simetría par en el dominio del tiempo, la salida será también simétrica, necesitando solo m coeficientes para describir la señal de salida.



(a) *Señal original.* **(b)** *Reflexión periódica de la señal..*

Figura 3.2: Reflexión periódica para uso con wavelets biortogonales.

Sin embargo, a excepción de los filtros basados en la wavelet de Haar, no existen filtros ortogonales de fase mínima. Para solventar este problema debemos ir a soluciones de filtros wavelet biortogonales, tal como se describe en [Mallat99]. Las condiciones que deben cumplir los filtros biortogonales para conseguir reconstrucción perfecta son:

$$\begin{aligned} &\sum_{k=-\infty}^{\infty} h[k]\,\tilde{h}[k-2l] = \delta_l \\ &\sum_{k=-\infty}^{\infty} h[k] = 1 \\ &\sum_{k=-\infty}^{\infty} \tilde{h}[k] = 1 \\ &\tilde{g}[n] = (-1)^n h[1-n] \\ &g[n] = (-1)^n \tilde{h}[1-n] \end{aligned} \tag{3.2}$$

En esta tesis se ha optado por probar con filtros biortogonales diseñados en el dominio de la frecuencia como los propuestos en [Cohen95].

3.1.2. La descomposición wavelet packet

La descomposición wavelet packet es una generalización de la descomposición wavelet clásica. Difiere de esta última a partir del segundo nivel de profundidad de la descomposición. En este caso se considera también la posibilidad de realizar un análisis

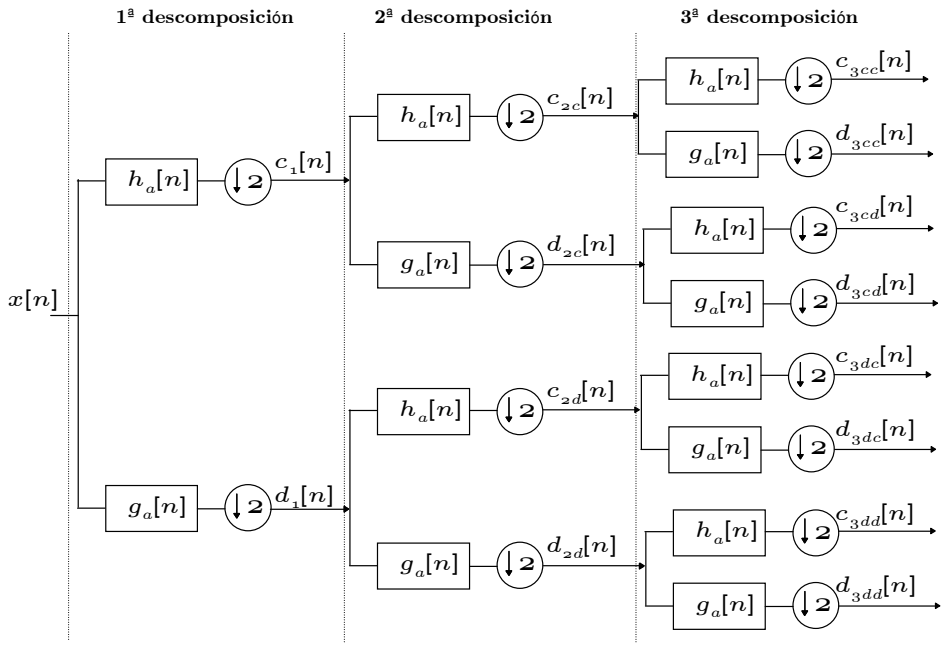


Figura 3.3: Árbol completo de descomposición wavelet packet.

de la señal de detalles $d_1[n]$. El árbol completo para tres niveles de descomposición se muestra en la figura (3.3). La pregunta inmediata al observar el árbol completo de decisión es si es necesario descomponer siempre todas las señales obtenidas en los pasos previos. Surge así la idea del mejor árbol de descomposición, guiados por un criterio de decisión. El criterio más extendido para realizar la división está basado en la entropía [Coifman92], definida para una señal genérica $a[n]$ según:

$$H(a[n]) = -\sum_n a^2[n] \log\left(a^2[n]\right) \tag{3.3}$$

Con este criterio, se calcula la entropía asociada a cada una de las señales de la descomposición. Si la suma de las entropías de los descendientes de un nodo del árbol wavelet packet es menor que la entropía de la señal originada en dicho nodo, esta descomposición se mantiene en el árbol. Esta idea se muestra en la figura (3.4) donde se ha dibujado en línea continua el árbol de descomposición definitivo y en línea discontinua aquellas posibles divisiones que no forman parte del árbol por no cumplir el criterio dado. Se representan en rojo las entropías de aquellos descendientes de un nodo cuya suma supera la entropía del nodo origen. Existen otros criterios para guiar el árbol de descomposición como se describe en [Wickerhauser94], pero todos consideran la información de cada una de las señales bajo estudio. En el campo de la nariz y lengua electrónica, este método ha sido utilizado en algunos trabajos [Panigrahi08], [Panigrahi05] de forma muy preliminar. Sin embargo, los autores de estos trabajos consideran la descomposición wavelet packet basada en la entropía para cada una de las

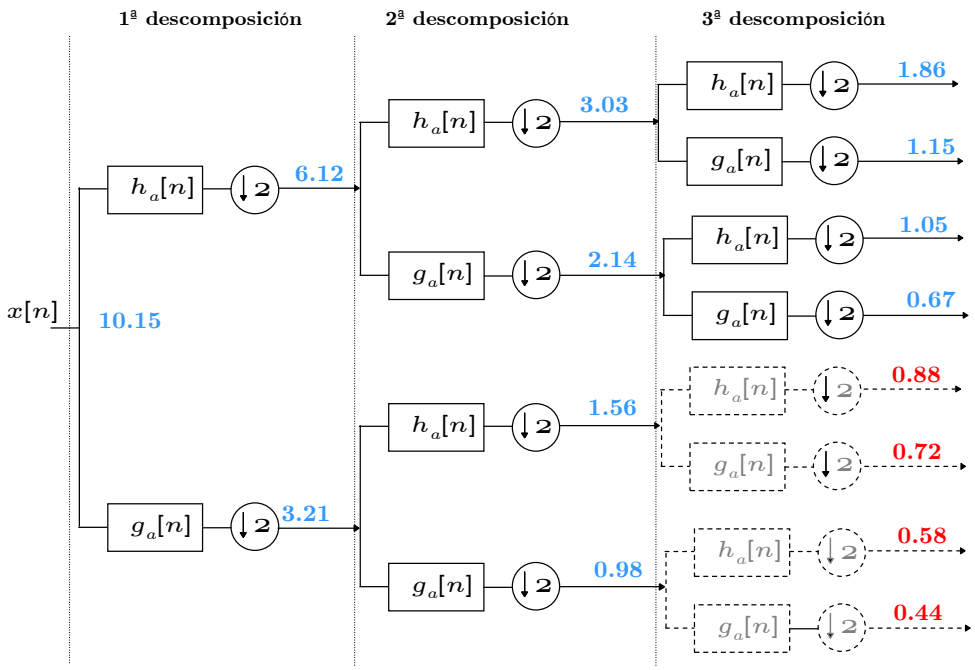


Figura 3.4: Descomposición wavelet packet con criterio entropía.

señales. Esto presenta un claro problema de cara a la clasificación, dado que una misma característica de patrones diferentes hace referencia a coeficientes distintos.

3.1.3. Algoritmo Propuesto

Como aportación original de esta tesis se propone un algoritmo basado en la transformada wavelet que presenta las siguientes ventajas:

- Aprovecha las ventajas de la extensión periódica y la extensión simétrica utilizando en este último caso filtros biortogonales.
- El algoritmo establece el árbol de descomposición wavelet packet partiendo de un conjunto de entrenamiento, de forma que para las futuras muestras de test, la división wavelet packet estará impuesta por la división óptima encontrada en el conjunto de entrenamiento.
- Se realiza una selección de las características de forma automática.

Para la aplicación del algoritmo se debe seleccionar un conjunto de entrenamiento y uno de test. Aunque en la sección de resultados se detalla con mayor profundidad cómo se realizan los experimentos, para la mejor comprensión del algoritmo, es necesario exponer que para un experimento concreto, el conjunto de entrenamiento se selecciona

considerando un quinto de las muestras totales. Esta selección de las señales de entrenamiento se realiza de forma aleatoria pero con una supervisión que comprueba que la distribución de clases se mantiene respecto a la del conjunto original.

Antes de comenzar, se seleccionará el máximo nivel de descomposición. Una vez seleccionado dicho nivel se debe seleccionar la mejor base de descomposición wavelet packet, para lo cual se siguen los pasos:

1. Realizar el árbol completo de descomposición. Para un nivel de profundidad 3, sería equivalente a seguir el esquema planteado en la figura (3.3). Para realizar esta descomposición, se debe tener en cuenta:

   - Si se trabaja con señales cuyo final es similar al comienzo, se usará la extensión periódica y filtros ortogonales. Dentro de este grupo de señales se encuentran las procedentes de nariz y lengua electrónica en sistemas modulados en flujo.
   - Si se trabaja con señales cuyo inicio y final no sean similares, se utilizará la reflexión simétrica y filtros biortogonales. Dentro de este grupo se encontrarían las señales de nariz electrónica procedentes de modular en temperatura, y en el campo de lengua electrónica las señales procedentes de realizar una espectrofotometría y los ciclovoltiamperogramas.

2. Calcular la media de la entropía, definida según la ecuación (3.3), de los descendientes pertenecientes al mismo nodo del árbol wavelet packet.

3. Si la media de las entropías de dos descendientes de un nodo es superior a la medida de las entropías de dicho nodo la descomposición no tiene lugar.

4. Se guarda el árbol de descomposición a aplicar a futuras señales de test. Se almacena la información procedente de las señales de entrenamiento $x_i[n]$, concatenando los coeficientes resultantes de los nodos terminales de árbol de descomposición, creando una nueva señal transformada $p_i[k]$.

Es importante destacar la diferencia que existe entre el algoritmo propuesto en esta tesis y el método tradicional de descomposición del árbol wavelet packet. En este último se considera la descomposición atendiendo exclusivamente a cada señal, mientras que en nuestro problema debemos pensar que la descomposición debe ser la misma para todas las señales, motivo por el que se trabaja con las medias de las entropías. Una vez que se tiene el conjunto $\boldsymbol{P}$ que contiene a todas las señales $p_i[k]$ hay que seleccionar aquellos coeficientes de cada señal que formarán la información de entrada del clasificador. Al igual que en el caso de la descomposición, la selección de dichos coeficientes debe realizarse de una forma global para todas las señales, de manera que el clasificador esté

trabajando con las mismas variables para todas las señales. Se proponen dos criterios para la selección de dichos coeficientes:

Selección por energía En este caso se seleccionan los d coeficientes con una media de mayor energía. Esta estrategia establece un valor asociado a cada coeficiente wavelet j según:

$$J_j = \sum_{i=1}^{l} p_i^2 [j] \tag{3.4}$$

donde l es el número de señales del conjunto de entrenamiento.

Selección por separabilidad de clases Si bien la selección anterior consigue quedarse con aquellos coeficientes que, de media, aportan más información a la señal, en el campo del reconocimiento de patrones está muy extendida la idea según la cuál la selección debe hacerse entre aquellas características, coeficientes en nuestro caso, para las que existe una mayor separación entre una clase y el resto. Así, para un problema de clasificación binaria, la función que establece el valor de un coeficiente wavelet j sería:

$$J_j = \frac{\left(\mu_j^+ - \mu_j^-\right)^2}{\left(\sigma_j^+\right)^2 + \left(\sigma_j^-\right)^2} \tag{3.5}$$

donde $\left(\sigma_j^+\right)^2, \left(\sigma_j^-\right)^2$ es la varianza de los coeficientes j dentro de las clases positiva y negativa respectivamente, mientras que μ_j^+ y μ_j^- son las medias de los coeficientes j de dichas clases. Al igual que en el caso anterior, se seleccionan los d coeficientes con mayor valor de la función criterio.

Para problemas multiclase se ha optado por una estrategia uno contra todos. Así, en esta etapa de preprocesado se comenzará con esta estrategia, generando un modelo de preprocesado diferente por cada clase. Esto significa que se divide el conjunto de entrenamiento entre la clase bajo estudio y el resto de señales y se ejecuta el algoritmo propuesto, repitiendo este proceso para cada una de las clases del problema. En la etapa de test, cada muestra será procesada por tantos modelos diferentes como clases existan en el problema bajo estudio.

Aunque la transformada wavelet ya había sido aplicada en el campo de la nariz electrónica, la aplicación de filtros biortogonales en este tipo de sistemas y el algoritmo propuesto de descomposición y selección de coeficientes wavelet constituyen una aportación original de esta tesis. Estas ideas fueron parcialmente publicadas en [Acevedo05].

3.2. Extracción mediante fixed kernel ridge regression

Supongamos que tuviéramos una serie de señales procedentes de un sensor con información dinámica, como las mostradas en la figura (3.5), donde cada color representa una sustancia. Una forma rápida de clasificar estas señales sería suponerlas lineales y obtener la pendiente y el punto de corte con el eje de ordenadas de cada recta. Así, cada señal puede ser caracterizada mediante:

$$\boldsymbol{y}_i(x) \rightarrow f_i(x) = x\omega_i + b \tag{3.6}$$

donde x es la variable independiente de la señal dinámica de un sensor, que como se vio en el capítulo anterior puede ser el tiempo, tensión o longitud de onda. En el ejemplo de la figura (3.5) esta variable independiente x es el tiempo. De esta forma, las rectas

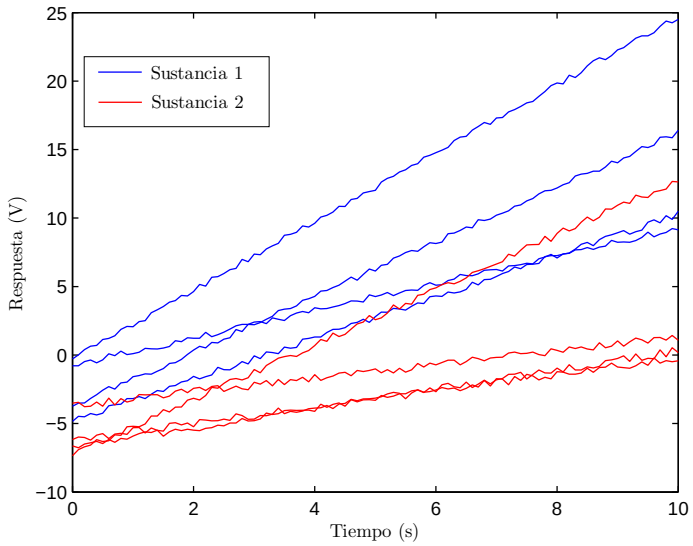


Figura 3.5: Señales con forma casi lineal.

asociadas a las señales de nuestro ejemplo marcadas en azul tienen por lo general un menor punto de corte con el eje de ordenadas b_i y una mayor pendiente ω_i que las rectas asociadas a las señales rojas. Este hecho se representa en la figura (3.6), de forma que habríamos conseguido definir nuestras señales por medio de dos parámetros, pendiente y punto de corte con el eje de ordenadas, que pueden ser utilizados como componentes de los vectores entrada del sistema de reconocimiento.

En la práctica, las señales no se asemejan a las rectas del ejemplo anterior, pero nos sirve como introducción a la parametrización de señales para su clasificación. Suponiendo que podemos aproximar las señales obtenidas de los sensores a una forma dada, en el ejemplo anterior una recta, basta con determinar los parámetros de la función

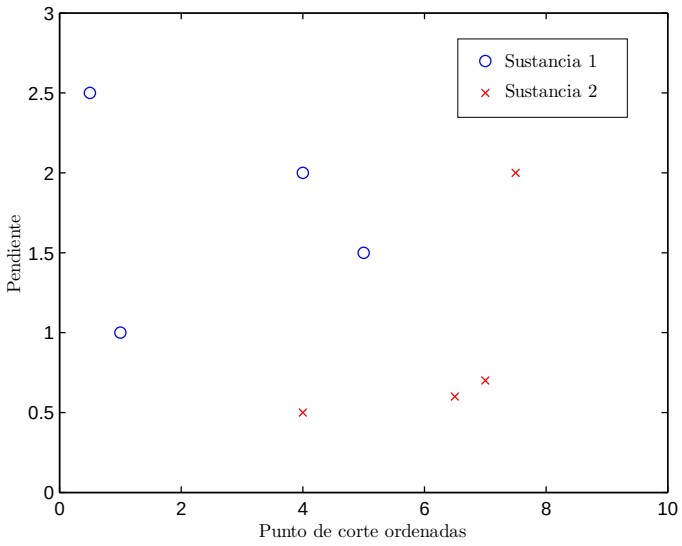


Figura 3.6: Representación del punto de corte del eje de ordenadas contra la pendiente de señales linealizadas.

para que éstos sirvan de entrada al clasificador, tal y como se hace con la pendiente y el punto de corte con el eje de ordenadas en el caso descrito.

La parametrización de una señal puede ser hecha mediante una estrategia de conocimiento previo de la expresión matemática. Esta estrategia es la que se ha aplicado en la ecuación (3.6) para las rectas o, como se describe en el anterior capítulo, suponiendo sumas de exponenciales cuando se modula por flujo en los sensores de gas. En este último caso, suponíamos que la señal podía aproximarse por N exponenciales de forma que los parámetros que se extraían eran las N amplitudes asociadas y las N constantes de tiempo.

Sin embargo, la estrategia anterior requiere que se presuponga la forma de las señales, lo que no es viable para muchos tipos de señales procedentes de los sistemas de sensores descritos en el anterior capítulo. Existe la posibilidad de aproximar una señal obtenida a una función de la variable independiente como un desarrollo de funciones base, que en este caso denominaremos kernel. Así, una señal $\boldsymbol{y}_i$ que contiene la información dinámica de un sensor puede aproximarse mediante:

$$\boldsymbol{y}_i(x) \simeq f_i(x) = \sum_{j=1}^{N} \alpha_{ij} \boldsymbol{\Phi}_j(x) \tag{3.7}$$

La obtención de los parámetros $\boldsymbol{\alpha}_i$ puede ser resuelta mediante técnicas de regresión. La aportación de esta tesis es utilizar los coeficientes $\boldsymbol{\alpha}_i$ que aproximan la señal como componentes del vector de entrada del sistema de clasificación. Así, nos centraremos

en los métodos de regresión kernel, ya que suponen que la señal de entrada puede ser transformada en un espacio de Hilbert transformado donde puede ser aproximada a una recta y por tanto el ejemplo que se ha planteado a modo de introducción es válido. Entre las técnicas de regresión kernel podemos destacar la denominada *kernel ridge regression* que aproxima una señal a una extensión de funciones kernel, como se plantea en la ecuación (3.7). El problema de este método es que el número de parámetros d obtenidos es igual a la longitud de la señal, por lo que no podría ser utilizado para nuestros propósitos. Para solventar este problema se propone el uso de la denominada *fixed kernel ridge regression*, en la que el número de parámetros d se establece a priori.

En esta sección, se describe en primer lugar la regresión lineal para poder entender el método kernel ridge regression. Aunque, como se ha comentado anteriormente, este método no es práctico para extraer parámetros de cara a una clasificación, se describe para facilitar la comprensión del método fixed kernel ridge regression, que será expuesto a continuación. El objetivo final para los propósitos de esta tesis es contar con un método de regresión con un número de parámetros fijo y reducido que nos sirva como método para extraer parámetros que caractericen la señal.

3.2.1. Kernel ridge regression

El método conocido como kernel ridge regression puede considerarse como la versión no lineal del método ridge regression que trata de aproximar una señal de entrada a una función lineal, como se describe en la ecuación (3.6). Con el objetivo de explicar los métodos kernel en la regresión, se expone a continuación cómo formular la regresión lineal mediante productos escalares. La ecuación de la recta estimada en la ecuación (3.6) puede escribirse en forma discreta según:

$$f[x] = [x, 1]\begin{bmatrix} \omega \\ b \end{bmatrix} = \tilde{\boldsymbol{x}}^T\tilde{\boldsymbol{\omega}} \tag{3.8}$$

donde $\tilde{\boldsymbol{x}} = [\boldsymbol{x}, 1]$ es el vector variable independiente extendida y $\tilde{\boldsymbol{\omega}} = [\boldsymbol{\omega}, b]$ es el vector solución del problema de regresión lineal. Supongamos que la señal del sensor es observada en m puntos de la variable independiente podemos construir la matriz de valores independientes extendidos $\tilde{\boldsymbol{X}}_\tau = [\tilde{\boldsymbol{x}}_{\tau 1}, \tilde{\boldsymbol{x}}_{\tau 2}, ...\tilde{\boldsymbol{x}}_{\tau m}]^T$, de forma que la matriz $\tilde{\boldsymbol{X}}_\tau$ tendrá dimensiones $m \times 2$. Por otro lado, tendremos el vector de respuestas observadas a esos valores de la variable independiente $\boldsymbol{y} = [y_1, y_2, ...y_m]$. Se estima el vector solución $\tilde{\boldsymbol{\omega}}$ como:

$$\tilde{\boldsymbol{\omega}} = \tilde{\boldsymbol{X}}_\tau^\dagger \boldsymbol{y} \tag{3.9}$$

donde la matriz $\tilde{\boldsymbol{X}}_\tau^\dagger$ es la pseudo-inversa de la matriz de valores independientes extendidos $\tilde{\boldsymbol{X}}_\tau$. Dado que la pseudo-inversa no siempre existe, se añade un factor δ, de forma que su cálculo vendrá dado por:

$$\boldsymbol{X}_\tau^\dagger = \boldsymbol{X}_\tau^T \left(\boldsymbol{X}_\tau \boldsymbol{X}_\tau^T + \delta \boldsymbol{I} \right)^{-1} \tag{3.10}$$

Por tanto, sustituyendo la expresión de la ecuación (3.9) en la ecuación (3.8) obtenemos que la función de regresión se puede expresar como:

$$f\left[x\right] = \tilde{\boldsymbol{x}}.^T \boldsymbol{X}_\tau^T \left(\boldsymbol{X}_\tau \boldsymbol{X}_\tau^T + \delta \boldsymbol{I} \right)^{-1} \boldsymbol{y} \tag{3.11}$$

Definimos la matriz

$$\boldsymbol{K} = \boldsymbol{X}_\tau \boldsymbol{X}_\tau^T \tag{3.12}$$

de forma que cada elemento $K_{ij} = \langle \tilde{\boldsymbol{x}}_{\tau i}, \tilde{\boldsymbol{x}}_{\tau j} \rangle$ puede expresarse como un producto escalar. Ahora definiremos el vector $\boldsymbol{\alpha}$ como:

$$\boldsymbol{\alpha} = \left(\boldsymbol{K} + \delta \boldsymbol{I} \right)^{-1} \boldsymbol{y} \tag{3.13}$$

Podemos incluir la expresión anterior en la ecuación (3.11), de forma que para cualquier valor de la variable independiente podemos encontrar su respuesta linealizada según:

$$f\left(x\right) = \tilde{\boldsymbol{x}}^T \boldsymbol{X}_\tau^T \boldsymbol{\alpha} \tag{3.14}$$

Se construye el vector $\boldsymbol{k}$ según:

$$\boldsymbol{k} = \tilde{\boldsymbol{x}}^T \boldsymbol{X}_\tau^T \tag{3.15}$$

donde cada elemento k_i puede ser calculado mediante $k_i = \langle \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_{\tau i} \rangle$. Así, puede redefinirse la ecuación (3.14) y calcular la respuesta linealizada para cualquier valor de la variable independiente como:

$$f\left(x\right) = \boldsymbol{k}\boldsymbol{\alpha} \tag{3.16}$$

La expresión anterior puede ser desarrollada para dar como resultado:

$$f\left(x\right) = \sum_{i=1}^{m} \alpha_i \langle \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_{\tau i} \rangle = \sum_{i=1}^{m} \alpha_i \langle x, x_{\tau i} \rangle + b \tag{3.17}$$

Obsérvese que en la expresión anterior el término $\langle x, x_{\tau i} \rangle$ es el producto escalar de la variable independiente sobre la que se quiere estimar la información y un valor fijo de dicha variable independiente para el cual es conocido el valor de la señal del sensor.

En el desarrollo anterior se ha considerado una variable independiente unidimensional, por lo que el producto escalar puede sustituirse por el producto de ambas variables. Sin embargo, nuestro objetivo es expresar la ecuación (3.17) mediante productos escalares, ya que estos pueden ser sustituidos por una función kernel, que efectuará el producto escalar en un espacio de Hilbert, como se describió en la ecuación (2.64). Así, los elementos de la matriz $\boldsymbol{K}$, denominada matriz de Gram, son calculados mediante:

$$K_{ij} = \kappa\left(\tilde{\boldsymbol{x}}_{\tau i}, \tilde{\boldsymbol{x}}_{\tau j}\right) \tag{3.18}$$

y la expresión (3.17) quedará

$$f\left(x\right) = \sum_{i=1}^{m} \alpha_i \kappa\left(x, x_{\tau i}\right) + b \tag{3.19}$$

De esta forma, llegamos a una expresión similar a la planteada en la ecuación (3.7), no teniendo que suponer que la señal observada es una recta, sino que es aplicable a cualquier tipo de señal. Sin embargo, dado que el planteamiento es utilizar los coeficientes α_i como los elementos de los vectores de entrada de un clasificador, en la ecuación (3.19) se tienen tantos coeficientes como elementos tiene la señal observada.

3.2.2. Fixed kernel ridge regression

La propuesta de esta tesis es utilizar un número q de coeficientes α_i más el parámetro b que representen la señal dinámica obtenida de un sensor y constituyan la entrada del clasificador. Para conseguir este fin, nos fijamos en el método propuesto en [Suykens02] denominado Fixed Kernel Ridge Regression (FKR). Si en la regresión no lineal expuesta en la sección anterior, el método se basaba en utilizar el denominado truco del kernel para calcular los productos escalares, en este caso el método se basa en utilizar una aproximación a la transformación $\phi\left(\boldsymbol{x}\right)$ implícita cuando se utiliza un kernel.

De nuevo se parte de la matriz $\boldsymbol{X}_\tau$ de vectores extendidos de la variable independiente. Se selecciona un subconjunto de q valores de la variable independiente tal que $\boldsymbol{X}_\tau^q \subset \boldsymbol{X}_\tau$. A partir de este subconjunto se calcula la matriz de Gram $\boldsymbol{K}_{\boldsymbol{X}_\tau^q}$ asociada, cuyos elementos estarán definidos por:

$$K_{i,j} = \kappa\left(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j\right), \ \tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j \in \boldsymbol{X}_\tau^q \tag{3.20}$$

Esta matriz de Gram puede descomponerse como:

$$\boldsymbol{K}_{\boldsymbol{X}_\tau^q} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \tag{3.21}$$

donde $\boldsymbol{U}$ es la matriz de autovectores $\boldsymbol{u}_i$ y los elementos de la diagonal de la matriz $\boldsymbol{\Lambda}$ son los autovalores asociados. En [Williams01] se describe cómo, a partir de los autovalores y autovectores de una matriz de Gram dada, utilizando el método de Nyström se

puede aproximar la transformación $\phi(\tilde{\boldsymbol{x}})$ a un vector $\hat{\phi}^q(\tilde{\boldsymbol{x}})$ de q elementos calculados como:

$$\hat{\phi}^q(\tilde{\boldsymbol{x}})_i = \sqrt{\frac{q}{\lambda_i}}\boldsymbol{k}^T\boldsymbol{u}_i \tag{3.22}$$

donde $\boldsymbol{k}$ es un vector cuyo elemento j-ésimo se calcula mediante:

$$k_j = \kappa(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_j),\ \tilde{\boldsymbol{x}}_j \in \boldsymbol{X}_\tau^q \tag{3.23}$$

De esta forma, una vez seleccionado el subconjunto $\boldsymbol{X}_\tau^q$, podemos calcular su matriz de Gram asociada y a partir de esta, podemos transformar cualquier vector extendido de la variable independiente $\tilde{\boldsymbol{x}}$ en un vector transformado $\hat{\phi}^q(\boldsymbol{x})$. Así, todos los vectores extendidos de $\boldsymbol{X}_\tau$ serán transformados:

$$\boldsymbol{X}_\tau \to \phi(\boldsymbol{X}_\tau) \tag{3.24}$$

Si se representa el vector respuesta $\boldsymbol{y}$ en el nuevo espacio transformado cuyas variables independientes quedan definidas por $\phi(\boldsymbol{X}_\tau)$, la señal formará un plano lineal aproximado $\tilde{\boldsymbol{\omega}}$, que puede ser calculado mediante:

$$\tilde{\boldsymbol{\omega}} = \phi(\boldsymbol{X}_\tau)^\dagger \boldsymbol{y} \tag{3.25}$$

Finalmente, podemos calcular la señal $f(x)$ que aproxima a la señal $y(x)$ mediante la expresión:

$$f(x) = \sum_{i=1}^{q} \alpha_i \kappa(\boldsymbol{x}, \boldsymbol{X}_{\tau i}^q) + b \tag{3.26}$$

donde los coeficientes α_i se calculan como:

$$\alpha_i = \boldsymbol{\omega}^T \boldsymbol{L}\boldsymbol{V}_i \tag{3.27}$$

siendo $\boldsymbol{L}$ una matriz diagonal de dimensiones $q \times q$ cuyos elementos tienen un valor $L(i,i) = \sqrt{q/\lambda_i}$ y $\boldsymbol{V}_i$ es un vector formado por las componentes i-ésimas de los autovectores de la matriz $\boldsymbol{K}$. El vector $\boldsymbol{\omega}$ es el vector del plano encontrado en el espacio transformado. La ecuación (3.26) hace que este método se denomine también como máquinas de vectores soporte por mínimos cuadrados de tamaño fijo (fixed size LS-SVM).

Para poder ilustrar mejor los conceptos hasta ahora expuestos, se plantea un ejemplo sencillo. En la figura (3.7) se ha representado una señal artificial cuya variable independiente toma valores $x \in [1, 20]$. A partir de esos valores de la variable independiente podemos formar la matriz:

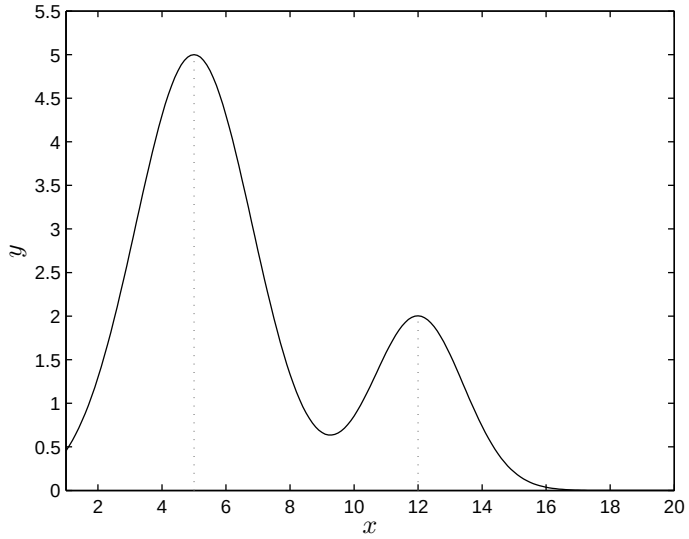


Figura 3.7: Señal artificial para explicación método FKR.

$$\boldsymbol{X}_\tau = \begin{bmatrix} 1,1 \\ 2,1 \\ 3,1 \\ \vdots \\ 20,1 \end{bmatrix} \tag{3.28}$$

Para nuestro ejemplo, seleccionaremos un valor de $q = 2$ para poder realizar representaciones. Tal como se señala en la figura, se ha elegido un subconjunto $\boldsymbol{X}_\tau^q$ tal que:

$$\boldsymbol{X}_\tau^q = \begin{bmatrix} 5,1 \\ 12,1 \end{bmatrix} \tag{3.29}$$

Seleccionamos un kernel gaussiano según la ecuación (2.67) con $\gamma = 0{,}2$. Así, la matriz de Gram de $\boldsymbol{X}_\tau^q$ será:

$$\boldsymbol{K}_{\boldsymbol{X}_\tau^q} = \begin{bmatrix} 1, & 5.54 \times 10^{-5} \\ 5.54 \times 10^{-5}, & 1 \end{bmatrix} \tag{3.30}$$

Esta matriz puede descomponerse en sus autovectores y autovalores

$$\boldsymbol{K}_{\boldsymbol{X}_\tau^q} = \begin{bmatrix} 1, & 5.54 \times 10^{-5} \\ 5.54 \times 10^{-5}, & 1 \end{bmatrix} = \begin{bmatrix} \frac{-\sqrt{2}}{2}, & \frac{-\sqrt{2}}{2} \\ \frac{-\sqrt{2}}{2}, & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} 1.001, & 0 \\ 0, & 0.9999 \end{bmatrix} \begin{bmatrix} \frac{-\sqrt{2}}{2}, & \frac{-\sqrt{2}}{2} \\ \frac{-\sqrt{2}}{2}, & \frac{\sqrt{2}}{2} \end{bmatrix} \tag{3.31}$$

De forma que ahora podemos transformar los vectores extendidos de la variable independiente en vectores de dos componentes. Por ejemplo, el vector $\tilde{\boldsymbol{x}}_{11} = [11, 1]$ queda transformado aplicando la expresión (3.22) en un vector cuyas $q = 2$ componentes se calculan:

$$\begin{aligned}
&\hat{\phi}^q(\tilde{\boldsymbol{x}}_{11})_1 = \sqrt{\tfrac{2}{1{,}001}}\left[exp(-0{,}2(36)), exp(-0{,}2(1))\right]\begin{bmatrix} \frac{-\sqrt{2}}{2} \\ \frac{-\sqrt{2}}{2} \end{bmatrix} = -0.8195 \\
&\hat{\phi}^q(\tilde{\boldsymbol{x}}_{11})_2 = \sqrt{\tfrac{2}{0{,}999}}\left[exp(-0{,}2(36)), exp(-0{,}2(1))\right]\begin{bmatrix} \frac{-\sqrt{2}}{2} \frac{\sqrt{2}}{2} \end{bmatrix} = 0.818 \\
&\hat{\boldsymbol{\phi}}^q(\tilde{\boldsymbol{x}}_{11}) = [-0.8195, 0.818]
\end{aligned} \tag{3.32}$$

Si se representa la señal de la figura (3.7), transformando cada valor de la variable independiente x en un nuevo vector transformado $\hat{\boldsymbol{\phi}}^q(\tilde{\boldsymbol{x}})$ obtendremos la gráfica representada en la figura (3.8), de la que podemos calcular su plano asociado.

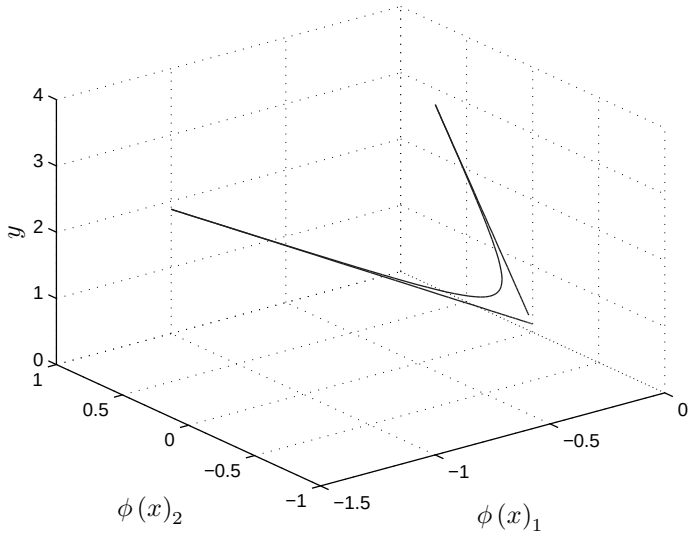


Figura 3.8: Señal representada en el nuevo espacio.

Una vez calculado el plano, podemos calcular los $q = 2$ coeficientes α_i de la expresión (3.26) y obtener el valor de la función que aproxima a la señal obtenida del sensor. Con los anteriores valores se puede obtener la aproximación de la señal original. En la figura (3.10) se representa la señal original y la aproximación calculada.

A partir del ejemplo anterior podemos apreciar cómo la señal inicial queda representada por los q elementos del vector $\boldsymbol{\alpha}$ más el parámetro b, de forma que con $d = q+1$ parámetros queda definida la señal. Sin embargo, es necesario destacar que si bien todos los puntos de la señal observada son utilizados para el cálculo del plano, los q

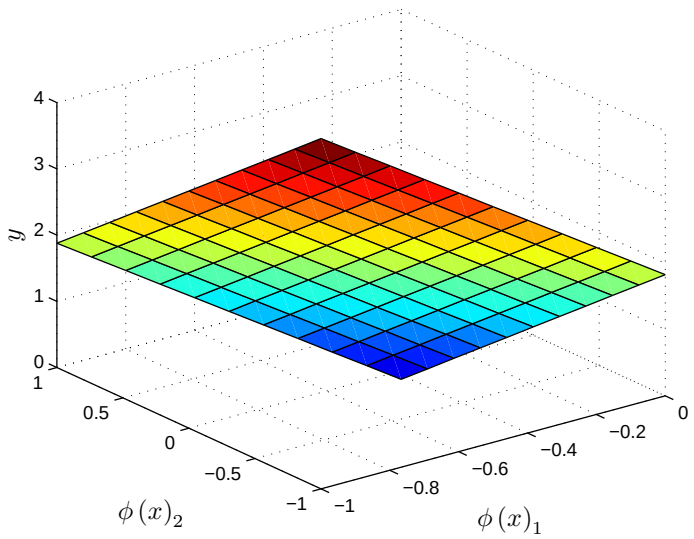


Figura 3.9: Plano de regresión de la señal en el nuevo espacio.

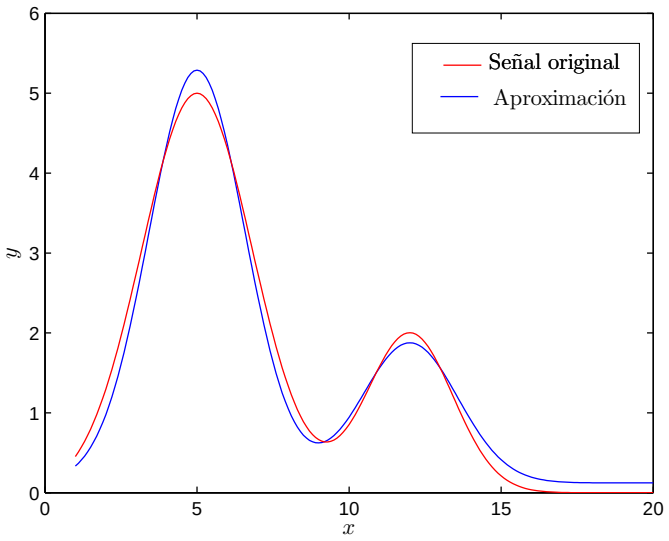


Figura 3.10: Comparación de la señal original y aproximación obtenida mediante FKR.

Tabla 3.1: Definición de kernels de tipo spline utilizados.

spline	$\kappa(\boldsymbol{x},\boldsymbol{y}) = \prod_{i=1}^{n}\left(1 + x_i y_i + \frac{1}{2}x_i y_i \min(x_i,y_i) - \frac{1}{6}\min(x_i,y_i)^3\right)$
Anova Spline 1	$\kappa(\boldsymbol{x},\boldsymbol{y}) = \prod_{i=1}^{n}\left(1 + x_i y_i + x_i y_i \min(x_i,y_i) - \frac{(x_i+y_i)}{2}\min(x_i,y_i)^2 + \frac{1}{3}\min(x_i,y_i)^3\right)$
Anova Spline 2	$\kappa(\boldsymbol{x},\boldsymbol{y}) = \prod_{i=1}^{n}\left(1 + x_i y_i + (x_i y_i)^2 + (x_i y_i)^2 \min(x_i,y_i) - x_i y_i (x_i+y_i)\min(x_i,y_i)^2 + \frac{1}{3}(x_i^2 + y_i 2 + 4x_i y_i)\min(x_i,y_i)^3 - \frac{1}{2}(x_i+y_i)\min(x_i,y_i)^4 + \frac{1}{5}\min(x_i,y_i)^5\right)$

puntos de la variable independiente sobre los que se aproxima la transformación $\hat{\phi}^q(\boldsymbol{x})$ son de gran importancia para que la función obtenida $f(x)$ sea lo más similar a la señal de entrada. Es importante destacar que dichos puntos deben ser los mismos para todas las señales consideradas. Así, en nuestro ejemplo, partimos de los valores $[5, 12]$ para realizar todo el proceso de transformación. La pregunta es cómo seleccionar esos q puntos del conjunto de entrada. En el método propuesto en [Suykens02] la selección de los q puntos de la señal se realiza mediante aquel conjunto que consiga maximizar la *entropía cuadrática de Renyi* de la matriz $\boldsymbol{K}_{\boldsymbol{X}^q}$. En [Jenssen07] se relaciona dicha entropía, para un conjunto $\boldsymbol{X}^q$ dado, como:

$$H(\boldsymbol{X}^q) = -\log \frac{1}{q^2}\sum_{i=1}^{q}\sum_{j=1}^{q}\boldsymbol{K}_{\boldsymbol{X}^q}(i,j) \tag{3.33}$$

Sin embargo, para nuestros propósitos, si se utiliza un kernel gaussiano con la forma descrita en la ecuación (2.67), se puede deducir que el conjunto que maximiza la ecuación (3.33) será aquel conjunto de q puntos equiespaciados de la señal. La ecuación (3.26) puede entenderse como un sumatorio de interpolaciones no lineales entre el valor de la variable independiente x y cada uno de los valores seleccionados de dicha variable contenidos en $\boldsymbol{X}^q$. Dada la similitud existente con la interpolación no lineal, en esta tesis se propone extender el uso de kernels, de forma que puedan emplearse los kernels basados en splines descritos en [Gu02]. En concreto, se han seleccionado los kernels descritos en la tabla(3.1). Para este tipo de kernels de base no radial, el criterio de la máxima entropía cuadrática de Renyi resulta erróneo, pues maximizar la ecuación (3.33) equivale a quedarnos con los q primeros puntos, por lo que no se conseguirá la caracterización de la señal completa. En la próxima sección se describe la solución planteada en esta tesis.

3.2.3. Algoritmo propuesto

La aportación original de esta tesis plantea aplicar el método FKR para obtener un número de parámetros que caractericen la señal y utilizarlos como características de los patrones de entrada de un clasificador. Así, en la figura (3.11) se representan las señales dinámicas procedentes de un sensor correspondientes a dos sustancias diferentes. Una vez que se aplica el método descrito en la sección anterior para un número de puntos $q = 2$, tendremos un número de puntos $d = q+1$ correspondientes a los valores del vector $\boldsymbol{\alpha}$ y el parámetro b. En la figura (3.12) se representan los parámetros obtenidos para las señales del ejemplo, pudiendo construirse un hiperplano lineal para su separación.

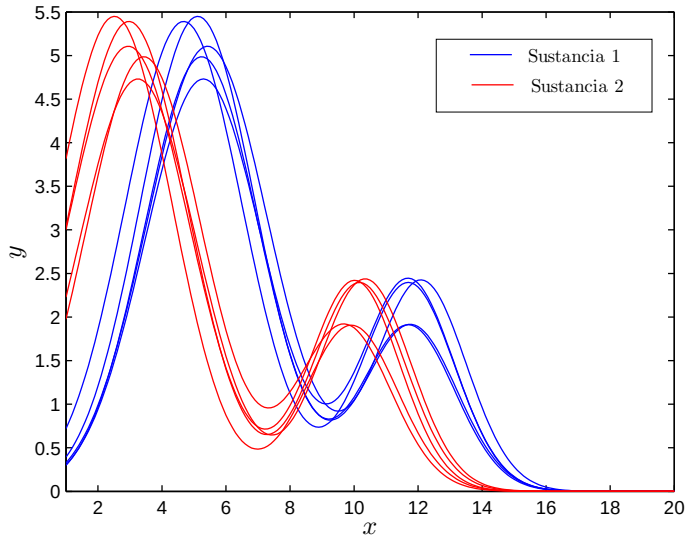


Figura 3.11: Señales artificiales correspondientes a dos sustancias.

Al igual que en el caso de la transformada wavelet es necesario contar con un conjunto de entrenamiento, en el que se determinarán cuales son los puntos óptimos para realizar el método FKR. Este conjunto de entrenamiento se selecciona de la misma forma que se describió en el algoritmo propuesto para la transformada wavelet.

El algoritmo para obtener los parámetros de cada una de las señales dinámicas proporcionadas por los sensores será el siguiente:

1. Seleccionar un tipo de kernel y prefijar el valor de sus parámetros.
2. Seleccionar un conjunto de q puntos de la variable independiente.
3. Calcular la matriz de Gram $\boldsymbol{K}_{\boldsymbol{X}^q}$ usando los q puntos seleccionados.
4. Calcular los autovectores y autovalores de la matriz $\boldsymbol{K}_{\boldsymbol{X}^q}$.

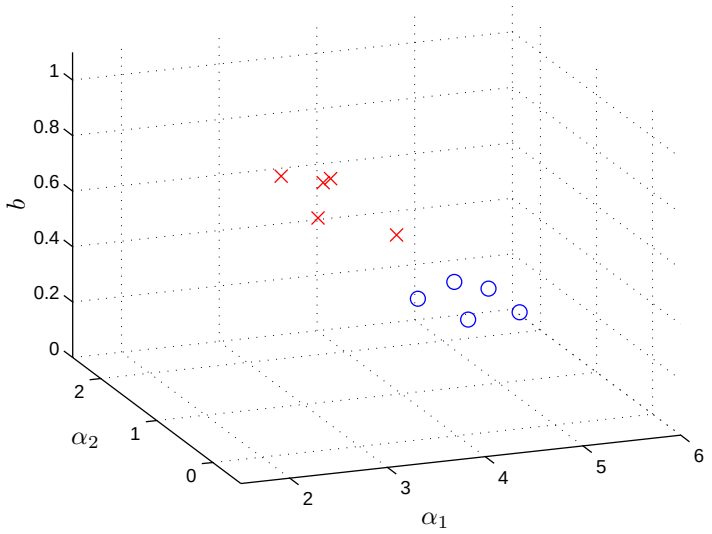


Figura 3.12: Representación de los parámetros obtenidos mediante FKR.

5. Transformar todos los puntos de la señal utilizando la ecuación (3.22).

6. A partir de los puntos $\hat{\phi}(\boldsymbol{x})$ obtenidos en el paso anterior obtener el plano lineal $\phi[\boldsymbol{\omega}, b]$ que tendrá dimensión q.

7. Obtener los coeficientes $\alpha_i = \boldsymbol{\omega}^T \boldsymbol{L}\boldsymbol{V}_i$.

El problema ahora se centra en seleccionar los q puntos de la señal. Esta selección puede hacerse de dos formas diferentes:

Puntos Equidistantes Como se ha mencionado, este criterio es el que maximiza la ecuación (3.33) para un kernel gaussiano. Si para todas las señales obtenidas siempre se toman las muestras en el mismo valor de la variable independiente, la matriz $\boldsymbol{X}^q$ será siempre la misma y por tanto se puede precalcular la matriz $\boldsymbol{K}_{\boldsymbol{X}^q}$, así como sus autovectores y autovalores y mantener una tabla con los valores de transformación de la señal. Resta solo calcular los pasos 6 y 7 del algoritmo descrito.

Puntos Óptimos En este caso se trataría de seleccionar los q puntos óptimos mediante algún proceso que minimice una función criterio. Si buscamos los q puntos para cada señal de forma individual, obtendremos una aproximación mucho más exacta a la señal dada, pero se plantea un problema muy similar al de la transformada wavelet, ya que nuestro objetivo es enviar la información a un clasificador, por lo que deberían seleccionar los mismos puntos. Por tanto en el esfuerzo de

buscar los puntos óptimos no solo debemos tener en cuenta aquellos que se ajustan bien a una señal concreta, sino cuáles se ajustan bien dentro de una clase de entrenamiento.

En el segundo caso, la búsqueda de los q puntos de una forma exhaustiva tiene un coste computacional excesivamente elevado. En su lugar, se plantea el siguiente algoritmo de búsqueda, en el que se considera que se ha establecido previamente el tipo de kernel y el valor de sus hiperparámetros.

1. Seleccionar el conjunto de puntos $\boldsymbol{X}_1^q$ equidistantes de la señal e inicializar el parámetro $J_{old} = \infty$ y $\boldsymbol{X}_{old}^q = \varnothing$.

2. Calcular la matriz de Gram $\boldsymbol{K}_{\boldsymbol{X}^q}$ usando los q puntos seleccionados.

3. Calcular los autovectores y autovalores de la matriz $\boldsymbol{K}_{\boldsymbol{X}^q}$.

4. Para cada una de las señales del conjunto de entrenamiento:

   a) Transformar todos los puntos de la señal utilizando la ecuación (3.22).

   b) A partir de los puntos $\hat{\phi}(\boldsymbol{x})$ obtenidos en el paso anterior obtener el plano lineal $\phi[\boldsymbol{\omega}, b]$ que tendrá dimensión q.

   c) Obtener los coeficientes $\alpha_i = \boldsymbol{\omega}^T \boldsymbol{L}\boldsymbol{V}_i$.

   d) Reconstruir la señal aproximada $\tilde{\boldsymbol{f}}$ mediante la ecuación (3.26).

5. A partir de las señales reconstruidas $\tilde{\boldsymbol{F}}$, calcular la función criterio

$$J(\boldsymbol{X}^q) = mse\left(\tilde{\boldsymbol{F}}, \boldsymbol{Y}\right) \tag{3.34}$$

   donde Y es el conjunto de las señales observadas de los sensores.

6. Si $J_{old} < J$ se recupera la combinación anterior $\boldsymbol{X}^q = \boldsymbol{X}_{old}^q$.

7. Si se ha llegado al número máximo de iteraciones se detiene el algoritmo y se devuelve la mejor combinación $\boldsymbol{X}^q$. Si no, se crea una nueva combinación $\boldsymbol{X}_{new}^q$ modificando cada elemento del conjunto con una variación:

$$\triangle X(i,1) = [(r - 0{,}5) * K(it) * Range(x)] \tag{3.35}$$

   donde $r \in [0,1]$ es un número aleatorio generado con distribución uniforme, $Range(x)$ es una función que transforma el valor obtenido en uno de los posibles valores de la variable independiente para los que la señal del sensor es conocida, it es el número de iteración actual y $K(it)$ es una función que sigue la ecuación:

$$K(it) = exp\left(-\zeta \frac{it}{N_{it}}\right) \tag{3.36}$$

siendo N_{it} el número máximo de iteraciones y ζ una constante, de forma que mediante la ecuación (3.36) se permiten variaciones grandes en las primeras iteraciones y pequeñas variaciones en las iteraciones finales. Además, en la generación de cada elemento de $\boldsymbol{X}^q_{new}$ se tienen en cuenta las siguientes restricciones:

$$\begin{cases} X(1,i) + \triangle X(1,i) > max(x) \;\rightarrow\; X(1,i) = max(x) \\ X(1,i) + \triangle X(1,i) < min(x) \;\rightarrow\; X(1,i) = min(x) \end{cases} \tag{3.37}$$

Además, se asegura que no se seleccione un mismo punto dos veces. Si esto sucediera, se vuelve a repetir la ecuación (3.35).

8. Con el nuevo conjunto $\boldsymbol{X}^q_{new}$ se repiten los pasos dos a siete hasta alcanzar el máximo número de iteraciones.

Es inmediato comprobar que la selección de los mejores puntos es mucho más costosa desde un punto de vista computacional que la elección de puntos equidistantes, pero en los resultados obtenidos es con este algoritmo con el que se han obtenido mejores resultados, por lo que se ha optado por este procedimiento. Al igual que en el caso de selección de coeficientes mediante wavelets o wavelet packets, para problemas multiclase se genera un modelo de preprocesado diferente por cada clase. Esto significa que se divide el conjunto entre la clase bajo estudio y el resto de señales y se ejecuta el algoritmo propuesto, repitiendo este proceso para cada una de las clases bajo estudio.

3.3. Descripción de los conjuntos de datos empleados

Para poder comprobar los métodos expuestos y hacer una comparación con los métodos planteados en el estado del arte, se han seleccionado diversos conjuntos de datos, con el objetivo de comprobar la validez de los métodos expuestos para diversos tipos de señales. A la hora de seleccionar los conjuntos de datos, se ha tenido en cuenta en primer lugar que se trate de señales dinámicas, bien con el tiempo como variable independiente, la tensión o la longitud de onda.

Clasificación de alcoholes por termomodulación. Se trata de un conjunto de datos generado en el marco de esta tesis doctoral, para lo que se desarrollaron todos los circuitos de acondicionamiento e instrumentación necesarios. Las señales proceden de un único sensor de dióxido de estaño de capa fina de la marca FIGARO, en concreto del sensor TGS2620. A dicho sensor se le aplicó una termomodulación sinusoidal de 50 mHz de frecuencia y se puso en contacto con los vapores de distintos alcoholes en una cámara cerrada. El propósito de este conjunto de datos es poder discriminar, con un único sensor, entre los diferentes tipos de alcholes

puros (etanol, propanol y metanol), aromáticos (isoamil,veratril y amil) y dos sustancias complejas (colonia y licor). El conjunto de datos incluye muestras sin diluir, y con diluciones del 75 %, 50 % y 25 %. Cada ciclo de termomodulación constituye una señal proporcionada por el sistema mediante la cuál se pretende clasificar el tipo de alcohol bajo análisis. Las señales, que fueron capturadas mediante una tarjeta PCI-DAS6031 con resolución de 8 bits, dan una respuesta normalizada entre 0 y 255 dependiendo del nivel de sustancia detectada. Este conjunto de datos fue obtenido como parte del proyecto CAM-UAH 2005/031.

Clasificación de gases por variación del flujo Este conjunto de datos fue proporcionado por el Laboratorio de Sensores del Instituto de Física Aplicada del CSIC. Se trata de un sistema de nariz electrónica basado en una modulación de flujo en la que se querían diferenciar los siguientes gases tóxicos: dióxido de nitrógeno, monóxido de carbono, tolueno y octano. Los sensores utilizados están basados en SnO_2 y TiO_2, dopados con platino mediante una técnica de deposición en radiofrecuencia. La información relativa a estos sensores puede encontrarse en [Horrillo98]. Este conjunto de datos está constituido por la información obtenida por cuatro sensores diferentes, cada uno de ellos con una señal dinámica obtenida.

Ciclovoltiamperogramas En este caso se trata de un conjunto de datos obtenido en el Departamento de Química Analítica de la Universidad de Alcalá en el que se tratan de diferenciar distintos tipos de vinos tintos y blancos mediante la aplicación de ciclovoltiamperogramas (CV) usando sensores de pasta de carbón modificados con enzimas, desarrollados en el mencionado departamento. Para cada medida, el conjunto de datos contempla cuatro CVs diferentes, procedentes de los distintos tipos de sensores. Este conjunto de datos fue obtenido como parte del proyecto GR/MAT/0916/2004.

Sistema FIA (Flow Injection Analysis) Al igual que en el caso anterior, este conjunto de datos fue obtenido en el Departamento de Química Analítica de la Universidad de Alcalá con los mismos sensores. Sin embargo, la técnica utilizada en este caso se basa en un sistema FIA, en el que se inyectaba solución tampón de forma continuada y a intervalos regulares se mezclaba un bolo de analito consistente en vino diluido y filtrado. Los vinos considerados son los mismos que para el conjunto anterior y la financiación del mismo también corresponde al proyecto GR/MAT/0916/2004.

Espectrofotometría UV-VIS Como los casos anteriores, se trata de un conjunto de datos obtenido en el Departamento de Química Analítica de la Universidad de Alcalá en la clasificación de vinos por su denominación de origen, pero a diferencia de los casos anteriores, se utiliza espectrofotometría UV-VIS con longitudes de

onda desde 200 hasta 800 nm. Este conjunto se subdivide en dos, uno que contiene los vinos blancos y otro que contiene los vinos tintos. Para la elaboración de las muestras se utilizaron diferentes marcas comerciales que contuvieran distintos tipos de uva. La toma de cada muestra procede de una nueva botella abierta en el momento de la medida para evitar los posibles efectos de oxidación del vino. Dicha muestra fue previamente filtrada para evitar la aparición de impurezas.

En la tabla (3.2) se muestra un resumen de los conjuntos de datos descritos.

Conjunto	Sensores	Técnica	Nº clases	Nº muestras	Nº sensores	Nº características
Alcoholes	SnO_2	Termo-modulación	8	849	1	71
CSIC	SnO_2 y TiO_2	Modulación por flujo	4	41	4	4x31
Ciclovolti-amperogramas	CPE modificada con enzimas	CV	4	149	4	4x320
FIA	CPE modificada con enzimas	FIA	4	99	4	4x51
Blancos	Espectrofotometría	UV-VIS	6	78	1	61
Tintos	Espectrofotometría	UV-VIS	8	155	1	61

Tabla 3.2: Resumen de los conjuntos de datos utilizados.

3.4. Resultados

En este apartado se muestran los resultados obtenidos con los diferentes conjuntos de datos procedentes de sistemas de nariz y lengua electrónica descritos en la anterior sección. En primer lugar, se procede a analizar los resultados obtenidos mediante el uso de la transformada wavelet packet con los algoritmos propuestos de selección de características y utilización de las técnicas de extensión periódica y reflexión de las señales. Además, se hace un estudio sobre las diferentes familias de filtros wavelet para cada caso.

A continuación se realiza un estudio similar aplicando la técnica FKR para los mismos conjuntos de datos. El estudio finaliza realizando una comparación con algunos de los métodos del estado del arte descritos en el capítulo anterior.

3.4.1. Selección mediante el algoritmo propuesto basado en la transformada wavelet

El algoritmo propuesto basado en la transformada wavelet se ha analizado con los diferentes conjuntos de datos descritos. Dicho algoritmo ha sido probado con la selección de coeficientes wavelet tanto por energía como por el criterio de máxima separación descrito en la ecuación (3.5). En todos los casos, se trata de medir la bondad de la técnica propuesta para describir las señales bajo estudio y la capacidad de extraer información para su clasificación posterior. Para tal fin se han probado una serie de filtros wavelet, tanto ortogonales como biortogonales para aprovechar la extensión periódica simétrica en aquellos conjuntos de datos que sea conveniente su aplicación. Los filtros empleados están descritos en la tabla (3.3). Aunque la programación de todas las rutinas ha sido desarrollada en esta tesis, se han utilizado las funciones de los filtros procedentes de MATLAB. En estas tablas además, se describe la longitud del filtro de descomposición de la aproximación o paso bajo (LD), el filtro de descomposición de detalles o paso alto (HD) y sus correspondientes filtros de recuperación (LR y HR).

Tabla 3.3: Familias de filtros wavelet empleados.

Nombre Matlab	Simétrico	Orden			
		LD	HD	LR	HR
Db4	No	8	8	8	8
Db5	No	10	10	10	10
Db6	No	12	12	12	12
Bior2.2	Si	5	3	3	5
Bior2.4	Si	9	3	3	9
Bior4.4	Si	9	7	7	9
Coif 1	No	6	6	6	6
Coif 2	No	12	12	12	12

Los resultados presentados constan del siguiente proceso:

Representación de la señal y recuperación. Se exponen las gráficas de una señal del conjunto de datos y su recuperación considerando un número diferente de coeficientes. Esta representación cumple un doble objetivo. Por un lado, es importante mostrar el tipo de señales bajo estudio, para poder realizar un análisis previo de qué filtros serán los más adecuados al conjunto de datos. El segundo objetivo consiste en estimar el número de puntos que va a ser necesario para obtener tener una buena descripción de la señal. Para la realización de esta parte se han escogido señales no singulares dentro del conjunto de datos, esto es, señales que no han sido consideradas como outliers en los experimentos realizados.

Cálculo del error cuadrático medio. Para cada conjunto se comprueba el error cuadrático medio ante diferentes familias de filtros wavelet, descritas en la tabla (3.3), y distinto número de puntos. El error cuadrático medio viene dado por la expresión:

$$mse = \frac{1}{lm} \sum_{j=1}^{l} \sum_{i=1}^{m} \left(y_i^j - f_i^j \right)^2 \tag{3.38}$$

donde N es el número de componentes de la señal, y_i^j es la componente i-ésima de la señal original y^j, mientras que la señal recuperada es f^j. Hay que tener en cuenta que para cada conjunto de datos se crean tantos modelos como clases existen. Para el cálculo del error cuadrático medio, cada modelo ejecutará solo las l señales propias de la clase para la que fue creado el mismo. Es importante destacar que para los resultados mostrados se ha multiplicado dicho error por un factor de 100, por motivos de presentación.

Acierto en clasificación. En este apartado se pretende calcular si el método propuesto consigue extraer la información necesaria para la posterior etapa de clasificación, para lo que se mide el acierto de clasificación (*accuracy*), definido como:

$$\rho = \frac{1}{n} \sum_{i=1}^{n} \delta \left(\hat{\mathcal{C}}_i, \mathcal{C}_i \right) \tag{3.39}$$

siendo n el número de muestras de test, $\mathcal{C}_i$ la clase de la muestra i, $\hat{\mathcal{C}}_i$ es la clase estimada de la muestra i y $\delta \left(\hat{\mathcal{C}}_i, \mathcal{C}_i \right)$ una función definida como:

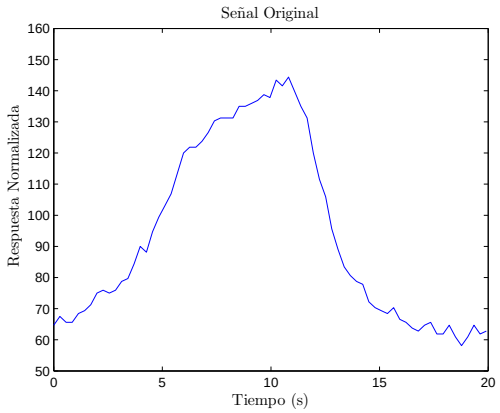
$$\delta \left(\hat{\mathcal{C}}_i, \mathcal{C}_i \right) = \begin{cases} 1, & \text{Si } \hat{\mathcal{C}}_i = \mathcal{C}_i \\ 0, & \text{Si } \hat{\mathcal{C}}_i \neq \mathcal{C}_i \end{cases} \tag{3.40}$$

Para probar este segundo objetivo se optó por un clasificador basado en una SVM lineal, dado que como se expuso en el capítulo anterior es un clasificador extremadamente sencillo que prácticamente no necesita parámetros para ser ajustado. Aunque los resultados de clasificación no son satisfactorios en muchos casos, no es el objetivo de esta parte examinar la exactitud de la clasificación, sino la comparación entre las diferentes familias de wavelets, la extensión simétrica y el número de puntos utilizado. La mejora de la exactitud en la clasificación será tratada en los próximos capítulos centrados en los métodos de clasificación.

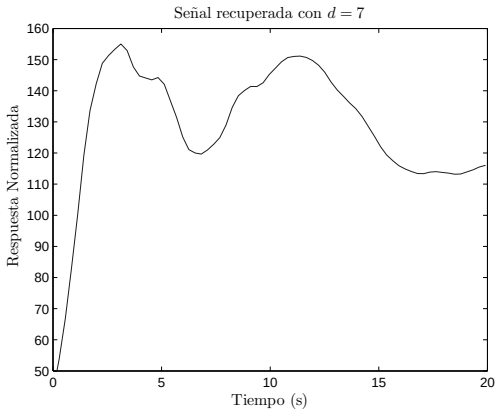
Método de selección de características. Por cada conjunto se han realizado dos tablas que contienen la información descrita en los puntos anteriores. La primera de las tablas contiene la información cuando el método de selección de los coeficientes wavelet ha sido el criterio de quedarnos con aquellos que, una vez analizado el conjunto, acumulan mayor energía. Para la realización de la segunda

tabla se ha optado por el método de máxima separabilidad de las clases en la selección de los coeficientes wavelet.

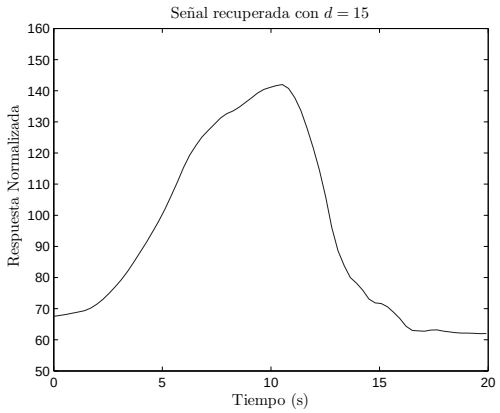
El primer conjunto considerado es el de discriminación de alcoholes. En la figura (3.13(a)) se ha representado la respuesta del sensor normalizada entre 0 y 255 ante un ciclo de termomodulación en presencia de etanol. Para el ejemplo descrito en esta figura se seleccionó un filtro wavelet ortogonal de Daubechies de longitud 8, por lo que no se aplica la extensión simétrica. La razón de aplicación de este filtro es que la señal no presenta cambios entre el final de la misma y el principio, dado que estamos considerando el ciclo completo de termomodulación, por lo que se puede realizar la extensión periódica no simétrica. El nivel de profundidad aplicado fue de cuatro. En las figuras



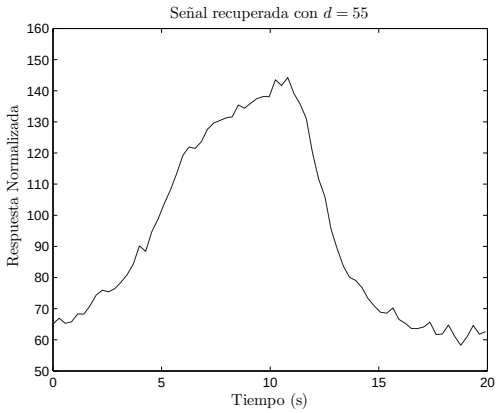
(a) *Señal termomodulada bajo presencia de Etanol*



(b) *Señal recuperada con 7 puntos*



(c) *Señal recuperada con 15 puntos.*



(d) *Señal recuperada con 55 puntos.*

Figura 3.13: Recuperación de señal termomodulada en presencia de etanol utilizando diferente número de coeficientes wavelet.

(3.13(b)-3.13(d)) se ha representado la recuperación de esta señal seleccionando un

número diferente de coeficientes por el método de máxima energía. Se puede apreciar, cómo la información contenida en la señal de cara a su recuperación empieza a ser muy completa a partir de un número de componentes $d = 15$. Es importante destacar cómo se produce un efecto de gran distorsión si el número de coeficientes seleccionado no alcanza el número de coeficientes de la aproximación de la última etapa, tal como sucede cuando $d = 7$. En las tablas (3.4 y 3.5) se han representado los resultados obtenidos para el método de selección de características por máxima energía y por separabilidad de clases respectivamente. Para este conjunto de datos se puede ver cómo no hay gran diferencia entre los diferentes tipos de familias utilizados, pues la extensión periódica no simétrica no introduce componentes de alta frecuencia. Por otro lado, la diferencia entre los criterios de selección de coeficientes wavelet es más significativa cuando se escogen pocos coeficientes como es el caso de $d = 7$.

Filtro Wavelet	d = 7		d = 15		d = 25		d = 55	
	Mse	Acc	Mse	Acc	Mse	Acc	Mse	Acc
Db4	8.10	0.59	0.03	0.71	0.00	0.74	0.00	0.80
Db5	8.02	0.65	1.40	0.73	0.01	0.78	0.00	0.79
Db6	9.07	0.65	3.45	0.67	0.03	0.75	0.00	0.79
Bior2.2	2.34	0.65	0.01	0.72	0.00	0.77	0.00	0.80
Bior2.4	2.34	0.67	0.01	0.70	0.00	0.77	0.00	0.80
Bior4.4	4.14	0.65	0.01	0.78	0.00	0.79	0.00	0.80
Coif 1	5.67	0.63	0.02	0.71	0.00	0.78	0.00	0.81
Coif 2	7.92	0.55	3.18	0.61	0.03	0.74	0.00	0.80

Tabla 3.4: Comparación de familias de filtros y wavelets para el conjunto de datos alcoholes con selección por criterio de máxima energía.

En la figura (3.14) se representa la señal por espectrofotometría obtenida para un vino tinto con denominación de origen de La Mancha. En este caso, se ha optado por el filtrado wavelet biortogonal, pues como se puede apreciar en dicha imagen la absorbancia obtenida para longitudes de onda bajas difiere mucho de la obtenida para longitudes de onda altas, por lo que la extensión correcta es la extensión simétrica. Se puede ver, al igual que en el caso anterior, cómo el error obtenido entre la señal original y la señal recuperada disminuye a medida que se aumenta el número de coeficientes seleccionados. Sin embargo, en este caso podemos apreciar cómo la información fundamental de la señal se obtiene antes que en el caso anterior. Este hecho es debido a que la información contenida en señales procedentes de espectrofotometría varía muy lentamente. De hecho, en la selección de coeficientes podemos apreciar que no existe diferencia entre la aplicación de la transformada wavelet y la transformada wavelet packet, pues siempre trata de escoger los coeficientes de la aproximación.

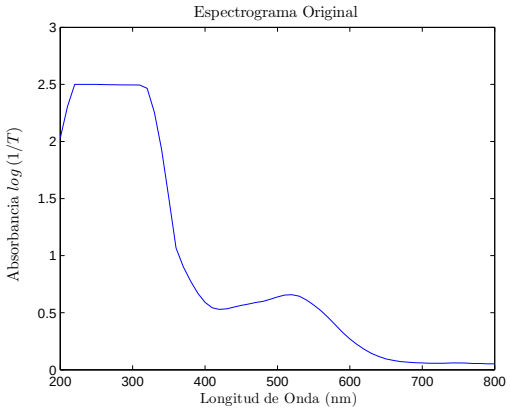
Filtro Wavelet	d = 7		d = 15		d = 25		d = 55	
	Mse	Acc	Mse	Acc	Mse	Acc	Mse	Acc
Db4	11.77	0.63	3.89	0.80	2.02	0.81	0.79	0.81
Db5	12.75	0.74	4.09	0.81	2.05	0.81	1.44	0.81
Db6	12.68	0.62	6.94	0.76	2.30	0.81	1.40	0.81
Bior2.2	6.38	0.75	2.17	0.80	1.56	0.80	0.90	0.79
Bior2.4	6.37	0.74	2.17	0.80	1.56	0.80	0.90	0.80
Bior4.4	6.47	0.75	2.10	0.80	1.46	0.80	0.50	0.80
Coif 1	10.97	0.73	3.61	0.81	1.89	0.80	1.16	0.81
Coif 2	11.86	0.60	7.46	0.78	2.39	0.81	0.78	0.81

Tabla 3.5: Comparación de familias de filtros y wavelets para el conjunto de datos alcoholes con selección por criterio de separabilidad entre clases.

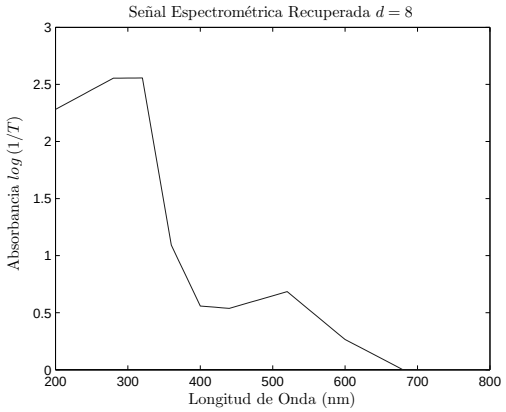
En las tablas (3.6 y 3.7) se exponen los resultados para el conjunto de datos obtenido procedente de las diferentes denominaciones de origen de vino tinto utilizando espectrofotometría UV-VIS, cuyos resultados son similares a los obtenidos con vinos blancos. En ambos conjuntos las conclusiones que se pueden obtener son muy similares, verificando el mejor comportamiento de los filtros biortogonales frente a los filtros ortogonales por la diferencia entre la extensión simétrica y la extensión periódica no simétrica. Se comprueba cómo la recuperación de la señal es, para estos conjuntos de datos, mejor cuando el criterio de selección es la energía de los coeficientes, mientras que la exactitud de la clasificación es ligeramente mejor en el caso de selección por el criterio de separación entre clases.

Filtro Wavelet	d = 8		d = 14		d = 16		d = 20	
	Mse	Acc	Mse	Acc	Mse	Acc	Mse	Acc
Db4	0.92	0.75	0.28	0.77	0.21	0.77	0.10	0.77
Db5	2.19	0.79	0.35	0.79	0.27	0.82	0.20	0.83
Db6	5.85	0.71	0.60	0.77	0.45	0.79	0.27	0.81
Bior2.2	0.17	0.70	0.08	0.73	0.05	0.79	0.01	0.82
Bior2.4	0.17	0.71	0.08	0.77	0.05	0.81	0.01	0.85
Bior4.4	0.18	0.79	0.06	0.82	0.04	0.85	0.01	0.88
Coif 1	0.37	0.73	0.19	0.77	0.15	0.80	0.09	0.83
Coif 2	1.66	0.75	0.49	0.79	0.40	0.81	0.24	0.81

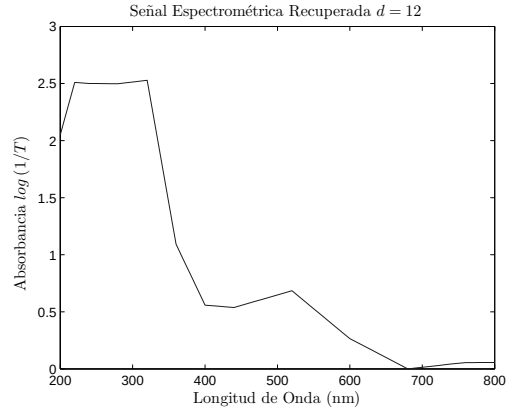
Tabla 3.6: Comparación de familias de filtros y wavelets para el conjunto de datos de vinos tintos por espectrometría UV-VIS con selección por criterio de máxima energía.



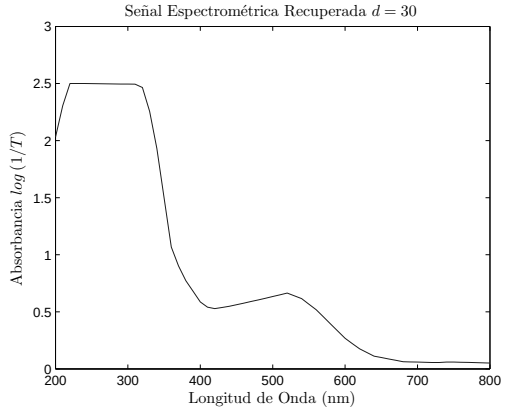
(a) *Señal espectrométrica Original para un vino tinto con denominación de origen de La Mancha tos.*



(b) *Señal espectrométrica recuperada con 8 puntos.*



(c) *Señal espectrométrica recuperada con 12 puntos.*



(d) *Señal espectrofotométrica recuperada con 30 puntos.*

Figura 3.14: Recuperación de espectrofotograma UV-VIS utilizando diferente número de coeficientes wavelet.

En la figura (3.15) se ha representado la recuperación de la señal obtenida del conjunto de datos FIA sometido a un bolo de vino tinto de denominación de origen Madrid. En este caso, la señal registrada solo considera el proceso oxidación de la reacción, por lo que el uso de filtros biortogonales con extensión simétrica dará mejores resultados que la extensión periódica empleada con filtros ortogonales. Se puede observar cómo cuando se seleccionan pocos coeficientes la señal recuperada dista mucho de la señal original. Se puede observar cómo la información proporcionada por este sensor tiene parte de alta frecuencia que será seleccionada al utilizar los coeficientes procedentes de los detalles de la primera descomposición.

Las tablas (3.8 y 3.9) muestran los resultados para la selección de características por energía y por máxima separación de clases. En ambos casos los resultados son bastante

Filtro Wavelet	d = 8		d = 12		d = 16		d = 20	
	Mse	Acc	Mse	Acc	Mse	Acc	Mse	Acc
Db4	35.49	0.69	28.74	0.77	28.00	0.77	25.42	0.83
Db5	41.84	0.64	41.48	0.74	36.49	0.78	28.90	0.84
Db6	39.74	0.65	39.93	0.77	39.96	0.79	33.45	0.79
Bior2.2	27.39	0.67	20.14	0.77	20.13	0.80	20.10	0.84
Bior2.4	27.46	0.75	27.08	0.77	27.23	0.83	19.94	0.87
Bior4.4	31.19	0.71	30.11	0.80	30.06	0.82	20.55	0.89
Coif 1	39.63	0.68	35.64	0.75	25.11	0.79	25.40	0.83
Coif 2	34.28	0.71	27.59	0.73	27.77	0.78	27.69	0.83

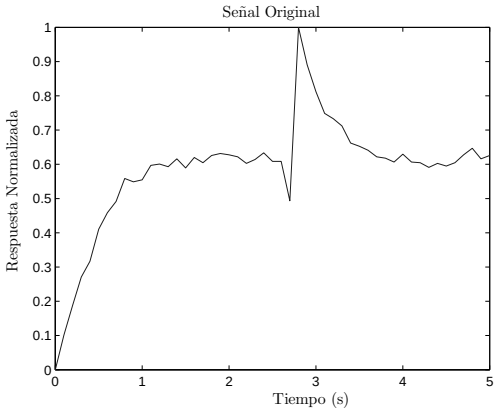
Tabla 3.7: Comparación de familias de filtros y wavelets para el conjunto de datos de vinos tintos por espectrofotometría UV-VIS con selección por criterio de separabilidad entre clases.

similares, por lo que puede deducirse que la selección en ambos casos es similar. Se comprueba que los familias de filtros biortogonales dan mejor resultado aunque las diferencias en este caso no son sensibles.

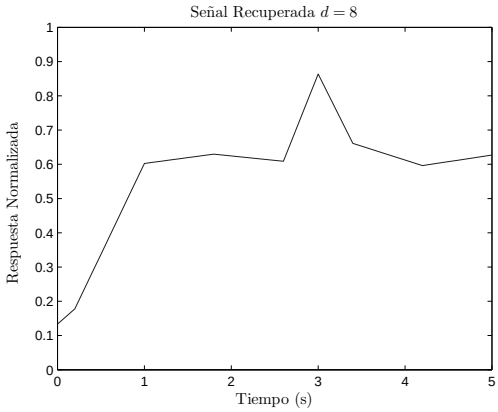
Filtro Wavelet	d = 8		d = 14		d = 20	
	Mse	Acc	Mse	Acc	Mse	Acc
Db4	1.31	0.91	0.57	0.91	0.26	0.92
Db5	2.33	0.88	0.93	0.91	0.46	0.91
Db6	7.93	0.86	0.93	0.89	0.43	0.91
Bior2.2	0.51	0.92	0.26	0.94	0.18	0.92
Bior2.4	0.50	0.88	0.26	0.94	0.18	0.94
Bior4.4	0.54	0.89	0.25	0.92	0.17	0.94
Coif 1	1.02	0.88	0.31	0.91	0.23	0.92
Coif 2	9.39	0.85	0.73	0.91	0.52	0.89

Tabla 3.8: Comparación de familias de filtros y wavelets para el conjunto de datos FIA con selección por criterio de máxima energía.

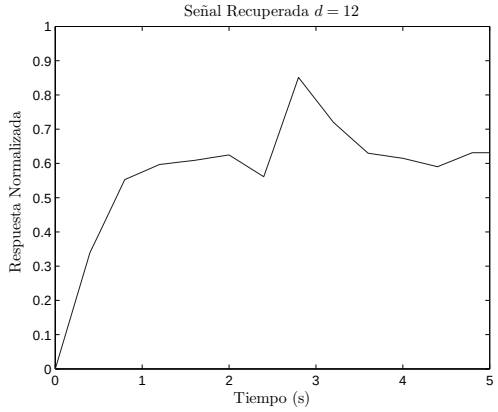
Por último, en la figura (3.16) se ha procedido de forma similar a los conjuntos de datos anteriores, representando el ciclovoltiamperograma del sensor modificado con enzima de tirosinasa. En la representación de la señal original vemos cómo la curva descrita por el ciclovoltiamperograma no tiene por qué cerrarse para el rango de tensiones considerado, aunque las diferencias no son grandes. Esto hace que podamos utilizar tanto la extensión periódica como la extensión simétrica sin demasiadas diferencias.



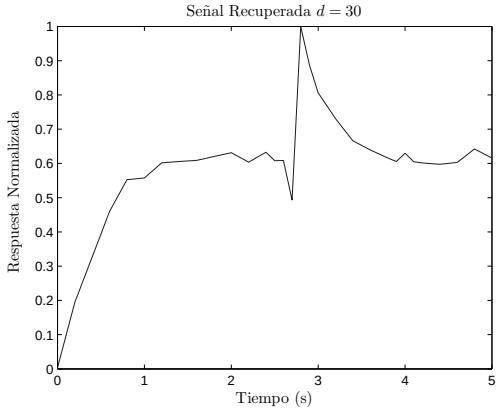
(a) *Señal original para un vino tinto con denominación de origen de Madrid.*



(b) *Señal recuperada con 8 puntos*



(c) *Señal recuperada con 12 puntos*



(d) *Señal recuperada con 30 puntos.*

Figura 3.15: Recuperación de señal en sistema FIA con diferentes coeficientes wavelet.

Se puede apreciar cómo para un número excesivamente bajo de coeficientes parte de la curva se pierde, por lo que la información completa de la histéresis no se consigue recuperar. Estas curvas se caracterizan por una variación suave de las mismas, por lo que no se han apreciado grandes diferencias entre utilizar un esquema wavelet clásico y utilizar un esquema wavelet packet.

En las tablas (3.10 y 3.11) se ha hecho una comparación similar a los casos anteriores, utilizando diversos tipos de filtrado wavelet y número de puntos seleccionados. En este caso podemos apreciar cómo los resultados son muy similares en el caso de considerar familias ortogonales y biortogonales, con sus extensiones simétrica y periódica. Se aprecia que a partir de un número de puntos los resultados de clasificación no varían sensiblemente, por lo que la información discriminante se está recogiendo en los coeficientes seleccionados. También se aprecia cómo cuando tenemos un caso de selec-

Filtro Wavelet	d = 8		d = 14		d = 20	
	Mse	Acc	Mse	Acc	Mse	Acc
Db4	12.32	0.89	8.06	0.89	6.10	0.91
Db5	13.88	0.88	9.41	0.89	7.89	0.92
Db6	15.54	0.88	10.71	0.89	7.84	0.89
Bior2.2	8.07	0.88	4.81	0.88	1.82	0.89
Bior2.4	8.11	0.88	3.95	0.88	2.02	0.91
Bior4.4	8.66	0.88	7.49	0.89	4.25	0.89
Coif 1	9.61	0.88	8.07	0.89	5.87	0.91
Coif 2	21.16	0.88	7.73	0.89	7.39	0.89

Tabla 3.9: Comparación de familias de filtros y wavelets para el conjunto de datos FIA con selección por criterio de separabilidad de clases.

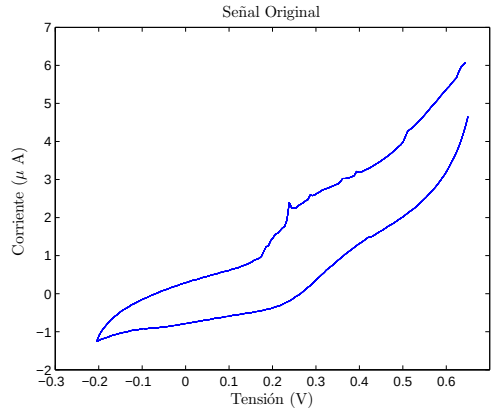
ción de pocos coeficientes, el criterio de selección por separación de clases proporciona mejores resultados que el criterio de selección por energía, pero estas diferencias no son significativas cuando el número de coeficientes seleccionado es mayor.

Filtro Wavelet	d = 10		d = 40		d = 80	
	Mse	Acc	Mse	Acc	Mse	Acc
Db4	4448.27	0.63	21.32	0.75	3.74	0.78
Db5	5315.61	0.60	25.76	0.78	2.95	0.79
Db6	5240.71	0.62	33.89	0.82	7.16	0.79
Bior2.2	1446.11	0.72	15.46	0.79	6.84	0.80
Bior2.4	1443.85	0.73	14.79	0.78	5.48	0.80
Bior4.4	1710.89	0.76	9.22	0.80	3.91	0.79
Coif 1	3360.27	0.67	24.32	0.78	8.71	0.78
Coif 2	5464.68	0.56	36.75	0.78	5.01	0.79

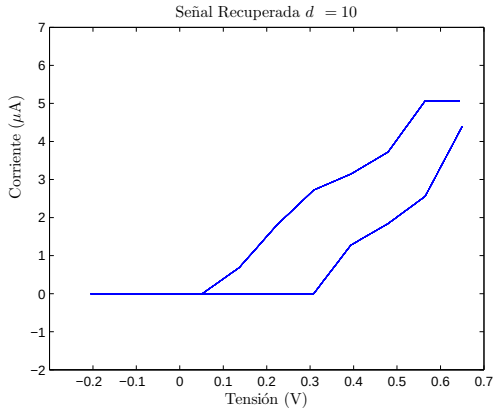
Tabla 3.10: Comparación de familias de filtros y wavelets para el conjunto de datos de ciclovoltiamperogrmas con selección por criterio de máxima energía.

Del análisis de estos conjuntos de datos podemos extraer una serie de conclusiones que deben ser tenidas en cuenta cuando se use la transformada wavelet como método de extracción de información en señales dinámicas:

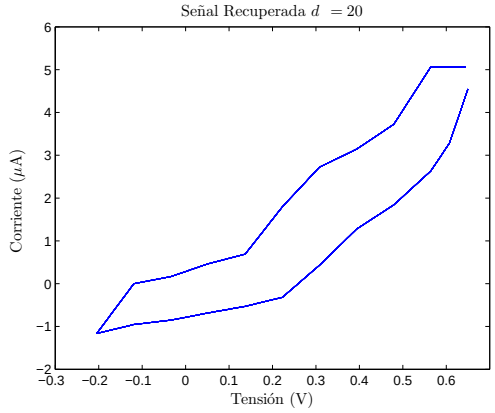
- Respecto al error en la recuperación de la señal, siempre es menor el cometido cuando se utiliza el criterio de selección por energía que por selección de separación de clases. Sin embargo, la parte más importante de esta etapa es proporcionar información que pueda ser correctamente clasificada en etapas posteriores.



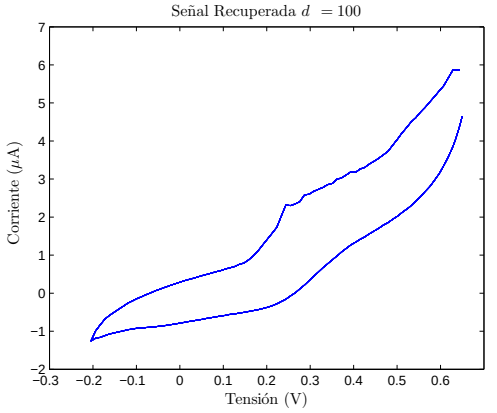
(a) *Señal respuesta de un sensor ante vino tinto.*



(b) *Señal recuperada con 10 puntos.*



(c) *Señal recuperada con 20 puntos.*



(d) *Señal recuperada con 100 puntos.*

Figura 3.16: Recuperación de señal de ciclovoltiamperograma con diferentes coeficientes wavelet

- Si el conjunto de datos es reducido y se utilizan pocas muestras para el proceso de entrenamiento, el criterio por separación de clases puede dar lugar a seleccionar coeficientes que no son discriminantes, obteniendo mejores resultados mediante la selección de coeficientes por energía. Sin embargo, si el conjunto es suficientemente grande y las señales de diferentes clases tienen un cierto parecido en su forma, la selección de coeficientes por separación de clases proporciona mejores resultados.

- La aplicación de un filtrado wavelet ortogonal o biortogonal y por tanto con extensión periódica o extensión periódica simétrica respectivamente depende del tipo de aplicación que estemos considerando y la forma en que ha sido guardada la información. Se deberá observar por tanto si existe diferencia entre el comienzo y el final del grupo de señales bajo estudio.

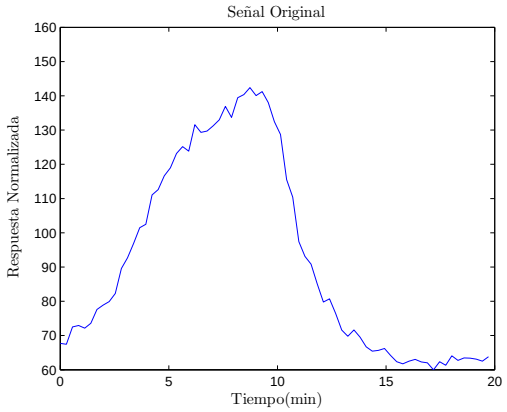
Filtro Wavelet	d = 10		d = 40		d = 80	
	Mse	Acc	Mse	Acc	Mse	Acc
Db4	5791.69	0.68	4419.64	0.66	3459.29	0.69
Db5	5927.71	0.70	4868.39	0.67	4158.76	0.66
Db6	6209.14	0.71	4982.39	0.66	4358.63	0.69
Bior2.2	6779.38	0.71	4488.74	0.72	3546.84	0.76
Bior2.4	6780.08	0.76	4564.21	0.72	3639.17	0.78
Bior4.4	6457.62	0.72	4940.29	0.72	3905.75	0.78
Coif 1	6766.77	0.67	5006.12	0.70	3862.94	0.79
Coif 2	6776.13	0.66	5394.16	0.69	4300.66	0.66

Tabla 3.11: Comparación de familias de filtros y wavelets para el conjunto de datos de ciclovoltiamperogramas con selección por criterio de separación de clases.

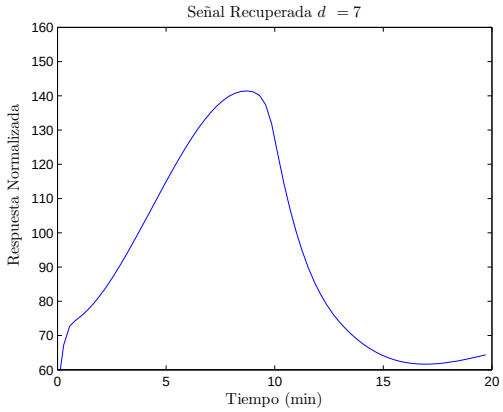
3.4.2. Selección mediante el algoritmo propuesto basado en FKR

Una vez realizada la comparación con el algoritmo propuesto basado en la transformada wavelet y sus variantes, en esta sección se comprobarán los resultados obtenidos con el método propuesto basado en FKR. Para poder realizar una comparación con el método anterior, se ha reproducido la recuperación de las mismas señales consideradas con dicho método. El procedimiento de comparación de los resultados es idéntico al expuesto para el método basado en la transformada wavelet, pero en este caso se utilizará para realizar los cálculos el kernel RBF descrito en la ecuación (2.67) y los kernel basados en splines descritos en la tabla (3.1). Solamente en el caso del kernel RBF es necesario ajustar el hiperparámetro γ, lo que se realizó mediante varias pruebas y se muestran los resultados obtenidos con el mejor valor encontrado. Al realizar las tablas se utilizaron los conjuntos de datos ya divididos en entrenamiento y test que fueron usados para mostrar los resultados de la sección anterior. En la figura (3.17) se ha procedido a recuperar la señal original procedente del conjunto de datos de alcoholes. En este caso se ha utilizado un kernel de tipo spline con distintos números de puntos considerados para proyectar el espacio de entrada en un espacio transformado. Se puede observar cómo la recuperación obtenida en la figura (3.17(b)) guarda una mejor similitud con la señal original que la obtenida en la figura (3.13(b)). Sin embargo, para un número de puntos elevado se observa en la figura (3.17(d)) cómo la señal queda más suavizada que en el caso de la figura (3.13(d)).

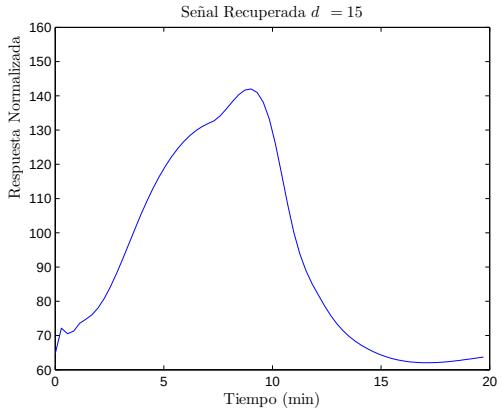
En la tabla (3.12) se muestra el error cuadrático medio y el parámetro de exactitud (*accuracy*) para los diferentes kernels utilizados y variando el número de puntos con el conjunto de datos de alcoholes. La primera observación que se debe hacer es que el parámetro de exactitud decrece cuando el número de puntos es elevado. El método sufre



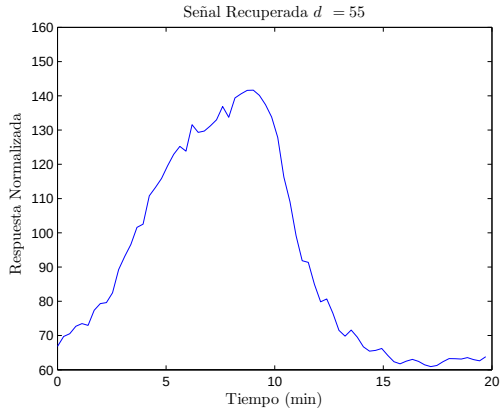
(a) *Señal termomodulada bajo presencia de etanol*



(b) *Señal recuperada con 7 puntos.*



(c) *Señal recuperada con 15 puntos.*



(d) *Señal recuperada con 55 puntos.*

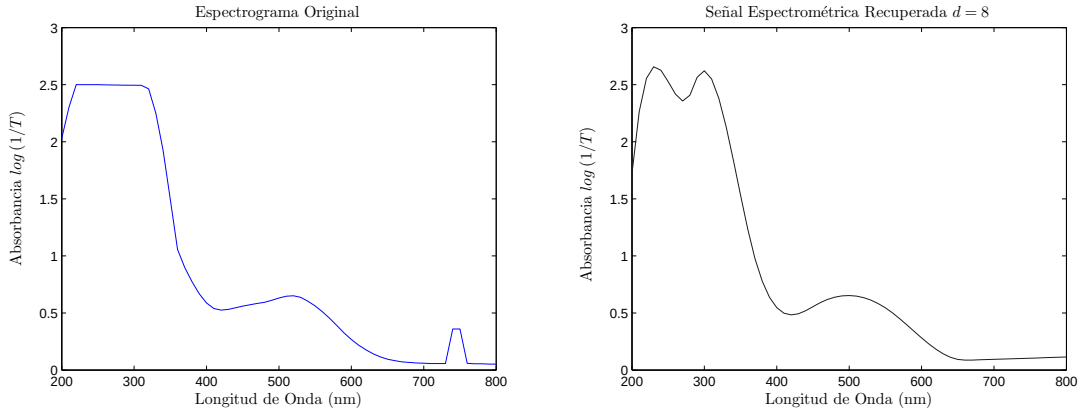
Figura 3.17: Recuperación de señal de etanol por medio de FKR.

por tanto de la denominada "maldición de la dimensión"([Bishop06]) especialmente utilizando un clasificador lineal. Sin embargo, cuando el número de puntos considerado es $d = 7$ se alcanzan resultados sensiblemente superiores a los obtenidos con el método de transformada wavelet con ese mismo número de puntos.

En la figura (3.18) se muestra la recuperación de un espectrofotograma cuyo original se ha obtenido ante la presencia de un vino tinto de La Mancha. En este caso, se ha utilizado un kernel gaussiano y en la figura (3.18(b)) podemos ver un efecto característico, que se ha observado en el desarrollo de este trabajo, cuando el número de puntos considerado no es suficiente. En este caso aparecen en las zonas bajas de longitud de onda una deformación gaussiana clara, mientras que la señal original se mantiene prácticamente constante en esa zona. No obstante, la señal recuperada presenta un mejor comportamiento para tan pocos puntos que en el caso de la transformada wavelet, lo que queda corroborado al comparar la tabla (3.13) con las tablas (3.6) y (3.7).

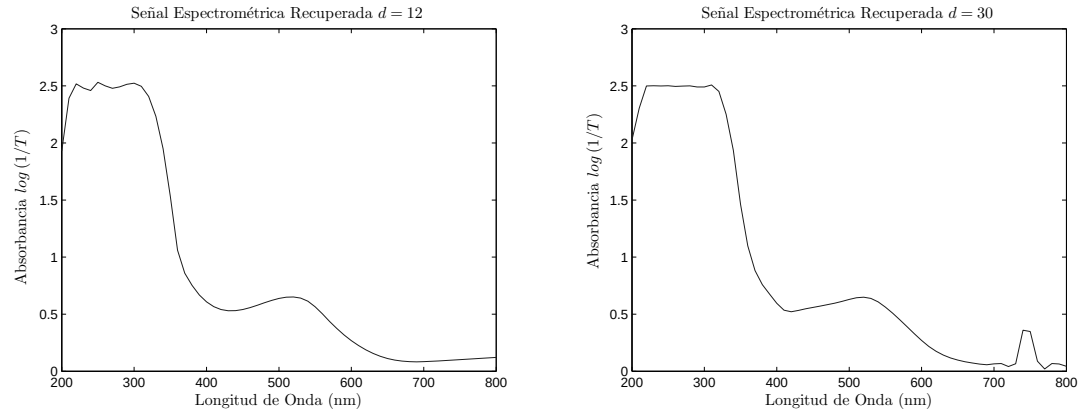
Kernel	d = 7		d = 15		d = 25		d = 55	
	Mse	Acc	Mse	Acc	Mse	Acc	Mse	Acc
RBF	0.43	0.71	0.22	0.71	0.17	0.71	0.01	0.54
Spline	0.38	0.83	0.16	0.77	0.12	0.71	0.01	0.65
Anova Spline1	0.82	0.73	0.46	0.73	0.39	0.76	0.01	0.66
Anova Spline2	0.66	0.68	0.42	0.77	0.21	0.81	0.01	0.12

Tabla 3.12: Comparación de kernels y número de puntos para conjunto de datos de alcoholes.



(a) *Señal espectrofotométrica original para un vino tinto con denominación de origen de La Mancha.*

(b) *Señal espectrofotométrica recuperada con 8 puntos.*



(c) *Señal espectrofotométrica recuperado con 12 puntos.*

(d) *Señal espectrofotométrica recuperado con 30 puntos.*

Figura 3.18: Recuperación de señal procedente de espectrofotometría UV-VIS mediante FKR.

Las tabla (3.13) muestra los resultados con diferentes kernels y número de puntos para los conjuntos de espectrofotometría UV-VIS de vinos tintos. Para estos conjuntos de datos los kernels ANOVA spline tienen un mejor comportamiento, debido a que las señales procedentes de la espectrofotometría UV-VIS son de muy baja frecuencia y son muy bien aproximadas por funciones de suavizado como los kernels propuestos. Es importante destacar que se produce cierta caída de la exactitud de clasificación cuando se aumenta el número de puntos, aunque en este caso no llega a ser tan exagerado como en el conjunto de alcoholes.

Kernel	d = 8		d = 14		d = 16		d = 20	
	Mse	Acc	Mse	Acc	Mse	Acc	Mse	Acc
RBF	0.59	0.73	0.24	0.70	0.20	0.72	0.18	0.73
Spline	1.28	0.70	0.20	0.85	0.16	0.86	0.09	0.85
Anova Spline1	1.47	0.76	0.96	0.89	0.60	0.92	0.51	0.90
Anova Spline2	1.71	0.67	0.57	0.72	0.34	0.75	0.17	0.76

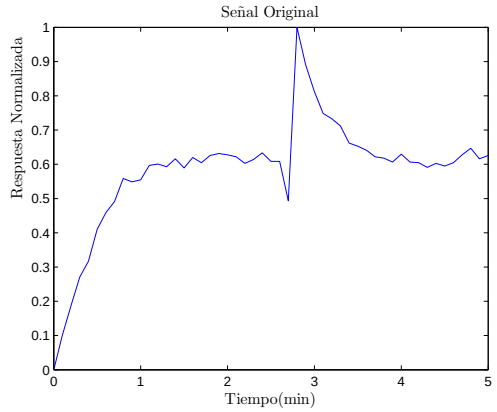
Tabla 3.13: Comparación de kernels y número de puntos para conjunto de datos de espectrofotograma de vinos tintos

En la figura (3.19) se ha realizado la recuperación de la misma señal original que en el caso de la figura (3.15), mientras que en la tabla (3.14) se muestran los resultados con diferentes kernels y variando el número de puntos. Nuevamente se comprueba el mejor comportamiento frente a la transformada wavelet, tanto en error cuadrático medio como en exactitud, de este método cuando el número de puntos es bajo. Sin embargo, vuelve a aumentar el error de clasificación cuando seleccionamos un número de puntos elevado.

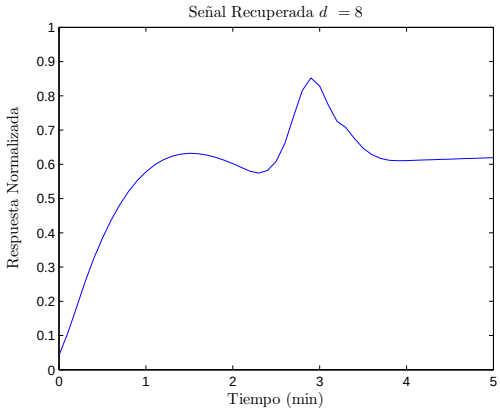
Kernel	d = 8		d = 14		d = 20	
	Mse	Acc	Mse	Acc	Mse	Acc
RBF	0.56	0.91	0.35	0.85	0.26	0.73
Spline	0.50	0.80	0.32	0.74	0.25	0.73
Anova Spline1	0.53	0.88	0.34	0.74	0.28	0.71
Anova Spline2	0.48	0.83	0.32	0.77	0.26	0.76

Tabla 3.14: Comparación entre diferentes kernels y diferentes puntos para el conjunto de datos FIA

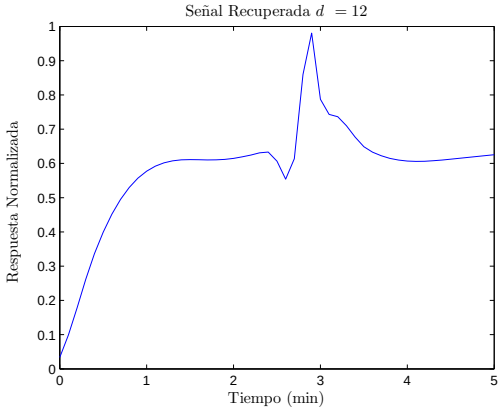
Por último, se muestran en la figura (3.20) y en la tabla (3.15) los resultados obtenidos con el conjunto de datos de ciclovoltiamperogramas. En este caso el kernel RBF



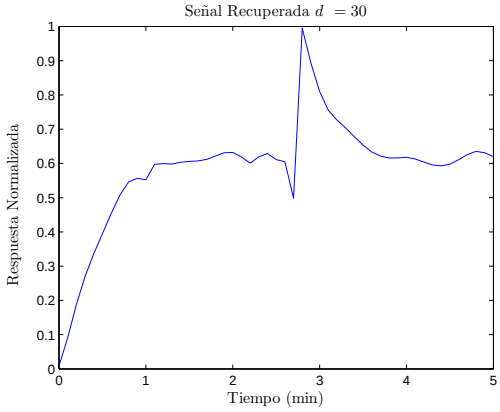
(a) *Señal original para un vino tinto con denominación de origen de Madrid.*



(b) *Señal recuperada con 8 puntos.*



(c) *Señal recuperada con 12 puntos.*



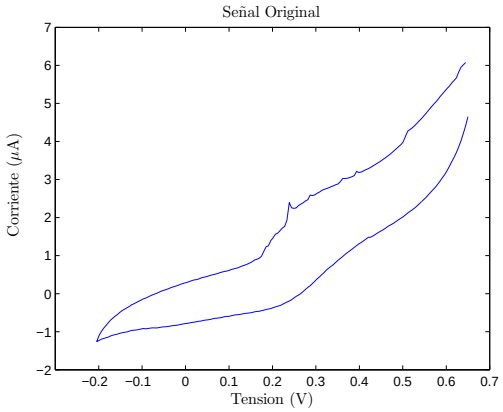
(d) *Señal recuperada con 30 puntos.*

Figura 3.19: Recuperación señal FIA mediante FKR.

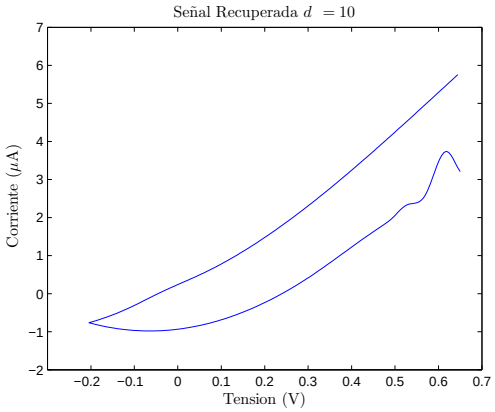
obtiene unos resultados muy superiores a los kernel de tipo spline, pero todos ellos fallan en la clasificación cuando el número de puntos considerado es elevado.

Kernel	d = 10		d = 40		d = 100	
	Mse	Acc	Mse	Acc	Mse	Acc
RBF	3068.67	0.84	43.15	0.74	16.70	0.41
Spline	107.20	0.65	39.42	0.33	16.43	0.32
Anova Spline1	157.48	0.74	47.64	0.52	27.24	0.39
Anova Spline2	116.14	0.65	75.43	0.34	22.54	0.30

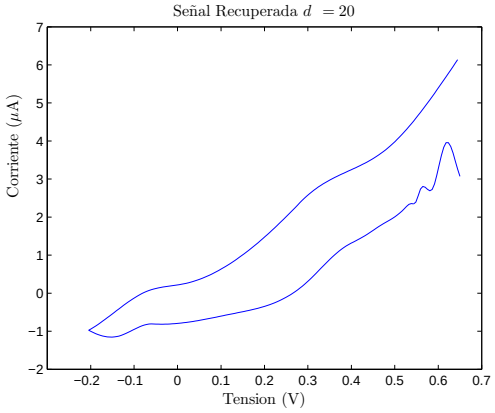
Tabla 3.15: Comparación entre diferentes kernels y diferentes puntos para el conjunto de datos Ciclovoltiamperograma



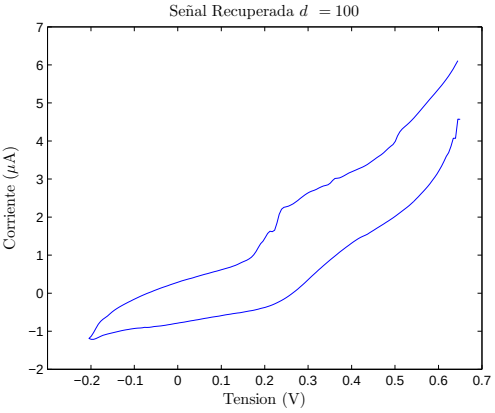
(a) *Señal respuesta de un sensor ante vino tinto*



(b) *Señal recuperada con 10 puntos*



(c) *Señal recuperada con 20 puntos.*



(d) *Señal recuperada con 100 puntos.*

Figura 3.20: Recuperación de ciclovoltiamperograma utilizando FKR.

Los resultados mostrados permiten obtener interesantes conclusiones sobre el método FKR:

- El método sufre de la "maldición de la dimensión" debiendo ser muy cuidadosos con el número de puntos escogido sobre el que se aproxima la transformación. Sin embargo, es necesario constatar que si nuestro interés es representar la señal con mejor exactitud, en términos de error cuadrático medio, cuanto mayor sea el número de puntos mejor comportamiento presenta el método.

- No existe un kernel que se haya demostrado superior al resto entre los considerados al mostrar los resultados. Sí es necesario decir que se hicieron pruebas con otro tipo de kernels como el polinomial y el B-spline con resultados mucho menos satisfactorios.

- Cuando el número de puntos considerado es muy reducido el método se comporta mejor que la transformada wavelet. Aunque esta afirmación será ampliada en la siguiente sección, se destaca aquí la importancia que tiene el hecho de obtener mejores ratios con números de puntos bajos, pues ahorrará muchas operaciones tanto en esta etapa de extracción de características como en la etapa de clasificación para futuras muestras a ser evaluadas.

3.5. Comparación con otros métodos

Los resultados presentados hasta el momento corresponden a un estudio pormenorizado de los métodos expuestos en este capítulo aplicados a los conjuntos de datos descritos. Estos resultados nos han permitido obtener interesantes conclusiones sobre dichos métodos y obtener un conocimiento adecuado de los mismos para el campo de aplicación bajo estudio. En esta sección se comparan estos métodos con algunos de los descritos en la revisión bibliográfica.

Observando las diferentes señales procedentes de los conjuntos de datos considerados podemos ver que todos los métodos que tratan de parametrizar las señales como un conjunto de exponenciales, como son Padé-Z o el expuesto como ridge regression no son válidos para los conjuntos de datos considerados en esta tesis, pues las señales obtenidas distan mucho de la hipótesis de partida de estos métodos. Por otro lado, no se ha considerado el espacio de fase dentro de esta comparación. Las únicas referencias bibliográficas que han sido expuestas en el capítulo anterior no hacen posible la variación del número de puntos a considerar y como se comentó en el capítulo anterior, existen referencias como [Röck08] que desaconsejan su uso. Por tanto, en esta sección de los métodos considerados en el estado del arte se han utilizado los modelos AR, por ser más fáciles de ajustar que los modelos ARMA, el análisis por componentes principales (PCA) y el análisis discriminante lineal (LDA). La comparación de dichos métodos debe ser más exhaustiva que en la sección anterior, donde el objetivo era obtener conclusiones sobre los métodos propuestos. En esta sección, para realizar la comparación se realiza la división de los conjuntos de entrenamiento y test treinta veces de forma aleatoria, pero asegurando las proporciones entre clases tanto en el conjunto de entrenamiento como en el de datos, y se han obtenido las medias de los resultados analizando para dos casos. En el primero de ellos se consideran muy pocos puntos para comprobar la hipótesis de la sección anterior, según la cuál el método de FKR tiene un buen comportamiento en esta situación. En el segundo caso se ha considerado un número medio de puntos para comprobar el funcionamiento de los mismos. En la tabla (3.16) se muestra la media del resultado de acierto en clasificación utilizando de nuevo para tal fin un clasificador consistente en una SVM lineal. Indicar que para los méto-

dos wavelet y FKR se han utilizado para cada conjunto los filtros y kernels que habían obtenido mejores resultados en las tablas de la sección anterior.

Conjunto	Método	Número de Puntos		Conjunto	Método	Número de Puntos	
Alcoholes		d=7	d=15	CSIC		d=7	d=13
	Wavelet	0.754	0.813		Wavelet	0.513	0.56
	FKR	0.828	0.750		FKR	0.659	0.617
	PCA	0.767	0.808		PCA	0.507	0.569
	LDA	0.715	0.749		LDA	0.549	0.603
	AR	0.6	0.65		AR	0.44	0.47
Blancos		d=8	d=16	FIA		d=8	d=14
	Wavelet	0.671	0.685		Wavelet	0.852	0.911
	FKR	0.765	0.773		FKR	0.904	0.847
	PCA	0.836	0.86		PCA	0.898	0.818
	LDA	0.748	0.789		LDA	0.716	0.722
	AR	0.557	0.547		AR	0.672	0.729
Tintos		d=8	d=16	CV		d=10	d=40
	Wavelet	0.771	0.801		Wavelet	0.739	0.793
	FKR	0.773	0.912		FKR	0.843	0.664
	PCA	0.7535	0.881		PCA	0.795	0.780
	LDA	0.776	0.785		LDA	0.640	0.657
	AR	0.457	0.5		AR	0.4	0.437

Tabla 3.16: Comparación de acierto en clasificación entre los diferentes métodos

Del análisis de los resultados de la tabla (3.16) se pueden obtener varias conclusiones interesantes. En primer lugar se destaca el mal comportamiento de los modelos autorregresivos (AR) en todos los casos. Tal como se destacaba en el capítulo anterior, este tipo de métodos son válidos cuando la información de clasificación se encuentra en el dominio espectral de la señal y nos encontramos ante procesos estacionarios. Sin embargo, en los conjuntos considerados en esta tesis la información se encuentra en la forma de onda en el tiempo. Junto con los métodos propuestos en este capítulo, el análisis por componentes principales presenta unos resultados muy competitivos. Hay que tener presente que la información contenida en esta tabla reflejan la media de la exactitud en el acierto, pero un análisis más riguroso desde un punto de vista estadístico tiene que considerar si las diferencias presentadas en la tabla son significativas. Para completar el análisis se ha realizado un T-test por pares con significancia del 5 %. Si el test tiene un parámetro de probabilidad por debajo de $p = 0{,}05$ indica que podemos suponer que estadísticamente existe diferencia entre un método y otro, mientras que si el valor de probabilidad es más bajo, aunque se hayan presentado diferencias en la tabla

(3.16) indicará que no podemos afirmar que un método sea mejor que el otro. Este tipo de situaciones se ha destacado en negrita en la tabla. De este análisis se ha omitido el método AR por ser trivial al analizar los resultados afirmar que su comportamiento es peor que el resto de métodos.

Conjunto	d=7				Conjunto	N=15			
Alcoholes	Método	FKR	PCA	LDA	Alcoholes	Método	FKR	PCA	LDA
	Wavelet	1	**0**	1		Wavelet	1	**0**	1
	FKR		1	1		FKR		1	**0**
	PCA			1		PCA			1
Conjunto	d=8				Conjunto	d=16			
Blancos	Método	FKR	PCA	LDA	Blancos	Método	FKR	PCA	LDA
	Wavelet	1	1	1		Wavelet	1	1	1
	FKR		1	**0**		FKR		1	1
	PCA			1		PCA			1
Conjunto	d=8				Conjunto	d=16			
Tintos	Método	FKR	PCA	LDA	Tintos	Método	FKR	PCA	LDA
	Wavelet	**0**	**0**	**0**		Wavelet	1	1	**0**
	FKR		**0**	**0**		FKR		**0**	1
	PCA			**0**		PCA			1
Conjunto	d=7				Conjunto	d=13			
CSIC	Método	FKR	PCA	LDA	CSIC	Método	FKR	PCA	LDA
	Wavelet	1	**0**	**0**		Wavelet	1	**0**	1
	FKR		1	1		FKR		1	**0**
	PCA			**0**		PCA			1
Conjunto	d=7				Conjunto	d=13			
FIA	Método	FKR	PCA	LDA	FIA	Método	FKR	PCA	LDA
	Wavelet	1	1	1		Wavelet	1	1	1
	FKR		**0**	1		FKR		1	1
	PCA			1		PCA			1
Conjunto	d=7				Conjunto	d=13			
CV	Método	FKR	PCA	LDA	CV	Método	FKR	PCA	LDA
	Wavelet	1	1	1		Wavelet	1	**0**	1
	FKR		1	1		FKR		1	1
	PCA			1		PCA			1

Tabla 3.17: T-test por pares para comprobar diferencias estadísticas

La tabla (3.17) complementa la información proporcionada por la tabla de medias, dando la información conjunta de ambas tablas de la que se pueden sacar conclusiones muy interesantes. Para el conjunto de alcoholes cuando $d = 7$ el método que mejor se comporta es FKR, mientras que para $d = 15$ los métodos que mejor se comportan son PCA y la transformada wavelet, sin que existan diferencias estadísticas significativas entre ambos. Para el conjunto de vinos blancos el método que mejor se comporta es PCA en ambos casos, mientras que para el conjunto de vinos tintos para $d = 8$ no existen diferencias estadísticas entre ninguno de los métodos y para $d = 16$ los mejores resultados se obtienen para PCA y FKR. En el caso del conjunto del CSIC para $d = 7$ el mejor método es el FKR y para $d = 13$ se obtendrían los mejores resultados con este mismo método y FKR. Para el conjunto FIA, cuando el número de puntos es bajo los mejores métodos son PCA y FKR, mientras que para $d = 14$ los mejores resultados los obtenemos para la transformada wavelet. Finalmente, para el conjunto de ciclovoltiamperogramas para $d = 10$ los mejores resultados los obtenemos con FKR, mientras que para $d = 40$ los mejores métodos son PCA y la transformada wavelet. Estos resultados destacan el buen comportamiento de los métodos propuestos y de PCA, no siendo en muchos casos significativas las diferencias entre los métodos. Sin embargo, hay que advertir que para los resultados de las tablas (3.16 y 3.17) no se ha tenido en cuenta un número elevado de puntos, donde el método FKR funciona sensiblemente peor. Frente a las ventajas que proporciona PCA como método de extracción figuran su velocidad para obtener el modelo de preprocesado y ser un método robusto cuyos resultados se han mostrado satisfactorios en la mayoría de los conjuntos de datos. La desventaja de este método es el número de operaciones que requiere en la fase de test y la información que se debe almacenar en memoria para dicha fase, lo que puede suponer un problema para sistemas autónomos de tiempo real. Los métodos propuestos en este capítulo han demostrado su buen comportamiento y requieren menos espacio de memoria y operaciones que PCA. Sin embargo, es necesario hacer un proceso de selección del tipo de filtro o tipo de kernel. En el caso de FKR el tiempo de entrenamiento es varios órdenes superior al resto de métodos y su comportamiento es muy sensible a que consiga encontrar un buen conjunto de datos sobre los que proyectar la transformación. Como desventaja adicional, este método presenta un rendimiento no aceptable para un número de puntos elevado.

3.6. Resumen de las aportaciones realizadas en el capítulo

En este capítulo se han presentado dos métodos de extracción de la información para señales dinámicas de sensores de gases y líquidos. La motivación de investigar en

este tipo de métodos fue resultado de la revisión bibliográfica sobre los métodos en el estado del arte, donde se encontraban métodos de extracción muy específicos para señales exponenciales pero que no se podían hacer extensivos a otro tipo de señales cada vez más utilizada en los sistemas de nariz y lengua electrónicas. Respecto a los métodos que sí son aplicables a todo tipo de señales, sobre algunos estaba documentado su mal comportamiento y sobre los modelos AR, en el capítulo anterior se formuló la hipótesis bajo la cuál estos métodos no tendrían por qué proporcionar buenos resultados.

El primer método propuesto es una extensión de la transformada wavelet, siendo ésta muy utilizada como método de extracción de características en los campos de la nariz y lengua electrónicas. Sin embargo, la selección de los coeficientes se hacía de forma no automatizada. La extensión del método incluye aplicar filtros biortogonales que permiten la extensión periódica simétrica de las señales y la transformada wavelet packet que permite la descomposición de los detalles de las sucesivas etapas de filtrado. Ambas ideas han sido ampliamente utilizadas en el campo de la compresión de señales, pero para poder ser utilizadas en este caso se ha tenido que considerar que trabajamos con conjuntos de señales siendo el objetivo final su clasificación. Para poder lograr este objetivo final se ha presentado un método de descomposición y selección de coeficientes **común** para el conjunto de señales, tanto por energía como por separabilidad de las clases.

El segundo método se basa en la regresión no lineal a partir de funciones kernel. Sin embargo, como lo que se pretende es reducir el número de coeficientes donde se almacena la información, dicha regresión tiene que ser realizada a partir de un conjunto de puntos dado. El método se aplicaba para otro tipo de problemas utilizando un kernel gaussiano y en esta tesis se ha ampliado dicho método a otro tipo de kernels, para lo que se tuvo que desarrollar un algoritmo de búsqueda de los puntos óptimos sobre los que proyectar la aproximación de la transformación. Como aportación adicional, se ha desarrollado la matemática necesaria para quedarnos con el valor de unos coeficientes que son utilizados como parámetros de entrada a un sistema de clasificación. Este método ha demostrado proporcionar muy buenos resultados cuando el número de puntos sobre el que se estima la transformación es bajo. Tanto este método como el basado en la transformada wavelet se han probado sobre conjuntos de datos variados procedentes de sistemas de nariz y lengua electrónicas.

Finalmente se aporta una comparación, con análisis estadístico, sobre el comportamiento de los métodos reflejados en el estado del arte y los métodos propuestos.

Capítulo 4

Métodos de Clasificación

Una vez vistos los métodos de extracción de características de las señales dinámicas proporcionadas por los sensores de gases y líquidos, en este capítulo se examinará el rendimiento de los diferentes métodos de clasificación expuestos en el estado del arte, tanto en lo que respecta al error de clasificación como al número de operaciones que son necesarias realizar para evaluar futuras muestras. Además de la comparación de los métodos ya expuestos, se introducen en este capítulo nuevos clasificadores que están teniendo un gran auge en el campo del reconocimiento de patrones. Esta comparación entre clasificadores es de gran importancia teniendo en cuenta que muchos de los trabajos publicados en el campo de la nariz y la lengua electrónica hacen uso de un único método de clasificación, que a menudo resulta inadecuado.

En esta etapa de los sistemas de nariz y lengua electrónica, la información de los sensores, bien mediante las señales RAW o la extracción de parámetros de las mismas, se presenta al clasificador en forma de vectores o patrones y como se expuso en la revisión bibliográfica, cada una de las componentes del vector será denominada característica. En el caso de aquellos sistemas en los que la información proviene de diferentes sensores, se podrá optar por dos estrategias diferentes: la primera es considerar sistemas multivectoriales, mientras que la segunda consiste en concatenar la información formando un solo vector.

Es importante destacar que en esta tesis se ha decidido trabajar con clasificadores cuyo método de entrenamiento es supervisado, esto es, se conoce en la fase de entrenamiento la clase a la que pertenece cada muestra. Puesto que los conjuntos de datos a analizar, descritos en el capítulo anterior, proceden de experimentos realizados en laboratorio, cada muestra está perfectamente etiquetada, por lo que la elección de métodos supervisados resulta adecuada.

4.1. Estimación del rendimiento de un clasificador

En el campo del reconocimiento de patrones, existe el principio de no superioridad a priori de ningún método de clasificación (*No free lunch theorem*) [Duda00], por lo que para una determinada aplicación se deberá estudiar cuál es el método de clasificación que proporciona mejores resultados. Uno de los factores decisivos a la hora de seleccionar un clasificador es el error obtenido para futuras muestras, o visto de otro modo, la precisión (*accuracy*) en la asignación de la clase correspondiente. Para poder determinar dicho error, se debería contar con un número infinito de muestras de test, lo que es imposible en la práctica, por lo que la decisión del clasificador a utilizar debe estar basada en algún estimador del error o de la precisión del método de clasificación.

En esta sección se exponen algunos de los métodos de evaluación del error de un clasificador que pueden ser aplicados de forma común a todos los clasificadores. Antes de exponer dichos métodos se destacan las siguientes consideraciones:

- Los problemas de clasificación a estudiar son problemas multi-clase. En este capítulo se utilizan algunos métodos de clasificación que tratan este tipo de problemas considerando todas las clases al mismo tiempo, mientras que para otros métodos los clasificadores solucionan problemas binarios de clasificación. Para estos clasificadores en los que se divide el problema en diversos clasificadores binarios, se puede optar por distintas estrategias, como son el uno contra todos, uno contra uno o clasificación jerárquica. En el siguiente capítulo se aborda en profundidad todas estas estrategias mientras que en este capítulo se ha utilizado una estrategia uno contra todos por ser la más común.

- Se debe tener presente que los conjuntos de datos proceden de experimentos reales, por lo que están limitados en número de muestras. La obtención de cada nueva muestra implica un nuevo ensayo de laboratorio, por lo que contar con conjuntos de datos muy extensos será inviable desde un punto de vista práctico.

- Cada clasificador tendrá un conjunto de parámetros $\boldsymbol{\theta}$, que deben ser ajustados en el entrenamiento.

- Los resultados obtenidos deberán ser validados con métodos estadísticos para comprobar la mejor elección en cada caso de un determinado clasificador.

Para facilitar la notación en toda la sección se define la función de decisión obtenida por un clasificador, con un conjunto de entrenamiento $\boldsymbol{X}$, con parámetros asociados $\boldsymbol{\theta}$ y evaluada sobre una muestra $\boldsymbol{x}$ como:

$$f^{X}(\boldsymbol{x}) = f_{(\boldsymbol{\theta},\boldsymbol{X})}(\boldsymbol{x}) \tag{4.1}$$

y se define la función de error para una muestra como:

$$u_i^X = \begin{cases} 1 & \text{Si } f^X(\boldsymbol{x_i}) = y_i \\ 0 & \text{Si } f^X(\boldsymbol{x_i}) \neq y_i \end{cases} \tag{4.2}$$

4.1.1. Validación Contra Conjunto de Test Externo

Este método se suele denominar en la literatura como *hold-out* [Devijver82]. Partiendo de un conjunto de patrones $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$, el método de validación contra un conjunto de test externo consiste en dividir en conjunto $\boldsymbol{X}$ en dos conjuntos, uno de entrenamiento $\boldsymbol{X}^{Train} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_l\}$ con l muestras y otro de test $\boldsymbol{X}^{Test} = \{\boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, ..., \boldsymbol{x}_p\}$ con un número de muestras $p = n - l$. Partiendo del conjunto de entrenamiento se obtiene un clasificador $f_{(\boldsymbol{\theta}, \boldsymbol{X}^{Train})}(\boldsymbol{x})$ y se estimará el error $\hat{\varepsilon}$ mediante la expresión

$$\hat{\varepsilon} = \frac{1}{p} \sum_{i=1}^{p} u_i^{X^{Train}} \quad \boldsymbol{x}_i \in \boldsymbol{X}^{Test} \tag{4.3}$$

En los conjuntos de datos utilizados en esta tesis, se ha observado que los patrones obtenidos provienen de experimentos realizados en distintos días. Para evitar que el conjunto de entrenamiento o el de test representen solo a determinados experimentos, se seleccionan los patrones de entrenamiento y los de test aleatoriamente, de forma que la distribución de clases de ambos conjuntos sea la misma que el conjunto total $\boldsymbol{X}$.

El problema de este método consiste en el sesgo que se puede producir al estimar el error contra un conjunto de test particular. Este problema viene determinado por cuán significativo, desde un punto de vista estadístico, es el conjunto de test. Como se ha mencionado, por la ley de los grandes números $\hat{\varepsilon} = \varepsilon$ cuando $p \to \infty$, pero en la práctica se deberá tener en cuenta la severa limitación en el número de muestras disponibles.

Normalmente, existe una variación del método propuesto, muy utilizada en el campo de las redes neuronales. Dicha variación consiste en trabajar con tres conjuntos: uno de entrenamiento, uno de test y otro de validación. Así, se ajustarán los parámetros del clasificador con el conjunto de validación y se comparan los resultados obtenidos con el conjunto de test. Sin embargo, de nuevo aparece como problema en las aplicaciones objeto de esta tesis la limitación en número de muestras.

4.1.2. Validación Cruzada

Este método se suele describir en la literatura como *cross-validation* o *k-fold cross-validation* [Devijver82]. Partimos del mismo conjunto de patrones que en el caso anterior $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$. El método consiste en efectuar k particiones del conjunto $\boldsymbol{X}$, de forma que tendremos $k-1$ particiones de tamaño $\lceil n/k \rceil$ y una partición de tamaño

$n-((k-1)\lceil n/k\rceil)$. El clasificador se entrena k veces, cada una de ellas dejando una de las particiones para el test y usando el resto para formar el conjunto de entrenamiento. El error estimado $\hat{\varepsilon}$ se obtiene como la media de los errores de cada clasificador. En [Luntz69] se demuestra que bajo condiciones de estabilidad, es decir, si los clasificadores obtenidos en los k entrenamientos son los mismos, la validación cruzada es una estimación insesgada del error de clasificación. Esta suposición será más débil generalmente a medida que disminuya k, pues los clasificadores formados no contemplan todas las muestras del conjunto de entrenamiento y sus funciones de decisión difieren.

Un caso extremo del método de validación cruzada lo constituye el método denominado *leave-one-out* (LOO). En este caso se hacen tantas particiones como datos tiene el conjunto de entrenamiento. Cada clasificador se construye con todas las muestras menos con una que se deja para el test. El error estimado se calcula según

$$\hat{\varepsilon} = \frac{1}{n}\sum_{i=1}^{n} u_i^{X_{/i}^{Train}} \tag{4.4}$$

donde $\boldsymbol{X}_{/i}^{Train}$ representa el conjunto de entrenamiento sin la muestra i. La ventaja del método leave-one-out es que los clasificadores construidos son muy similares al clasificador construido con todo el conjunto de entrenamiento, por lo que se puede suponer que estamos en un caso cercano a la estabilidad y constituye un estimador casi insesgado del error. La desventaja que presenta este método es su alto coste computacional, ya que para la estimación del rendimiento se deberán realizar n entrenamientos.

4.1.3. Error Bootstrap

En la sección anterior se describía cómo el error leave-one-out constituye un estimador casi insesgado del error de generalización. Sin embargo este estimador del error como estadístico tiene una gran varianza. Con el objetivo de reducir esta varianza en [Efron83] se propone otro método de estimación del error de forma que se reduce la varianza. El método bootstrap se basa en entender que el conjunto de entrenamiento $\boldsymbol{X}$ con los n patrones vistos en el caso anterior se genera a partir de un modelo de probabilidad F desconocido. Para poder aplicar el método bootstrap supondremos un modelo de probabilidad estimado $\widehat{F}$ basado exclusivamente en los patrones del conjunto observado $\boldsymbol{X}$. Cada subconjunto bootstrap $\boldsymbol{X}^*$ es un nuevo conjunto de l elementos generados a partir de las muestras de $\boldsymbol{X}$ con reemplazo.

Para poder entender mejor el concepto de subconjunto bootstrap supongamos que tenemos un conjunto de entrenamiento de $n = 10$ patrones $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{10}\}$. Supondremos que los subconjuntos bootstrap tienen un tamaño de $l = 5$ elementos. Así, podríamos tener los siguientes subconjuntos:

$$\begin{aligned}\boldsymbol{X}_1^* &= \{\boldsymbol{x}_2, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_6, \boldsymbol{x}_8\}\\ \boldsymbol{X}_2^* &= \{\boldsymbol{x}_1, \boldsymbol{x}_1, \boldsymbol{x}_1, \boldsymbol{x}_4, \boldsymbol{x}_{10}\}\\ \boldsymbol{X}_3^* &= \{\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_7, \boldsymbol{x}_8\}\end{aligned} \tag{4.5}$$

se puede observar cómo cada patrón de entrenamiento puede estar varias veces contenido en un subconjunto bootstrap o no formar parte del mismo. Además, podemos darnos cuenta cómo pueden existir patrones, en nuestro ejemplo $\boldsymbol{x}_5$ y $\boldsymbol{x}_9$, que no formen parte de ningún subconjunto bootstrap.

El método bootstrap permite estimar estadísticos de cualquier orden, pudiendo conocer el sesgo y la varianza del estimador del estadístico.

La estimación del error de cada subconjunto bootstrap se obtiene mediante:

$$\hat{\varepsilon} = \frac{1}{l-h}\sum_{j=1}^{l-h} u_i^{X^*},\ \boldsymbol{x}_i \notin \boldsymbol{X}^* \tag{4.6}$$

donde h es el número de patrones que, al menos una vez, forman parte del conjunto de entrenamiento $\boldsymbol{X}^*$. El proceso se reitera formando un nuevo subconjunto bootstrap en cada iteración. Así, se obtiene el conocido como error e_0 mediante

$$\hat{\varepsilon}_{eo} = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{l-h_i}\sum_{j=1}^{l-h_i} u_j^{X_i^*},\ \boldsymbol{x}_j \notin \boldsymbol{X}_i^* \tag{4.7}$$

donde B es el número de subconjuntos bootstrap considerados. La estimación ideal del error bootstrap consistiría en todas las posibles combinaciones del conjunto $\boldsymbol{X}$ de l elementos tomadas con repetición. En la práctica es suficiente con tomar $B = 200$ subconjuntos bootstrap.

Una modificación del error anterior lo constituye el denominado error .632b [Efron86]. La probabilidad de que un patrón $\boldsymbol{x}_i$ se encuentre en un subconjunto bootstrap $\boldsymbol{X}_j^*$ es de $1-(1-1/l)^l$, valor que se puede aproximar como 0,632 para l grandes. Por tanto, el error bootstrap 632b se calcula como

$$\hat{\varepsilon}_{632b} = \frac{1}{B}\sum_{i=1}^{B}\left(0{,}632\left(\frac{1}{l-h_i}\sum_{j=1}^{l-h_i} u_j^{X_i^*}\right) + 0{,}368\left(\frac{1}{l}\sum_{k=1}^{m} u_k^{X_i^*}\right)\right)\ \boldsymbol{x}_j \in X_i^*, \boldsymbol{x}_k \in \boldsymbol{X} \tag{4.8}$$

Es decir, para cada subconjunto bootstrap se calcula el error e_0 y se pondera por un factor de 0,632 y por otro lado se calcula el error sobre el conjunto $\boldsymbol{X}$ que engloba a todas las muestras, ponderándose este error por un factor de 0,368. Este último error definido como:

$$\hat{\varepsilon}_X = \frac{1}{l}\sum_{k=1}^{l} u_k^{X^*}\ \boldsymbol{x}_k \in \boldsymbol{X} \tag{4.9}$$

mide el error sobre el conjunto completo, de forma que también considera el error que comete el clasificador sobre el propio conjunto de entrenamiento. Sin embargo, para clasificadores con riesgo de sobreaprendizaje, el error sobre las muestras de entrenamiento tenderá a cero, por lo que la ecuación (4.9) tenderá a decrecer y el error $\hat{\varepsilon}_{632b}$ tenderá a ser un valor infraestimado del valor real. Para evitar este problema en [Efron97] se propone una rectificación del anterior denominada 632+ la cuál se calcula como:

$$\hat{\varepsilon}_{632+} = \hat{\varepsilon}_{632b} + \left(\hat{\varepsilon}_{e0} - \hat{\varepsilon}_X\right) \frac{0{,}2325\hat{R}}{1 - 0{,}368\hat{R}} \tag{4.10}$$

donde $\hat{R}$ se calcula como:

$$\hat{R} = \frac{\left(\hat{\varepsilon}_{e0} - \hat{\varepsilon}_X\right)}{\left(\hat{\gamma} - \hat{\varepsilon}_X\right)} \tag{4.11}$$

siendo $\hat{\gamma}$ un estimador de la forma:

$$\hat{\gamma} = \frac{1}{l^2} \sum_{i=1}^{l} \sum_{j=1}^{l} u_i^{X_j^*} \tag{4.12}$$

Hay que hacer notar que cualquiera de los errores basados en el método bootstrap es costoso desde un punto de vista computacional.

4.2. Metodología de comparación

La comparación de varios clasificadores implica ajustar los parámetros de cada uno de ellos y realizar una estimación del error. En [Duda00] se señala como sesgada la comparación que se establece contra un único conjunto de test externo si éste es utilizado también para ajustar los parámetros de un clasificador. Como se ha mencionado, el método más utilizado para realizar una comparación de clasificadores consiste en dividir en tres partes el conjunto de datos, formando los conjuntos de entrenamiento, test y validación.

Sin embargo, en el caso de las aplicaciones de nariz y lengua electrónica, como el número de patrones es reducido la división en tres partes del conjunto de datos puede llevar a conclusiones no contrastadas desde un punto de vista estadístico. En [Guyon98] se detalla cuál debe ser el tamaño mínimo del conjunto de test o validación para poder tener un intervalo de confianza adecuado. Así, supongamos dos clasificadores f_1 y f_2 obtenidos como resultado de entrenar con el conjunto $\boldsymbol{X}^{Train}$ con diferentes métodos de clasificación. Estos clasificadores tendrán asociados unos errores sobre el conjunto de test $\boldsymbol{X}^{Test}$ que denominamos $\hat{\varepsilon}_1$ y $\hat{\varepsilon}_2$ y una diferencia entre ellos $\widehat{\Delta\varepsilon} = \hat{\varepsilon}_1 - \hat{\varepsilon}_2$. Siguiendo el mencionado trabajo, para poder decir que un clasificador es superior al otro, con al

menos un 95 % de confianza, el número de muestras del conjunto de test necesarias es de:

$$p = \frac{10\varepsilon}{\widehat{\triangle\varepsilon}^2} \tag{4.13}$$

donde ε es una media de los errores estimados. De la expresión anterior, podemos suponer que para clasificadores con una estimación del error inferior al 10 %, para que una variación entre dos clasificadores de 0,01 sea significativa con un intervalo de confianza del 95 %, el número de muestras necesarias en el conjunto de test es de $p \approx 10{,}000$.

Aunque las condiciones del ejemplo puedan variar, la expresión anterior nos sirve para determinar que, para poder comparar estadísticamente dos clasificadores solamente sometidos a un único test, el número de muestras en el conjunto de test debe ser suficientemente elevado. El mismo razonamiento se puede aplicar para deducir que los parámetros $\boldsymbol{\theta}_1$ generalizan mejor que los parámetros $\boldsymbol{\theta}_2$.

El ejemplo anterior refuerza las recientes recomendaciones que se están realizando en varios trabajos relevantes [Demsar06] sobre realizar comparaciones con un único conjunto de test cuando el número de patrones de dicho conjunto no es elevado. Como se ha insistido a lo largo del capítulo, esta es precisamente la situación en aplicaciones de nariz y lengua electrónica, por lo que a la hora de realizar una comparación de métodos de clasificación y presentación de resultados se debe diseñar un método que contemple estos principios. No son admisibles por tanto los resultados presentados en múltiples publicaciones, llegando a considerar en algunos casos conjuntos de test de menos de cinco elementos.

La metodología de comparación que se propone en esta tesis se compone de experimentos sobre el conjunto $\boldsymbol{X}$. Cada experimento consistirá de los siguientes pasos:

1. Dividir el conjunto de datos $\boldsymbol{X}$ en dos partes $\boldsymbol{X}^{Train}$ y $\boldsymbol{X}^{Test}$ con número de muestras l y p respectivamente, de forma que $p \approx \lceil (1/3)l \rceil$. Esta división será aleatoria, previa variación del orden de los datos, pero se asegurará que los conjuntos $\boldsymbol{X}^{Train}$ y $\boldsymbol{X}^{Test}$ guarden aproximadamente la proporción de clases presente en el conjunto $\boldsymbol{X}$.

2. Para ajustar los parámetros de cada clasificador se realizará mediante el propio conjunto $\boldsymbol{X}^{Train}$ mediante alguno de los siguientes métodos:

   - Validación cruzada con $k = 5$.
   - Bootstrap .632
   - Bootstrap .632+

En esta parte de la tesis se descarta el uso del error leave-one-out por el elevado coste computacional que supone para la mayoría de los clasificadores. Una vez ajustados los parámetros se obtendrán las diferentes funciones de decisión

$$f_i^{X^{Train}}(\boldsymbol{x}) = f_{(\boldsymbol{\theta}_i, \boldsymbol{X}^{Train})}(\boldsymbol{x}) \tag{4.14}$$

Cada una de las funciones de decisión será obtenida mediante un método diferente de clasificación, una vez que sus parámetros han sido ajustados.

3. El conjunto de test $\boldsymbol{X}^{Test}$ se somete a las diferentes funciones de decisión $f^{X^{Train}}(\boldsymbol{x})$ y se guardan los resultados relativos al experimento. En aquellos conjuntos de datos procedentes de señales multisensoriales que no forman un único vector, sino que los patrones son multivectoriales, se concatena la información de los sensores para formar un único vector.

Cada experimento se repite un número $\{L \mid L > 30\}$ de veces, de forma que se pueden establecer test estadísticos sobre los resultados. Además de obtener la media de los resultados en acierto o error, atendiendo a las recomendaciones de comparación de clasificadores, en esta tesis se ha optado por realizar dos tipos de test:

T-test por parejas Este método ya fue empleado en el capítulo anterior de esta tesis y la información que proporciona es indicar si las diferencias encontradas entre las medias son estadísticamente significativas. El hecho de repetir el experimento en un número $L > 30$ hace que dicho test no tenga que presuponer que los resultados tengan distribuciones normales.

Wilcoxon signed-rank test En este caso se trata de un test no paramétrico propuesto en [Wilcoxon45] para establecer comparaciones entre clasificadores y en [Demsar06] se propone como alternativa al t-test por parejas para aquellos casos en los que los postulados de este último no se cumplen. Supuestos dos métodos de clasificación C_1 y C_2, cuyos resultados de tasa de acierto, definida según la ecuación (3.39) para un determinado experimento i sean ρ_1^i y ρ_2^i respectivamente, se determina la diferencia $D_i = \rho_1^i - \rho_2^i$ y se establece una lista de diferencias ordenadas de menor a mayor según el valor absoluto de D_i, definiendo la función $ord(D_i)$ como la posición que ocupa dentro de la lista. Si en dos experimentos i, j se produce una igualdad $|D_i| = |D_j|$ el orden asignado será en ambos casos la mitad de la suma de las posiciones que ocupan dichas diferencias. Se define a continuación los siguientes valores:

$$\begin{aligned} R^+ &= \sum_{D_i>0} ord(D_i) + \tfrac{1}{2} \sum_{D_i=0} ord(D_i) \\ R^- &= \sum_{D_i<0} ord(D_i) + \tfrac{1}{2} \sum_{D_i=0} ord(D_i) \\ T &= min(R^+, R^-) \end{aligned} \tag{4.15}$$

A continuación se establece el estadístico:

$$z = \frac{T - 0{,}25L\left(L+1\right)}{\sqrt{\frac{1}{24}L\left(L+1\right)\left(2L+1\right)}} \tag{4.16}$$

el cuál está distribuido aproximadamente de forma normal. La hipótesis nula, según la cuál no existe diferencia estadística entre ambos clasificadores se rechaza con un nivel de confianza del 95 %, ($\alpha = 0{,}05$) si $z < -1{,}96$. La ventaja de este tipo de test es que el comportamiento excepcionalmente bueno o malo sobre un único conjunto de test no influye tanto en la decisión como en el t-test.

4.3. Implementación de los clasificadores descritos en la revisión bibliográfica

Como se ha comentado anteriormente, cada método de clasificación depende de una serie de parámetros que deberán ser ajustados para proporcionar la función de decisión $f_{(\boldsymbol{\theta},\boldsymbol{X})}\left(\boldsymbol{x}\right)$. Dependiendo del tipo de clasificador, el número de parámetros a ajustar puede resultar excesivo, por lo que en esta sección se describe cómo se han implementado en esta tesis los diferentes clasificadores, qué parámetros se han fijado a un determinado valor y los posibles valores de aquellos parámetros que deben ser ajustados. Todos los clasificadores han sido implementados en lenguaje Matlab para facilitar la metodología de comparación propuesta.

Además de tener en cuenta la exactitud de un clasificador, o de forma equivalente el error que puede introducir ante nuevas muestras, un aspecto muy importante es el número de operaciones que se deben realizar para evaluar una nueva muestra, siendo éste un aspecto será fundamental en el diseño de sistemas en tiempo real. Es importante destacar que en esta parte de la tesis no se da relevancia al número de operaciones necesario para entrenar un clasificador, ya que esta fase de calibración de un modelo se realiza siempre de forma *off-line*.

En esta sección también se describe el número de operaciones necesarias para realizar la evaluación de una muestra en función de los parámetros del modelo, la dimensión del problema d y el número de clases M a considerar. El tiempo necesario para un microprocesador o microcontrolador para evaluar las operaciones descritas dependerá de la arquitectura propia del sistema y de la presencia o no de coprocesadores matemáticos. Por este motivo se ha decidido diferenciar entre cuatro tipos de operaciones básicas a la hora de describir las operaciones necesarias para cada clasificador en la fase de test:

- Comparaciones.

- Sumas
- Multiplicaciones.
- Funciones no lineales.

En el bloque de sumas quedan incluidas las restas, en el de multiplicaciones quedan también incluidas las divisiones y las funciones no lineales serán las exponenciales, sigmoides o tangentes hiperbólicas necesarias para alguno de los clasificadores.

4.3.1. Discriminante lineal de Fisher (FLD)

Se trata de un clasificador construido mediante planos lineales mediante una estrategia uno contra todos. La función de decisión de este clasificador viene dada por:

$$f(\boldsymbol{x}) = \underset{i}{Max}\left(\langle \boldsymbol{x}, \boldsymbol{w}_i \rangle + b_i\right) \;\; i = \{\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_M\} \tag{4.17}$$

donde cada plano lineal $(\boldsymbol{w}_i, b_i)$ es el resultado de maximizar la relación entre las matrices de dispersión interclase $\boldsymbol{S}_B$ e intraclase $\boldsymbol{S}_I$ definidas en la ecuación (2.19), siendo el resultado de dicho plano:

$$\boldsymbol{\omega}_i = \left(\boldsymbol{S}_I^i + \boldsymbol{S}_B^i\right)^{-1} \left(\boldsymbol{\mu}_i^+ - \boldsymbol{\mu}_i^-\right) \tag{4.18}$$

donde $\boldsymbol{\mu}_i^+$ es el vector media de las muestras de entrenamiento $X^{Train} \in \mathcal{C}_i$ y $\boldsymbol{\mu}_i^-$ es el vector media de las muestras $X^{Train} \notin \mathcal{C}_i$. Una vez encontrado el vector $\boldsymbol{\omega}_i$, el desplazamiento del plano se calcula mediante la siguiente expresión:

$$b_i = 0{,}5\left(\boldsymbol{\omega}_i^T \boldsymbol{\mu}_i^+ + \boldsymbol{\omega}_i^T \boldsymbol{\mu}_i^-\right) \tag{4.19}$$

Como se puede observar en las expresiones anteriores, en este caso no existe ningún parámetro a optimizar o prefijado. Sin embargo, es necesario destacar que en la ecuación (4.18), se precisa el cálculo de la matriz inversa de las sumas de las matrices de dispersión. Esta matriz puede ser cercana a la singularidad en varios de los conjuntos propuestos, por lo que es necesario añadir un término, de forma que se calcula

$$\boldsymbol{\omega}_i = \left(\boldsymbol{S}_I^i + \boldsymbol{S}_B^i + \delta \boldsymbol{I}\right)^{-1} \left(\boldsymbol{\mu}_i^+ - \boldsymbol{\mu}_i^-\right) \tag{4.20}$$

siendo δ un número prefijado a 10^{-4} en la implementación que se ha realizado en esta tesis. Este pequeño ajuste es de gran importancia para obtener resultados admisibles mediante este método.

Respecto al número de operaciones en la fase de test, atendiendo a la ecuación (4.17), deben realizase M productos escalares, siendo esta variable el número de clases del problema. Cada producto escalar consta de d multiplicaciones y $d-1$ sumas, siendo

d la dimensión del problema. A las sumas anteriormente descritas habrá que añadir una por el desplazamiento b_i del plano. Finalmente se debe establecer una comparación entre las salidas de la evaluación de cada plano. Resumiendo, el número de operaciones necesarias será:

- Comparaciones $= M - 1$.
- Sumas $= M\,(d-1) + M = M \times d$.
- Multiplicaciones $= M \times d$.
- Funciones no lineales $= 0$.

4.3.2. k-nearest neighbors (kNN)

En este método de clasificación la función de decisión se toma como la clase a la que pertenecen la mayoría de los k patrones de entrenamiento más cercanos respecto a la muestra a evaluar. El entrenamiento de este método únicamente consiste en almacenar la matriz de muestras de entrenamiento $\boldsymbol{X}^{Train}$, siendo una matriz de dimensiones $d \times l$, siendo l el número de muestras de entrenamiento. Dicha matriz se almacena etiquetando cada vector con su correspondiente clase. En este caso, el entrenamiento no se realiza mediante una estrategia uno contra todos, sino que se considera el problema multiclase en su conjunto.

El único parámetro libre del entrenamiento es el número de vecinos a considerar k. Este parámetro se optimiza por medio de los métodos de validación cruzada o mediante validación bootstrap entre los posibles valores $\boldsymbol{k} = \{3, 5, 7\}$. No se considera $k = 1$, ya que es un caso extremo de sobreaprendizaje del conjunto de entrenamiento y tampoco se han considerado valores de $k > 7$, de forma que no puedan tener más peso los vecinos lejanos que los de la propia clase [Duda00].

Para cada muestra de test nueva hay que calcular l distancias euclídeas, correspondiendo l al número de las muestras de entrenamiento consideradas. No es necesario realizar el cálculo de la raíz cuadrada en la distancia euclídea para realizar la comparación, pudiéndose calcular mediante la expresión:

$$\|\boldsymbol{x} - \boldsymbol{y}\|^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle + \langle \boldsymbol{y}, \boldsymbol{y} \rangle - 2\langle \boldsymbol{x}, \boldsymbol{y} \rangle \tag{4.21}$$

El primer término de la expresión anterior es común para todas las distancias, por lo que se puede obviar. El segundo término se calcula como el producto escalar de un vector del conjunto de entrenamiento consigo mismo, por lo que este término puede estar precalculado. Esto significa que cada distancia implicará el cálculo de un producto escalar más una multiplicación y una suma. Una vez calculadas todas las distancias euclídeas se deben realizar las comparaciones pertinentes para ver los

patrones que son más cercanos. Si se hubiera considerado $k = 1$ solo habría que realizar una comparación entre las l distancias obtenidas, mientras que para sucesivos valores de k habrá que realizar una comparación entre los patrones que no se hayan considerado como más cercanos. Por último, hay que realizar una comparación entre todos ellos para determinar la clase a la que pertenece la muestra. En resumen, el número máximo de operaciones necesarias será:

- Comparaciones $= k(l-k) + (k-1)$.
- Sumas $= l \times d$.
- Multiplicaciones $= l(d+1)$.
- Funciones no lineales $= 0$.

Existen implementaciones que pueden llegar a ser más eficientes en el número de operaciones de la fase de test como son el algoritmo K-LAESA [Moreno02] o la propuesta en [Gil07]. Sin embargo, no han sido implementadas en esta tesis ya que la reducción depende del conjunto de test bajo consideración y no puede calcularse a priori dicha reducción.

4.3.3. Perceptrón multicapa (MLP)

Los perceptrones implementados constan de una única capa oculta en las que las funciones de activación $f_h(.)$ son funciones tangente hiperbólica, mientras que para la capa de salida la función de activación $f_o(.)$ se ha elegido una función de activación lineal. La estrategia multiclase se ha abordado bajo un entrenamiento uno contra todos, por lo que cada perceptrón discriminará una clase de las demás. Deben considerarse por tanto M perceptrones, cada uno de los cuales, una vez entrenada la red tendrá una función de decisión descrita por:

$$f_i(\boldsymbol{x}) = f_o\left(\sum_{j=1}^{N_h} \omega_{oj}^i \left(f_h\left(\langle \boldsymbol{\omega}_h^j \cdot \boldsymbol{x}\rangle + b_{hj}\right)\right)\right) + b_i \tag{4.22}$$

Para el entrenamiento de los perceptrones se ha utilizado la toolbox de Matlab [Demuth05] con entrenamiento por descenso del gradiente por *momentum*. El número de neuronas de la capa oculta,N_h, será uno de los parámetros a optimizar en este tipo de clasificadores. Siguiendo los heurísticos de [Ripley93] y [Kanellopoulos97] se acota la búsqueda del número de neuronas mediante el siguiente vector de búsqueda:

$$\boldsymbol{N}_h = \{\lfloor d/2 \rfloor, \lfloor d/2 \rfloor + \lceil \frac{d}{8} \rceil, \lfloor d/2 \rfloor + \lceil \frac{d}{4} \rceil, ..., 2d\} \tag{4.23}$$

El criterio de parada de entrenamiento de la red siempre será el número de iteraciones máximas prefijadas, que será un parámetro a optimizar y cuyo rango de valores estará en función del número de neuronas ocultas consideradas. En la tabla (4.1) se reflejan los valores y rangos de los parámetros fijos y a optimizar en el entrenamiento de cada perceptrón.

Fijos	Nombre del Parámetro	Valor
	Objetivo	0
	Learning Rate	0.01
	Momentum constant	0.9
	Gradiente Mínimo	0
Variables	Nombre del Parámetro	Rango de Valores
	Neuronas ocultas	$\boldsymbol{N}_h = \{\lfloor d/2 \rfloor, \lfloor d/2 \rfloor + \lceil \frac{d}{8} \rceil, \lfloor d/2 \rfloor + \lceil \frac{d}{4} \rceil, ..., 2d\}$
	Iteraciones	$\boldsymbol{It} = \{50N_h, 100N_h, ..., 1000N_h\}$

Tabla 4.1: Parámetros fijos y variables en entrenamiento de MLP.

Para comprender mejor las operaciones necesarias, en la figura (4.1) se representa el esquema de una neurona oculta, donde vemos que debe realizar un producto escalar con la entrada, una función no lineal y la suma del bias. Así, el número de operaciones por cada neurona oculta será:

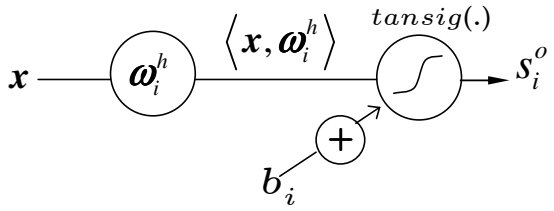


Figura 4.1: Esquema de neurona oculta en un perceptrón multicapa

- Comparaciones = 0
- Sumas = $d - 1 + 1$ (Correspondiente al bias) = d
- Multiplicaciones = d
- Funciones no lineales = 1

Respecto a la capa de salida, en la figura (4.2) se muestra el esquema correspondiente con la función de activación lineal, de forma que las operaciones necesarias son:

- Comparaciones = 0
- Sumas = $N_h - 1 + 1$ (Correspondiente al bias) = N_h
- Multiplicaciones = N_h
- Funciones no lineales = 0

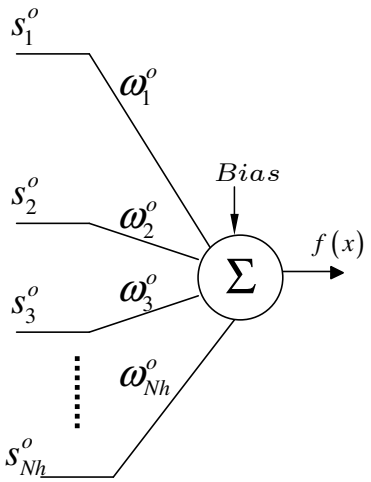


Figura 4.2: Esquema de la capa de salida de un perceptrón

Por tanto, si se consideran M clases en el problema, habrá que construir M perceptrones, cada uno de ellos con un número de neuronas ocultas N_h^i. Esto significa que el número de operaciones necesarias para clasificar una muestra mediante perceptrones multicapa viene dado por:

- Comparaciones = $M - 1$
- Sumas = $\sum_{i=1}^{M} \left(N_h^i d\right) + N_h^i = \sum_{i=1}^{M} N_h^i \left(d + 1\right)$
- Multiplicaciones = $\sum_{i=1}^{M} N_h^i \left(d + 1\right)$
- Funciones no lineales = $\sum_{i=1}^{M} N_h^i$

4.3.4. Redes neuronales de base radial (RBF)

Como se expuso en la sección (2.5.5), las redes de base radial tienen una estructura similar a los perceptrones multicapa de una única capa oculta, pero se diferencian de las anteriores en la función de dicha capa oculta, siendo en las redes de base radial

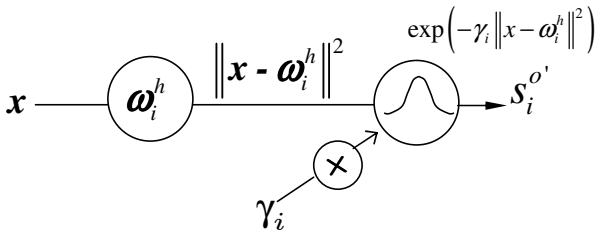


Figura 4.3: Esquema de la capa oculta de una red RBF.

una función $\phi(\boldsymbol{x})$ dependiente de la distancia, euclídea o de Mahalanobis. La función de base radial más popular, adoptada también en esta tesis, es la denominada función gaussiana:

$$\phi(\boldsymbol{x}, \boldsymbol{x'}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x'}\|^2}{2\sigma^2}\right) = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{x'}\|^2\right) \tag{4.24}$$

donde σ^2 es uno de los parámetros que debe ser fijado a priori. La función de decisión, para cada problema binario, una vez entrenada la red quedará:

$$f(\boldsymbol{x})_i = \sum_{j=1}^{N_h^i} \omega_j^o \phi_j(\boldsymbol{x}) + b \tag{4.25}$$

donde

$$\phi_j(\boldsymbol{x}) = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{\omega_j^h}\|^2\right) \tag{4.26}$$

siendo el valor del vector ω_j^h encontrado en el proceso de aprendizaje. A diferencia de las redes MLP, el número de neuronas ocultas N_h se encuentra de forma automática mediante en el entrenamiento.

Para el entrenamiento de las redes RBF también se ha utilizado la toolbox de Matlab [Demuth05] siendo el único parámetro a ajustar el valor de γ indicado en la ecuación (4.26).

Respecto al número de operaciones necesarias para evaluar cada clasificador binario, la capa de salida es igual a la calculada en el caso del perceptrón multicapa, esquematizado en la figura (4.2). Respecto a la capa oculta, cada neurona sigue un esquema como el de la figura (4.3), siendo las operaciones necesarias de esta capa el cálculo de una distancia euclídea al cuadrado, más una multiplicación. Siguiendo la ecuación (4.21) el cálculo de la distancia euclídea puede desarrollarse, tal como se explicó, como tres productos escalares, dos sumas y una multiplicación. Sin embargo, se propone en esta tesis replantear el cálculo de la expresión (4.25), mediante la ecuación:

$$
\begin{aligned}
f(\boldsymbol{x})_i &= \sum_{j=1}^{N_h^i} \omega_j^o \exp\left(-\gamma\langle \boldsymbol{x}, \boldsymbol{x}\rangle - \gamma\langle \boldsymbol{\omega}_j^h, \boldsymbol{\omega}_j^h\rangle + 2\gamma\langle \boldsymbol{\omega}_j^h, \boldsymbol{x}\rangle\right) + b \\
f(\boldsymbol{x})_i &= \exp\left(-\gamma\langle \boldsymbol{x}, \boldsymbol{x}\rangle\right) \sum_{j=1}^{N_h^i} \omega_j^{o'} \exp\left(+2\gamma\langle \boldsymbol{\omega}_j^h, \boldsymbol{x}\rangle\right) + b \\
\omega_j^{o'} &= \omega_j^o \exp\left(-\gamma\langle \boldsymbol{\omega}_j^h, \boldsymbol{\omega}_j^h\rangle\right)
\end{aligned}
\tag{4.27}
$$

El esquema propuesto se muestra en la figura (4.4). La ventaja de este cálculo es que los coeficientes $\omega_j^{o'}$ estarán precalculados y el producto $\exp\left(-\gamma\langle \boldsymbol{x}, \boldsymbol{x}\rangle\right)$ es común a todas las neuronas. Así, la capa oculta tendrá un número de operaciones de:

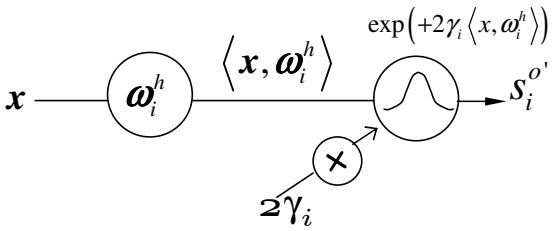
- Comparaciones $= 0$
- Sumas $= d - 1$
- Multiplicaciones $= d + 1$(Correspondiente a γ)
- Funciones no lineales $= 1$

mientras que la capa de salida tendrá un número de operaciones:

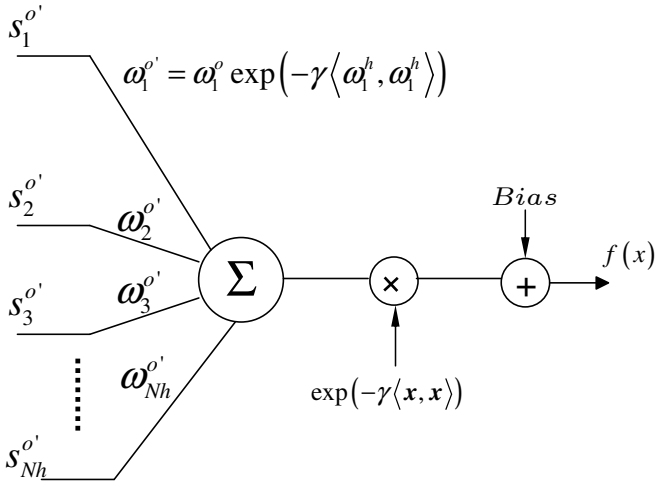
- Comparaciones $= 0$
- Sumas $= N_h - 1 + 1$(Debido al bias)$=N_h$
- Multiplicaciones $= N_h + d + 1$
- Funciones no lineales $= 1$

Supuesto que tenemos M clases y que realizamos una red RBF para separar cada clase del resto y siendo N_h^i el número de neuronas ocultas de la red binaria i, con el esquema propuesto tendremos un total de operaciones:

- Comparaciones $= M - 1$
- Sumas $= \sum_{i=1}^{M} N_h^i + N_h^i(d-1) = \sum_{i=1}^{M} N_h^i d$
- Multiplicaciones $= \sum_{i=1}^{M} N_h^i(d+1) + N_h^i + d + 1$
- Funciones no lineales $= N_h + 1$



(a) *Capa oculta*



(b) *Capa de salida.*

Figura 4.4: Esquema propuesto para evaluación de muestras en red RBF.

4.3.5. Red fuzzy artmap

Como se expuso en el capítulo 2, el objetivo del entrenamiento de una red fuzzy artmap es crear un mapa de N_{FA} neuronas, cada una de ellas con una clase asociada, pudiendo existir varias neuronas a una misma clase, de forma que $N_{FA} \geq M$. Durante la fase de entrenamiento se ajustan los pesos $\boldsymbol{\omega}$ de las neuronas, así como sus desplazamientos b_i. La implementación se realizó siguiendo los pasos descritos en la sección (2.5.4), donde el parámetro más importante a optimizar es el denominado parámetro de vigilancia máxima. Para el resto de parámetros descritos en la tabla (4.2), se comprobó que no afectaban al entrenamiento cuando se optimizaban.

Una vez entrenada una red fuzzy artmap, para determinar la clase a la que pertenece una nueva muestra $\boldsymbol{x}$, por cada neurona habrá que seguir el esquema de la figura (4.5). El complemento de la entrada $\boldsymbol{x} \rightarrow \boldsymbol{v}$, definido según (2.33), solo debe realizarse una vez, y con este complemento se calculan por cada neurona la distancia lógica y el

Fijos	Nombre del Parámetro	Valor
	Máximo Número de Neuronas	2000
	Velocidad de Ajuste	0.9
	Máximas iteraciones	2000
Variables	Nombre del Parámetro	Rango de Valores
	Vigilancia Máxima	$\boldsymbol{\rho} = \{0{,}7, 0{,}72, 0{,}74, ..., 0{,}9\}$

Tabla 4.2: Parámetros fijos y variables en entrenamiento de MLP.

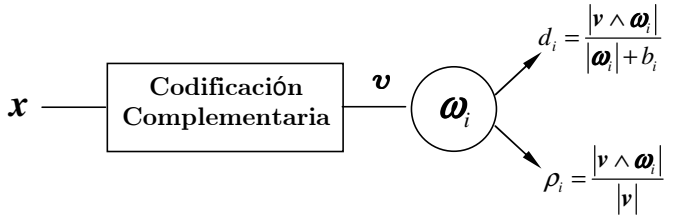


Figura 4.5: Esquema de operaciones por neurona en red fuzzy artmap.

parámetro de vigilancia, definidos según las expresiones (2.35) y (2.34):

$$\begin{aligned} &\boldsymbol{x} \in \mathbb{R}^d \rightarrow \boldsymbol{v} \in \mathbb{R}^{2d} \\ &d_j = \frac{\sum_{k=1}^{2d} \min\left(v_k, \omega_k^j\right)}{\sum_{k=1}^{2d} \omega_k + b_k} \\ &\rho_j = \frac{\sum_{k=1}^{2d} \min\left(v_k, \omega_k^j\right)}{\sum_{k=1}^{2d} v_k} \end{aligned} \tag{4.28}$$

Se asigna la clase del mapfield asignada a la neurona con menor distancia lógica y cuyo parámetro de vigilancia no supere el valor de vigilancia. De esta forma, habrá que realizar una única vez el cálculo del complemento, para lo que son necesarias d sumas y su norma lógica $\|\boldsymbol{v}\|$, lo que implica $2d-1$ sumas adicionales. Además, por cada neurona se debe calcular el término $|\boldsymbol{v} \wedge \boldsymbol{\omega}_i|$ lo que implica las siguientes operaciones:

- Comparaciones = $2d$.
- Sumas = $2d - 1$.
- Multiplicaciones = 2 (Correspondientes a las divisiones).
- Funciones no lineales = 0.

De forma que se supone la norma de los pesos $|\boldsymbol{\omega}_i| + b_k$ precalculada. El número de operaciones totales, supuestas N_{FA} neuronas en la red será de:

- Comparaciones = $N_{FA}2d+2\left(N_{FA}-1\right)$ (El 2 se incluye para comparar la distancia y el parámetro de vigilancia).
- Sumas = $N_{FA}\left(2d-1\right)+3d-1$.
- Multiplicaciones = $2N_{FA}$.
- Funciones no lineales = 0.

4.3.6. SIMCA

En este método, cada muestra nueva es evaluada mediante los M modelos creados, siendo M el número de clases. La creación de un modelo se realiza mediante el análisis de las Pc_j componentes principales de los vectores de entrenamiento pertenecientes a una misma clase, siendo el número de las Pc_j componentes principales el único parámetro necesario a optimizar, entre un rango que ha sido definido como $\boldsymbol{Pc_j} = \{1, 2, .., \lfloor d/4 \rfloor\}$.

Al igual que el resto de los algoritmos de clasificación expuestos, el código fue desarrollado en Matlab con el apoyo de llamadas a las funciones contenidas en [Franc04] para el cálculo del análisis de componentes principales.

Una vez realizado el entrenamiento, para determinar la clase de una nueva muestra $\boldsymbol{x}_i$, para cada modelo PCA se deben obtener los estadísticos $Q_{i,j}$ y $T^2_{i,j}$, definidos en las ecuaciones (2.17) y (2.18), donde el subíndice j indica el modelo PCA bajo consideración. A partir de dichos estadísticos, se obtienen sus valores normalizados $Q_{n(i,j)}$ y $T^2_{n(i,j)}$, descritos en la expresión (2.40) para poder obtener una distancia entre la muestra $\boldsymbol{x}_i$ y el modelo j definida como:

$$d_j^2\left(\boldsymbol{x}_i\right) = Q^2_{n(i,j)} + \left(T^2_{n(i,j)}\right)^2 \tag{4.29}$$

de forma que se asigna la muestra al modelo j cuya distancia es menor. Así, por cada modelo habrá que realizar los siguientes pasos

Cálculo de Q El cálculo del estadístico Q representa la distancia entre la muestra $\boldsymbol{x}$ y la muestra recuperada $\hat{\boldsymbol{x}}$, una vez que se ha aplicado el modelo PCA y se ha eliminado parte de su información. Por tanto, hay que proyectar la muestra actual sobre el modelo PCA de la clase bajo estudio y recuperar la muestra. En primer lugar se debe restar la media de la muestra $\boldsymbol{x}$. Posteriormente se proyectará la muestra sobre el nuevo modelo, lo que implica el cálculo de tantos productos escalares como componentes principales Pc_j tenga el modelo. A continuación se debe recuperar la muestra, lo que implica tantas operaciones como el cálculo de

la proyección. Finalmente el cálculo de la distancia entre ambas muestras implica el cálculo de una distancia euclídea, cuyo coste en operaciones ha sido descrito en otros métodos. Finalmente habrá que sumar una división por normalizar el estadístico Q.

Cálculo de T^2 Este estadístico mide la distancia entre la muestra proyectada y el nuevo centro de coordenadas generado por el modelo PCA. Como se describió en la ecuación (2.18), este estadístico puede ser calculado mediante:

$$T_j^2 = \boldsymbol{t}_j^T * \boldsymbol{\Lambda}_j \tag{4.30}$$

donde $\boldsymbol{t}_j$ es la proyección de la muestra $\boldsymbol{x}$ sobre el modelo j, mientras que $\boldsymbol{\Lambda}_j$ es una matriz diagonal con los autovalores del modelo.

Cálculo de la distancia Los valores anteriores de Q y T deben ser normalizados y elevados al cuadrado para poder realizar su suma y calcular de esta forma su distancia.

Los pasos anteriores, tenidos en cuenta para los diferentes M modelos se traducen en las siguientes operaciones

- Comparaciones $= M - 1$.
- Sumas $= \sum_{j=1}^{M} 2Pc_j\,(d+1) + 2 + 3d$.
- Multiplicaciones $= \sum_{j=1}^{M} (2Pc_j + 3)\,d + 3 + Pc_j$.
- Funciones no lineales $= 0$.

4.3.7. Partial least squares discriminant analysis (PLS-DA)

La descripción detallada de los pasos de entrenamiento ha sido descrita en detalle en la sección (2.5.7). Como parámetro de optimización se busca el número de variables latentes que proporciona un mejor rendimiento por cada clase, variando éste según $\boldsymbol{L_v} = \{1, 2, ..., \lfloor d/4 \rfloor\}$.

Una vez realizado el entrenamiento se obtienen M planos lineales $(\boldsymbol{\omega}_i, b_i)$, de forma que la clase asignada será la que cumpla:

$$\mathcal{C}_i = \underset{i}{Max}\langle \boldsymbol{x}, \boldsymbol{\omega}_i \rangle + b_i \tag{4.31}$$

Respecto al número de operaciones necesarias, se deberán ejecutar M productos escalares con los diferentes planos, por lo que en la fase de test el número de operaciones coincide con el descrito en la sección (4.3.1).

4.3.8. Máquinas de vectores soporte (SVM)

En la revisión del estado del arte se describieron las razones por las que se plantea una de las hipótesis de esta tesis doctoral, según la cuál las SVM son un método adecuado para los problemas de nariz y lengua electrónicas. El entrenamiento se basa en un problema de maximización regularizada del margen de separación de un problema binario, por lo que como se ha descrito, en este capítulo se abordan los problemas multiclase siguiendo una estrategia uno contra todos.

Antes de realizar el entrenamiento, tal como se expuso en la revisión bibliográfica, se deben fijar a priori el tipo de kernel y sus parámetros, si los tuviera, así como la constante de regularización C. Adicionalmente, aunque el problema expuesto en la ecuación (2.60) tiene una solución del vector $\boldsymbol{\alpha}$ con un único mínimo, en el proceso de entrenamiento se suele permitir un pequeño margen de tolerancia ε para considerar que se ha finalizado dicho entrenamiento. Cuando se fija ese parámetro a un valor muy pequeño, se acelera el proceso de aprendizaje y no influye significativamente en los resultados de clasificación. Sin embargo, los parámetros del kernel y la constante de regularización C influyen de manera muy significativa en los resultados obtenidos, por lo que deberán ser optimizados mediante la metodología propuesta.

Una vez realizado el entrenamiento, tendremos M máquinas cada una de ellas con una función de decisión:

$$f_i(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{j=1}^{N_{sv}} \alpha_j^i y_i \kappa\left(\boldsymbol{s}_j^i, \boldsymbol{x}\right) + b^i\right) \tag{4.32}$$

La implementación se realizó mediante una modificación del código proporcionado en [Chang01], que a su vez implementa el algoritmo secuencial de optimización presentado en [Platt98]. Puesto que se trata de un algoritmo secuencial implementado de forma eficiente en lenguaje C++, se optó por modificar el código para desarrollar un interfaz de tipo Mex, de forma que los resultados puedan ser comparados con el resto de clasificadores implementados. En esta tesis, como se explica en la sección (2.5.8), se ha optado por utilizar máquinas L1-SVM, pues el número de vectores soporte que proporcionan, para tasas de acierto similares, es menor que en el caso de las L2-SVM.

Respecto al número de operaciones necesarias dependerá de la función kernel seleccionada. En el caso de seleccionar un kernel lineal, se puede construir un plano de decisión:

$$\boldsymbol{\omega}_i = \sum_{j=1}^{l} \alpha_j^i \boldsymbol{s}_j^i \tag{4.33}$$

de forma que la ecuación (4.32) se convierte en:

$$f_i(\boldsymbol{x}) = \operatorname{sign}\left(\langle \boldsymbol{x}, \boldsymbol{\omega}_i \rangle + b_i\right) \tag{4.34}$$

Así, en caso de seleccionar un kernel lineal, el número de operaciones necesarias se calcula como la comparación de M productos escalares más el desplazamiento, por lo que las operaciones necesarias coinciden con las descritas en la sección (4.3.1).

En el caso de un kernel gaussiano, definido por la expresión:

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2\right) \tag{4.35}$$

cada SVM tiene una función de decisión que se corresponde con la función de decisión de las RBF, por lo que el número de operaciones será el mismo que para este clasificador con un número de neuronas en la capa oculta igual al número de vectores soporte obtenidos.

4.3.9. Random forest

Este tipo de clasificador se basa en crear múltiples árboles de decisión débiles, en el sentido que de forma aislada cada árbol no proporciona un nivel de exactitud bueno, pero teniendo en cuenta la decisión de todos los árboles, ésta se refuerza (*boosting*) proporcionando tasas de acierto de las decisiones mayoritarias adecuadas.

El algoritmo de entrenamiento implementado está basado en la descripción de [Breiman01], para lo que se deben construir un número de árboles N_{Trees} prefijados con anterioridad. Cada árbol se construye mediante los siguientes pasos:

1. Se selecciona un conjunto de entrenamiento $\boldsymbol{X}^*_{Train}$ a partir del conjunto inicial realizando un muestreo con reemplazamiento del mismo. El hecho de seleccionar un conjunto de muestras con reemplazamiento, en lugar de todo el conjunto de entrenamiento, asegura que el sistema no sobreaprenderá aunque se aumente el número de árboles del bosque.

2. Para cada nodo del árbol se seleccionan mty características aleatorias del conjunto $\boldsymbol{X}^*_{Train}$. El parámetro mty es necesario prefijarlo de antemano, de forma que $mty \ll d$.

3. Una vez fijadas las mty características solo se consideran éstas para crear la regla de decisión binaria del nodo del árbol. Dicha regla de decisión divide el conjunto $\boldsymbol{X}^*_{Train}$ en dos conjuntos:

$$\begin{array}{l} X^{*L}_{Train}, \text{ Si } x_m \leq v \\ X^{*R}_{Train}, \text{ Si } x_m > v \end{array} \tag{4.36}$$

   Para cada nodo debe establecerse qué característica x_m es la que discrimina y cuál debe ser su valor de decisión asociado v. La selección de dicha característica se

hace teniendo en cuenta qué decisión es la que minimiza la denominada impureza de Gini (I_G) [Duda00] en los siguientes nodos:

$$\begin{aligned}\triangle(I_G) = N_L\left(1-\sum_{j=1}^{M} P\left(\boldsymbol{x}\in\{\mathcal{C}_j, X^*_{Train}L\}\right)\right)+\\ N_R\left(1-\sum_{j=1}^{M} P\left(\boldsymbol{x}\in\{\mathcal{C}_j, X^*_{Train}R\}\right)\right)\end{aligned} \tag{4.37}$$

donde N_L y N_R es la proporción de muestras del conjunto $\boldsymbol{X}^*_{Train}$ que van al nodo izquierdo o al nodo derecho respectivamente, y $P\left(\boldsymbol{x}\in\{\mathcal{C}_j, X^*_{Train}L\}\right)$ es la proporción de patrones pertenecientes a la clase j que serán discriminados hacia el nodo izquierdo. Suponiendo un problema de clasificación binario, la ecuación (4.37) se minimizaría cuando todos los patrones pertenecientes a cada una de las diferentes clases fueran a nodos distintos.

4. Un nodo se marcará como terminal si ocurre uno de los siguientes casos:

   - El número de patrones del conjunto de entrenamiento que han llegado a ese nodo es uno.
   - Todos los patrones del conjunto de entrenamiento que han llegado a ese nodo pertenecen a la misma clase.
   - El incremento de I_G establecido en la ecuación (4.37) es menor que una cierta cantidad ε.

   En la construcción de los árboles, éstos se dejan crecer hasta que todos los nodos son terminales.

En la fase de test solo existen operaciones de comparación, no pudiéndose calcular el número de operaciones a priori por el hecho de permitir árboles asimétricos de descomposición. Sin embargo, este valor estará acotado por:

$$\text{Comparaciones Máximas} = \sum_{j=1}^{N_{Trees}} 2^{k_j} \tag{4.38}$$

donde k_j es el máximo nivel de descomposición del árbol j.

4.4. Aplicación de otros clasificadores

En esta sección se propone, como aportación de esta tesis, utilizar métodos de clasificación que están dando buenos resultados en otros campos de investigación y no han sido aplicados en la nariz y lengua electrónicas.

4.4.1. Máquinas de vectores soporte por mínimos cuadrados. LS-SVM

Las máquinas de vectores soporte por mínimos cuadrados (LS-SVM del inglés *Least Squares Support Vector Machines*) [Suykens99], siguen una formulación muy similar a las SVM y en concreto se asemejan mucho a la formulación de las L2-SVM expuestas en la expresión (2.62) pero a diferencia de éstas, en las que las condiciones KKT se plantean como una desigualdad, se trata de minimizar la siguiente expresión con restricciones:

$$\begin{aligned} &\min_{\boldsymbol{\omega},b} \tfrac{1}{2}\|\boldsymbol{\omega}\|^2 + \tfrac{1}{2}C\sum_{i=1}^{l}\xi_i^2 \\ &\text{Sujeto a: } \left(\boldsymbol{\omega}^T\phi\left(\boldsymbol{x}_i\right)+b\right) = y_i - \xi_i \end{aligned} \tag{4.39}$$

Se puede apreciar que la única diferencia respecto a la formulación referida de las L2-SVM consiste en sustituir la desigualdad en las condiciones por una igualdad. Así, podemos formar el Lagrangiano:

$$\mathcal{L} = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{1}{2}C\sum_{i=1}^{l}\xi_i^2 - \sum_{i=1}^{l}\alpha_i\left(\boldsymbol{\omega}^T\phi\left(\boldsymbol{x}_i\right)+b-y_i+\xi_i\right) \tag{4.40}$$

Calculando las derivadas parciales del Lagrangiano de la ecuación (4.40), se obtienen las siguientes expresiones:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\omega}} = 0 \;\rightarrow \boldsymbol{\omega} = \sum_{i=1}^{l}\alpha_i\phi\left(\boldsymbol{x}_i\right) \tag{4.41}$$

$$\frac{\partial\mathcal{L}}{\partial b} = 0 \rightarrow \sum_{i=1}^{l}\alpha_i = 0 \tag{4.42}$$

$$\frac{\partial\mathcal{L}}{\partial\xi_i} = 0 \rightarrow \xi_i = \frac{\alpha_i}{C} \tag{4.43}$$

$$\frac{\partial\mathcal{L}}{\partial\alpha_i} = 0 \rightarrow \left(\boldsymbol{\omega}^T\phi\left(\boldsymbol{x}_i\right)+b\right) - y_i + \xi_i = 0 \tag{4.44}$$

Sustituyendo la expresión obtenida en la ecuación (4.41) en cada una de las l ecuaciones obtenidas a partir de la expresión (4.44) obtendremos:

$$\sum_{j=1}^{l}\alpha_j\phi\left(\boldsymbol{x}_j\right)^T\phi\left(\boldsymbol{x}_i\right) + b - y_i + \xi_i = 0 \tag{4.45}$$

Podemos aprovechar una vez más el truco del kernel,

$$\kappa\left(\boldsymbol{x}_j,\boldsymbol{x}_i\right) = \phi\left(\boldsymbol{x}_j\right)^T\phi\left(\boldsymbol{x}_i\right) \tag{4.46}$$

De forma que la expresión (4.45) quedará:

$$\sum_{j=1}^{l} \alpha_j \kappa(\boldsymbol{x}_j, \boldsymbol{x}_i) + b - y_i + \xi_i = 0 \tag{4.47}$$

Sustituyendo la condición (4.43) en la expresión anterior, tendremos que cada una de las l ecuaciones cumplirá:

$$\sum_{j=1}^{l} \alpha_j \kappa(\boldsymbol{x}_j, \boldsymbol{x}_i) + b + \frac{\alpha_i}{C} = y_i \tag{4.48}$$

Aprovechando la ecuación (4.42) tendremos un sistema de $l+1$ ecuaciones lineales con l valores incógnita de los multiplicadores de Lagrange α_i, más el parámetro de desplazamiento b. Por tanto, el hecho de transformar la desigualdad de las L2-SVM en la igualdad expresada en la ecuación (4.39) hace que el problema se transforme en un problema de solución lineal, que puede ser expresado de forma matricial mediante:

$$\begin{bmatrix} \boldsymbol{K} + \frac{1}{C}\boldsymbol{I} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix} \tag{4.49}$$

donde $\boldsymbol{K}$ es la matriz de Gram del conjunto de entrenamiento, $\boldsymbol{I}$ es una matriz identidad $l \times l$ y los vectores descritos como $\mathbf{1}$ serán vectores de unos de dimensión l. La solución de los l valores de los multiplicadores de Lagrange más el desplazamiento b queda descrita por:

$$\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{K} + \frac{1}{C}\boldsymbol{I} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix} \tag{4.50}$$

Una vez entrenado el clasificador, la función de decisión para una nueva muestra puede escribirse como:

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{j=1}^{l} \alpha_j \kappa(\boldsymbol{s}_j, \boldsymbol{x}) + b \right) \tag{4.51}$$

donde los vectores $\boldsymbol{s}_j$ son los vectores de entrenamiento. Como se puede apreciar, dicha función es la misma que para las SVM, por lo que el número de operaciones será el mismo que el descrito anteriormente, considerando como l el número de vectores soporte.

La gran ventaja que aportan las LS-SVM es la velocidad de entrenamiento frente a las L1-SVM clásicas, pues se trata de resolver un problema de $l+1$ ecuaciones lineales frente a un problema cuadrático. Respecto a la densidad de la solución, la expresión (4.50) incluye tantos vectores soporte como patrones de entrenamiento, aunque se puede realizar un proceso de poda de aquellos multiplicadores de Lagrange próximos a cero [Suykens02].

4.4.1.1. Implementación Eficiente de las LS-SVM

Como se ha mencionado anteriormente, la ventaja de las LS-SVM frente a otro tipo de clasificadores no lineales es la velocidad en su entrenamiento. No obstante, la matriz inversa a calcular en la expresión (4.50) no parte de una matriz definida positiva, por lo que no se puede emplear la descomposición de Cholesky para el cálculo de la matriz inversa.

Sin embargo, en [Cawley07] se describe un método eficiente para calcular la inversa mediante la descomposición de Cholesky. Así, partimos de una pseudo-matriz de Gram a la que añadimos el parámetro de regularización C, procedente del problema de optimización (4.39), mediante:

$$\hat{\boldsymbol{K}} = \boldsymbol{K} + \frac{1}{C}\boldsymbol{I} \tag{4.52}$$

dicha matriz sí es definida positiva. Para encontrar la solución de el vector $\boldsymbol{\alpha}$ y el desplazamiento b, primero se resuelve:

$$\begin{aligned} \hat{\boldsymbol{K}}\boldsymbol{\rho} &= \mathbf{1} \\ \hat{\boldsymbol{K}}\boldsymbol{\nu} &= \boldsymbol{y} \end{aligned} \tag{4.53}$$

para posteriormente resolver los vectores deseados mediante:

$$b = \frac{\mathbf{1}^T\boldsymbol{\nu}}{\mathbf{1}^T\boldsymbol{\rho}} \qquad \boldsymbol{\alpha} = \boldsymbol{\nu} - \boldsymbol{\rho}b \tag{4.54}$$

La ventaja de este segundo método es que la matriz de la expresión (4.52) queda descompuesta según:

$$\hat{\boldsymbol{K}} = \boldsymbol{U}^T\boldsymbol{U} \tag{4.55}$$

donde $\boldsymbol{U}$ es una matriz triangular superior. Si definimos $\boldsymbol{M} = \boldsymbol{U}^{-1}$, como la inversa de la matriz triangular superior, ésta última también será triangular superior por lo que su cálculo implica la mitad de operaciones y memoria que el cálculo de una matriz inversa normal. Así podemos escribir:

$$\hat{\boldsymbol{K}}^{-1} = \boldsymbol{M}\boldsymbol{M}^T \tag{4.56}$$

para solucionar la expresión (4.53) de una forma eficiente.

4.4.1.2. Propuesta de aprendizaje incremental

A partir del estudio de la descomposición indicada en la expresión (4.55) se propone en esta tesis un método eficiente para el aprendizaje incremental de las LS-SVM. El aprendizaje incremental supone una herramienta de gran utilidad en los sistemas de

nariz y lengua electrónica, ya que es habitual que, antes de continuar con un número grande de experimentos de laboratorio, se quieran ver los resultados obtenidos a medida que se van ejecutando dichos experimentos. La propuesta en este sentido, tiene como objetivo obtener el nuevo vector $\boldsymbol{\alpha}$ y el parámetro b sin necesidad de tener que volver a entrenar de nuevo.

Supongamos que tenemos una pseudo matriz Gram de l muestras, definida según la expresión (4.52), la cuál referiremos como $\hat{\boldsymbol{K}}^l$, mientras que la matriz triangular superior procedente de la descomposición Cholesky será referida como $\boldsymbol{U}^l$ y su matriz inversa asociada queda determinada por $\boldsymbol{M}^l$. Los vectores asociados, encontrados mediante la expresión (4.53) se definen como $\boldsymbol{\rho}^l$ y $\boldsymbol{\nu}^l$ respectivamente. Si suponemos una muestra adicional para el conjunto de entrenamiento $\boldsymbol{x}_{l+1}$ podemos definir el vector:

$$\boldsymbol{k}^{l+1} = \{\kappa(\boldsymbol{x}_1, \boldsymbol{x}_{l+1})/C, \kappa(\boldsymbol{x}_2, \boldsymbol{x}_{l+1})/C, ..., \kappa(\boldsymbol{x}_l, \boldsymbol{x}_{l+1})/C\} \tag{4.57}$$

La ventaja de la descomposición Cholesky para el aprendizaje incremental es que podemos expresar:

$$\begin{aligned} \boldsymbol{U}^{l+1} &= \begin{bmatrix} \boldsymbol{U}^l & \boldsymbol{u} \\ \boldsymbol{0}^T & u_{l+1} \end{bmatrix} \\ \boldsymbol{M}^{l+1} &= \begin{bmatrix} \boldsymbol{M}^l & \boldsymbol{m} \\ \boldsymbol{0}^T & m_{l+1} \end{bmatrix} \end{aligned} \tag{4.58}$$

donde los elementos del vector $\boldsymbol{u} \in \mathbb{R}^l$ se calculan mediante:

$$u_i = \frac{\boldsymbol{k}_i^{l+1} - \sum_{j=1}^{i-1} U^l(j,i)\, u_j}{U^l(i,i)} \tag{4.59}$$

y el elemento u_{l+1} se calcula mediante:

$$u_{l+1} = \sqrt{\kappa(\boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+1}) - \|\boldsymbol{u}\|^2} \tag{4.60}$$

Para incluir este último elemento dentro del vector $\boldsymbol{u}$, formamos un nuevo vector $\tilde{\boldsymbol{u}} = [\boldsymbol{u}, u_{l+1}]$. Así, el vector $\boldsymbol{m}$ de nuevos elementos de la matriz inversa $\boldsymbol{M}^{l+1}$ puede ser calculado mediante:

$$m_i = \frac{-\sum_{j=i}^{l} M^l(i,j)\, \tilde{u}_j}{\tilde{u}_{l+1}} \tag{4.61}$$

A los que hay que añadir el cálculo del elemento:

$$m_{l+1} = \frac{1}{\tilde{u}_{l+1}} \tag{4.62}$$

Al igual que hicimos con el vector $\boldsymbol{u}$, extenderemos el vector encontrado para incluir este último elemento, de forma que $\tilde{\boldsymbol{m}} = [\boldsymbol{m}, m_{l+1}]$. Si definimos ahora los vectores extendidos:

$$\begin{aligned} \tilde{\boldsymbol{\rho}} &= [\boldsymbol{\rho}^l, 0] \\ \tilde{\boldsymbol{\nu}} &= [\boldsymbol{\nu}^l, 0] \end{aligned} \tag{4.63}$$

Podemos calcular los nuevos vectores $\boldsymbol{\rho}^{l+1}$ y $\boldsymbol{\nu}^{l+1}$ mediante:

$$\begin{aligned} \boldsymbol{\rho}^{l+1} &= \tilde{\boldsymbol{\rho}} + \triangle\tilde{\boldsymbol{\rho}} = \tilde{\boldsymbol{\rho}} + \sum_{i=1}^{l+1} m_i \tilde{\boldsymbol{m}} \\ \boldsymbol{\nu}^{l+1} &= \tilde{\boldsymbol{\nu}} + \triangle\tilde{\boldsymbol{\nu}} = \tilde{\boldsymbol{\nu}} + \sum_{i=1}^{l+1} y_i m_i \tilde{\boldsymbol{m}} \end{aligned} \tag{4.64}$$

y a partir de la expresión (4.54) encontramos los valores de $\boldsymbol{\alpha}^{l+1}$ y de b^{l+1} sin necesidad de tener que entrenar con los l patrones del conjunto de entrenamiento y evitando el cálculo de la descomposición de Cholesky y la matriz inversa asociada. Además, mediante este método, podemos comprender cómo afecta una nueva muestra a la función de decisión completa.

4.4.2. Máquinas de Vectores Relevantes

Las máquinas de vectores relevantes (RVM del inglés *Relevance Vector Machines*) [Tipping01], buscan una función de decisión similar a las funciones de decisión empleadas en las SVM o LS-SVM:

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_{j=1}^{l} \alpha_j \kappa(\boldsymbol{s}_j, \boldsymbol{x}) + b\right) \tag{4.65}$$

donde los diferentes vectores $\boldsymbol{s}_j$ son muestras del conjunto de entrenamiento que han sido consideradas relevantes para el problema de clasificación. Al igual que en el caso de las SVM o las LS-SVM este tipo de clasificadores busca una solución de baja densidad (*sparse*), pero trata de mejorar los anteriores métodos en los siguientes puntos:

1. No hay que ajustar ningún parámetro de regularización a priori, como la constante C en las SVM o en las LS-SVM.

2. La salida que proporcionan tiene un significado probabilístico de pertenencia a una clase, por lo que es posible la comparación entre diferentes máquinas de clasificación empleadas en problemas multiclase.

3. No es necesario que los kernels definan matrices definidas positivas para la solución del problema.

En las SVM o las LS-SVM el entrenamiento se basa en la maximización del margen como elemento de regularización para evitar el sobreaprendizaje, estando definido dicho margen en el espacio de entrada o en un espacio de Hilbert transformado del que podemos calcular su producto interno mediante una función kernel. En las RVM el entrenamiento se basa en un concepto bayesiano en el que las funciones kernel no tienen ninguna interpretación geométrica. Se parte de un problema de clasificación binario cuyas muestras de entrenamiento $\boldsymbol{x_i}$ forman la matriz $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_l}\}$ etiquetadas según $y_i \in \{1, 0\}$. El objetivo de las RVM es encontrar una función $g(\boldsymbol{x})$ tal que:

$$g(\boldsymbol{x}) = \frac{1}{1 + exp(-f(\boldsymbol{x}))} \tag{4.66}$$

donde $f(\boldsymbol{x})$ es la función descrita en la ecuación (4.65). El aprendizaje de las RVM fija su atención en buscar los valores del vector de pesos $\boldsymbol{\alpha}$, contemplados en la función $f(\boldsymbol{x})$, de forma que se maximice a posteriori la $P(\boldsymbol{\alpha}/\boldsymbol{X}, \boldsymbol{y})$.

Siguiendo una distribución de Bernoulli para $P(y_i/\boldsymbol{x}_i)$ y tomando un clasificador $g(\boldsymbol{x}, \boldsymbol{\alpha})$ definimos la relación de verosimilitud como:

$$p(\boldsymbol{y}/\boldsymbol{X}, \boldsymbol{\alpha}) = \prod_{i=1}^{l} g(\boldsymbol{x}, \boldsymbol{\alpha})^{y_i} (1 - g(\boldsymbol{x}, \boldsymbol{\alpha}))^{1-y_i} \tag{4.67}$$

El concepto clave en el aprendizaje de las RVM es suponer que los pesos α_i siguen una distribución normal a priori de media cero y varianza β^{-1}. Así, supuesto que se conoce la precisión (inversa de la varianza) β_i de cada estimador, podemos establecer para el vector de pesos $\boldsymbol{\alpha}$ la siguiente distribución

$$p(\boldsymbol{\alpha}/\boldsymbol{\beta}) = \prod_{i=1}^{n} \mathcal{N}\left(\alpha_i/0, \beta^{-1}\right) \tag{4.68}$$

Puesto que el objetivo de las RVM es encontrar el vector de pesos $\boldsymbol{\alpha}_{MAP}$ que maximiza la probabilidad a posteriori $P(\boldsymbol{\alpha}/\boldsymbol{X}, \boldsymbol{y})$, podemos aplicar la regla de Bayes para deducir que

$$P(\boldsymbol{\alpha}/\boldsymbol{X}, \boldsymbol{y}) \propto P(\boldsymbol{y}/\boldsymbol{X}, \boldsymbol{\alpha}) p(\boldsymbol{\alpha}/\boldsymbol{\beta}) \tag{4.69}$$

es decir, la probabilidad a posteriori es proporcional al producto de una probabilidad a priori por una relación de verosimilitud. En lugar de maximizar la ecuación (4.69), podemos aprovechar que el logaritmo neperiano de la misma es una función monótonamente creciente, por lo que encontrar el vector $\boldsymbol{\alpha}_{MAP}$ que maximiza dicho logaritmo equivale a encontrar el vector que maximiza la ecuación (4.69). Así, la función a maximizar resulta

$$\log\left(P\left(\boldsymbol{y}/\boldsymbol{X},\underset{\boldsymbol{\alpha}}{\boldsymbol{\alpha}}\right)p\left(\boldsymbol{\alpha}/\boldsymbol{\beta}\right)\right)=\sum_{i=1}^{n}\left[y_i\log g\left(\boldsymbol{x_i}\right)+\left(1-y_i\right)\left(1-\log g\left(\boldsymbol{x_i}\right)\right)\right]-\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{B}\boldsymbol{\alpha} \tag{4.70}$$

donde $\boldsymbol{B}$ es una matriz diagonal cuyos elementos $B_{ii}=\beta_i$ son las precisiones (inversas de la varianza) de los estimadores de cada peso α_i. La ecuación (4.70) se puede entender como la optimización de una relación de verosimilitud en forma logarítmica pero regularizada por un parámetro impuesto por suponer diferente la varianza de cada estimador de pesos. El entrenamiento, por tanto es muy similar al realizado en el aprendizaje Bayesiano [Bishop06].

La maximización de la ecuación (4.70) supone conocidos a priori las precisiones β_i, por lo que el problema de encontrar el vector $\boldsymbol{\alpha}_{MAP}$ se plantea como un problema iterativo que engloba los siguientes pasos:

1. Suponer un valor inicial para el vector de precisiones $\boldsymbol{\beta}$. Se comienza con la iteración $k=1$

2. Con los valores dados del vector $\boldsymbol{\beta}^k$ se encuentra en este paso el valor del vector de pesos $\boldsymbol{\alpha}^k$, mediante una optimización basada en el método de Newton-Raphson, para lo cuál se calcula el gradiente y la Hessiana de la ecuación (4.70) mediante

$$G=\nabla\log\left(P\left(\boldsymbol{Y}/\boldsymbol{X},\boldsymbol{\alpha}\right)p\left(\boldsymbol{\alpha}/\boldsymbol{\beta}\right)\right)=\boldsymbol{K}^T\left(\boldsymbol{y}-\boldsymbol{g}\right)-\boldsymbol{B}\boldsymbol{\alpha} \tag{4.71}$$

$$H=\nabla\nabla\log\left(P\left(\boldsymbol{Y}/\boldsymbol{X},\boldsymbol{\alpha}\right)p\left(\boldsymbol{\alpha}/\boldsymbol{\beta}\right)\right)=-\left(\boldsymbol{K}^T\boldsymbol{M}\boldsymbol{K}+\boldsymbol{B}\right) \tag{4.72}$$

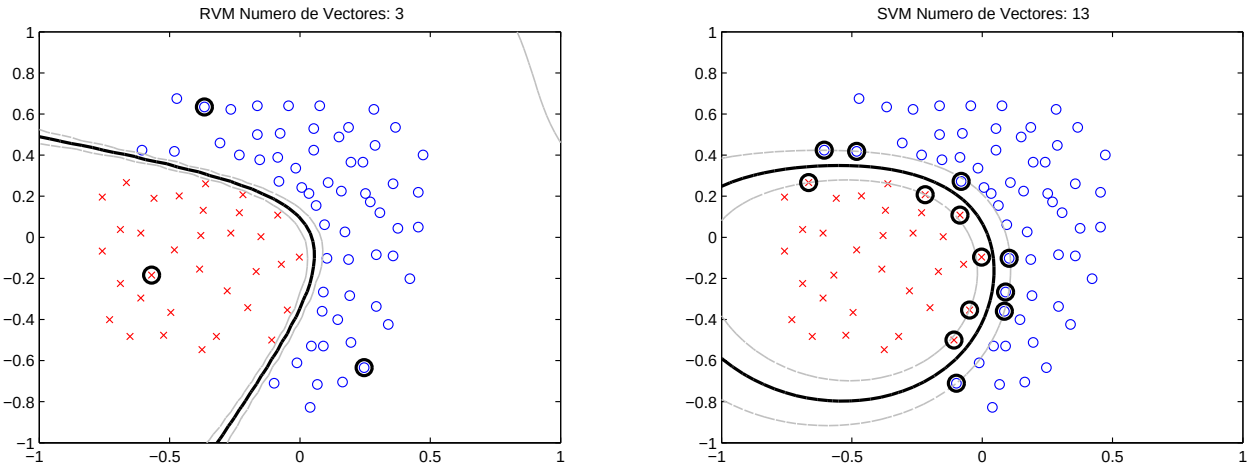
   donde $\boldsymbol{M}$ será una matriz diagonal cuyos elementos son $M_{ii}=g\left(\boldsymbol{x}_i\right)\left(1-g\left(\boldsymbol{x}_i\right)\right)$ y $\boldsymbol{K}$ es la matriz de Gram del conjunto de entrenamiento.

3. Una vez encontrado el nuevo vector $\boldsymbol{\alpha}^k$ y la Hessiana de la ecuación (4.70), se actualiza la precisión de los estimadores mediante

$$\beta_i^{k+1}=\frac{1-\beta_i\left(H\right)_i^{-1}i}{\left(\alpha_i^{k+1}\right)^2} \tag{4.73}$$

4. Si el nuevo vector de precisión $\boldsymbol{\beta}$ no ha variado por encima de un mínimo respecto a la iteración anterior o si se ha alcanzado el número máximo de iteraciones se detiene el algoritmo. De lo contrario se vuelve a repetir desde el paso 2.

Al realizar este método se puede observar cómo muchos de las precisiones β_i tienden a infinito, por lo que la varianza del estimador tiende a cero, indicando que el estimador de $\alpha_i\to 0$ y por tanto el vector asociado del conjunto de entrenamiento no es relevante



(a) **(b)**

Figura 4.6: Comparativa de métodos RVM y SVM para un problema de clasificación artificial.

para la clasificación. Al finalizar el proceso de aprendizaje, solo aquellos vectores del conjunto de entrenamiento que resulten con $\alpha_i \neq 0$ serán considerados en la ecuación (4.65).

En la figura (4.6) se pueden apreciar las fronteras de decisión creadas para un problema artificial de dos dimensiones, utilizando en el caso de la figura (4.6(a)) una máquina de vectores relevantes con un kernel gaussiano y en el caso de la figura (4.6(b)) una SVM con idéntico kernel y fijando su parámetro de regularización $C = 10$. Se puede ver cómo en el primer caso solo son necesarios tres vectores soporte por los trece necesarios obtenidos de las SVM. Al evaluar nuevas muestras el número de operaciones queda sensiblemente reducido en el caso de las RVM. Las líneas marcadas en gris corresponden a aquellas muestras con probabilidades de pertenencia a la clase superiores al 75 % en el caso de las RVM, mientras que en el caso de las SVM corresponden a un valor de $f(\boldsymbol{x}) \geq 1$.

La implementación del entrenamiento RVM en esta tesis está basado en el algoritmo descrito en esta sección. En [Tipping03] se describe los pasos para un algoritmo de entrenamiento secuencial que puede proporcionar velocidades de entrenamiento muy superiores al algoritmo descrito con conjuntos de datos de elevadas muestras.

Clasificador	Optimización Cross-Validation	Optimización Bootstrap .632	Optimización Bootstrap .632+
FLD	0.554	0.554	0.554
*k*NN	0.951	0.951	0.951
Perceptrón Multicapa	0.941	0.954	0.954
Fuzzy Artmap	0.931	0.929	0.935
RBF-NN	0.971	0.968	0.975
SIMCA	0.913	0.913	0.913
PLS-DA	0.592	0.592	0.592
SVM (Lineal)	0.761	0.761	0.761
SVM (Kernel RBF)	0.978	0.981	0.985
Random forest	0.965	0.965	0.965
LS-SVM (Lineal)	0.613	0.613	0.613
LS-SVM (Kernel RBF)	0.988	0.974	0.984
RVM (Kernel Lineal)	0.592	0.592	0.592
RVM (Kernel RBF)	0.963	0.961	0.971

Tabla 4.3: Comparativa de Resultados para el Conjunto de Alcoholes

4.5. Comparativa de clasificadores

4.5.1. Comparativa en exactitud y operaciones

En esta sección se analizará la comparación de los resultados obtenidos de la aplicación de los clasificadores descritos a los conjuntos de datos descritos en el capítulo anterior. Dichos resultados mostrarán la media de la tasa de acierto obtenida a partir de múltiples experimentos de división en conjuntos de entrenamiento y test. Estos resultados sobre las medias serán validados posteriormente con los test de significancia estadística. Adicionalmente, en esta sección también se muestra el número de operaciones necesarias para evaluar una nueva muestra con los diferentes métodos de clasificación.

En la tabla (4.3) se muestran los resultados de la clasificación de diversos alcoholes obtenidos con un único sensor de SnO_2 aplicando termomodulación. En primer lugar destaca el mal comportamiento de todos los métodos lineales, no proporcionando buenos resultados en ninguno de los casos, aunque las SVM con kernel lineal proporcionan un notable incremento de la tasa de acierto frente a sus competidores. La optimización de parámetros en estos clasificadores no afecta mucho, ya que en varios de ellos, como FLD y las RVM con kernel lineal, no existen parámetros a optimizar y en otros casos dichos parámetros no influyen significativamente en el resultado.

Clasificador	Comparaciones	Sumas	Multiplicaciones	No lineales
FLD	8.00	568.00	568.00	0.00
kNN	380.00	8.06×10^4	8.13×10^4	0.00
Perceptrón Multicapa	8.00	2.30×10^4	2.30×10^4	320.00
Fuzzy ArtMap	4.15×10^3	8.19×10^3	57.67	0.00
RBF-NN	8.00	5.68×10^5	5.71×10^5	1920
SIMCA	8.00	5.15×10^3	5.16×10^3	0.00
PLS-DA	8.00	568.00	568.00	0.00
SVM (Kernel Lineal)	8.00	568.00	568.00	0.00
SVM (Kernel RBF)	8.00	6.15×10^4	6.18×10^4	288.83
Random forest	3.09×10^5	0.00	0.00	0.00
LS-SVM (Kernel Lineal)	8.00	568.00	568.00	0.00
LS-SVM (Kernel RBF)	8.00	6.48×10^5	6.51×10^5	2352
RVM (Kernel Lineal)	8.00	473.33	473.33	0.00
RVM (Kernel RBF)	8.00	1.68×10^4	1.68×10^4	78.67

Tabla 4.4: Comparativa de Operaciones para el Conjunto de Alcoholes con parámetros optimizados

Respecto a los métodos no lineales, tanto SIMCA como las redes fuzzy artmap proporcionan resultados aceptables pero ciertamente alejados del resto de métodos no lineales. Si bien random forest y kNN no proporcionan los mejores resultados, sí se debe destacar que en los resultados obtenidos no influye en exceso la optimización de los parámetros. Random forest, tal como se indica en [Breiman01] no es un método que sufra de sobrentrenamiento y por tanto se obtienen buenos resultados con un número suficiente de árboles, como el seleccionado por defecto con un valor de 200.

Los mejores resultados obtenidos para este conjunto de datos se han encontrado con los métodos kernel descritos: SVM, LS-SVM y RVM. Es necesario destacar sin embargo, la importancia que tiene la optimización de parámetros, especialmente la anchura del kernel gaussiano para los resultados finales. Este hecho será tratado ampliamente en el siguiente capítulo. En cuanto a la comparación entre estos métodos kernel destacan ligeramente las SVM y las LS-SVM frente a las RVM.

Respecto al número de operaciones necesarias, se ha seleccionado aquellos clasificadores optimizados por cross-validation. La tabla (4.4) muestra la media de operaciones necesarias para evaluar cada muestra. Esta media se ha obtenido promediando el número de operaciones necesarias para evaluar una muestra dados los clasificadores de un experimento. La primera conclusión que se obtiene al observar la comparación de resultados es la gran diferencia que existe entre los métodos lineales y los no lineales.

Aunque para este dataset en concreto los resultados de precisión, o de forma equivalente de error obtenido, desaconsejan el uso de clasificadores lineales, se debe destacar que el número de operaciones necesarias es varios órdenes de magnitud menor que para el resto de clasificadores.

Las redes fuzzy artmap y SIMCA presentan un orden de operaciones notablemente inferior al resto de clasificadores no lineales. Random forest es un método que solo necesita comparaciones para poder determinar la clase de una muestra, aunque el número de éstas es bastante elevado. Sin embargo, la comparación más interesante se puede encontrar en los métodos kernel. Se puede observar cómo las LS-SVM, que proporcionaban ligeramente mejores resultados que las SVM o las RVM, necesitan un orden de operaciones muy elevado, solo comparable a las necesarias por las redes neuronales de base radial. Es necesario comentar que no se ha utilizado ningún algoritmo de poda, que podría hacer descender el número de operaciones necesarias, ya que en este caso pierden su ventaja en la comparación de exactitud media. Tal como se planteaba en la teoría, las RVM presentan un número de vectores relevantes inferior al número de vectores soporte en las SVM, aunque su entrenamiento requiere mucho más tiempo, del orden de horas, y no alcanzan las tasas de acierto de las SVM.

Los siguientes conjuntos de datos en ser examinados serán los procedentes de la clasificación de vinos procedentes de espectrofotometría UV-VIS de baja resolución. Al igual que en el capítulo 3 estos conjuntos de datos se han separado considerando los vinos blancos por un lado y los vinos tintos en el otro caso. Los resultados de exactitud se muestran en las tablas (4.5) y (4.6).

De los resultados de ambas tablas vemos que en este caso, para ambos conjuntos de datos, los clasificadores lineales tienen un mejor comportamiento que los no lineales. Estos resultados reflejan que se debe seleccionar el clasificador menos complejo que sea capaz de obtener buenos resultados (*Occam's razor*) [Duda00]. Destacan el mal comportamiento de los métodos SIMCA y PLS-DA, pese a que son métodos muy extendidos en el reconocimiento de señales espectroscópicas.

Si se atiende al número de operaciones promedio para evaluar una muestra, descrito en las tablas (4.7) y (4.8), no hay duda de que para estos conjuntos de datos la selección de clasificadores lineales es mucho más adecuada.

Los restantes conjuntos de datos se diferencian de los anteriores en que la información a clasificar proviene de diferentes sensores concurrentes. Para poder integrar la información de todos ellos se ha seguido una estrategia consistente en concatenar la información de los sensores y formar un único vector.

El primero de los conjuntos multisensoriales estudiados es el procedente del análisis por inyección en flujo (*FIA*) que fue explicado en el capítulo anterior. En la tabla (4.9) se pueden apreciar los resultados de exactitud con los diferentes clasificadores. En este caso se obtienen unas tasas de acierto altas para todos los clasificadores menos para las redes

Clasificador	Optimización Cross-Validation	Optimización Bootstrap .632	Optimización Bootstrap .632+
FLD	0.901	0.901	0.901
kNN	0.731	0.731	0.731
Perceptrón Multicapa	0.838	0.814	0.814
Fuzzy Artmap	0.825	0.824	0.831
RBF-NN	0.846	0.837	0.856
SIMCA	0.499	0.499	0.499
PLS-DA	0.582	0.593	0.593
SVM (Lineal)	0.889	0.889	0.889
SVM (Kernel RBF)	0.868	0.836	0.848
Random forest	0.845	0.825	0.825
LS-SVM (Lineal)	0.905	0.905	0.905
LS-SVM (Kernel RBF)	0.865	0.865	0.865
RVM (Kernel Lineal)	0.758	0.758	0.758
RVM (Kernel RBF)	0.796	0.796	0.796

Tabla 4.5: Comparativa de Resultados para el Conjunto de Blancos

fuzzy artmap y para el método SIMCA, para los que la tasa de acierto no es aceptable. El comportamiento de los clasificadores lineales es bastante bueno y requieren de un número de operaciones muy bajo, reflejado en la tabla (4.10). Por otro lado, los métodos no lineales incrementan la tasa de acierto hasta en un 5 % respecto a los métodos lineales aunque requieren una carga computacional muy superior comparados con estos últimos. Dependiendo de las necesidades de tasa de acierto y requerimientos de tiempo de test, se optará por la solución más adecuada.

Es importante destacar la influencia que tiene el ajuste de los parámetros en los métodos kernel, ya que cuando se ejecutan estos métodos sin optimización de parámetros los resultados pueden ser inferiores a los conseguidos con los métodos lineales.

Los resultados para el conjunto de datos procedente de la ciclovotiamperometría pueden verse en la tabla (4.11). En este caso, al igual que el anterior, los resultados para los métodos lineales quedan ligeramente por debajo de los obtenidos para los métodos kernel. El peor comportamiento lo encontramos en las redes fuzzy artmap, mientras que los mejores resultados los encontramos para las LS-SVM. Sin embargo, tal como se puede apreciar en la tabla (4.12), este último método requiere de un número de operaciones muy superior a sus competidores directos. Se puede apreciar cómo el método RVM proporciona unos resultados muy cercanos a los conseguidos con SVM

Clasificador	Optimización Cross-Validation	Optimización Bootstrap .632	Optimización Bootstrap .632+
FLD	0.882	0.882	0.882
*k*NN	0.772	0.772	0.772
Perceptrón Multicapa	0.822	0.835	0.832
Fuzzy Artmap	0.853	0.853	0.853
RBF-NN	0.758	0.758	0.758
SIMCA	0.624	0.631	0.617
PLS-DA	0.506	0.494	0.494
SVM (Lineal)	0.885	0.885	0.885
SVM (Kernel RBF)	0.841	0.847	0.813
Random forest	0.851	0.858	0.850
LS-SVM (Lineal)	0.888	0.888	0.888
LS-SVM (Kernel RBF)	0.844	0.844	0.844
RVM (Kernel Lineal)	0.815	0.815	0.815
RVM (Kernel RBF)	0.778	0.808	0.818

Tabla 4.6: Comparativa de Resultados para el Conjunto de Tintos

o con LS-SVM con un número de operaciones en un orden de magnitud inferior a las requeridas por estos métodos.

El último conjunto de datos analizado es el procedente del CSIC, cuyos resultados se muestran en la tabla (4.13). Se puede apreciar cómo no se han conseguido elevadas tasas de acierto con ninguno de los métodos, debido a la naturaleza de los datos. Sin embargo, hay que destacar el pésimo comportamiento de alguno de los métodos, como es el caso de SIMCA y de las redes fuzzy artmap. Los métodos lineales quedan lejos de los métodos kernel, aunque en este caso existen diferencias claras entre ellos. Así, los resultados obtenidos mediante las SVM o los métodos propuestos en este capítulo, LS-SVM y RVM, obtienen una clara ventaja respecto al resto de métodos. También se destaca el peor comportamiento de random forest con este conjunto de datos respecto a los métodos anteriores cuando utilizan un kernel gaussiano.

El número de operaciones se presenta en la tabla (4.14) y se puede observar cómo en este caso las RVM no distan tanto del número de operaciones máximo, que como en los casos anteriores, se produce para las LS-SVM. Este comportamiento es lógico, ya que al ser un problema de clasificación difícil el número de vectores relevantes en el conjunto de entrenamiento aumenta de forma notable.

En la siguiente sección se hará un análisis estadístico de los resultados obtenidos para finalmente enunciar las conclusiones sobre los métodos de clasificación empleados.

Clasificador	Compara-ciones	Sumas	Multiplica--ciones	No lineales
FLD	6.00	366.00	366.00	0.00
kNN	28.29	5.15×10^3	5.20×10^3	0.00
Perceptrón Multicapa	6.00	1.49×10^4	1.49×10^4	240.00
Fuzzy ArtMap	1.47×10^3	2.89×10^3	23.71	0.00
RBF-NN	6.00	7.32×10^3	7.36×10^3	40.00
SIMCA	6.00	2.92×10^3	2.93×10^3	0.00
PLS-DA	6.00	366.00	366.00	0.00
SVM (Kernel Lineal)	6.00	366.00	366.00	0.00
SVM (Kernel RBF)	6.00	2.07×10^4	2.08×10^4	113.29
Random forest	1.16×10^4	0.00	0.00	0.00
LS-SVM (Kernel Lineal)	6.00	366.00	366.00	0.00
LS-SVM (Kernel RBF)	6.00	3.11×10^4	3.12×10^4	169.71
RVM (Kernel Lineal)	6.00	366.00	366.00	0.00
RVM (Kernel RBF)	6.00	7.32×10^3	7.36×10^3	40.00

Tabla 4.7: Comparativa de Operaciones para el Conjunto de Blancos con parámetros optimizados

4.5.2. Test de significancia

En la sección anterior se mostró la media de la tasa de acierto de cada clasificador para los diferentes conjuntos empleados en la tesis. Sin embargo, tal como se ha propuesto en la metodología a utilizar, es necesario validar estos resultados de forma que las conclusiones de la sección anterior, basadas en las comparaciones entre medias de tasas de acierto, tengan significado estadístico. Para tal fin se han realizado los test de Wilcoxon y t-test por cada pareja de clasficadores. Sin embargo, para ganar en claridad en los resultados presentados, en esta sección solo se muestran los test entre clasificadores que mejor comportamiento han tenido en cada caso. El objetivo de esta parte es verificar que la diferencia entre medias tiene significado estadístico. La hipótesis nula, según la cuál no existe diferencia estadística entre dos clasificadores, no podrá ser rechazada si:

- En el caso del test Wilcoxon, cuando el valor del estadístico z, descrito en la ecuación (4.16), tenga un valor superior a -1.96. En las tablas se visualiza el valor de $-z$ para facilitar la comprensión de los resultados.
- En el caso del t-test, para nuestro caso, cuando la probabilidad de la hipótesis nula sea superior al 5 %. En los resultados, el valor de dicha probabilidad se mostrará en tanto por ciento para facilitar su visión.

Clasificador	Compara-ciones	Sumas	Multiplica--ciones	No lineales
FLD	8.00	488.00	488.00	0.00
kNN	66.00	1.20 x 10^4	1.21 x 10^4	0.00
Perceptrón Multicapa	8.00	1.98 x 10^4	1.98 x 10^4	320.00
Fuzzy ArtMap	2.45 x 10^3	4.82 x 10^3	39.50	0.00
RBF-NN	8.00	8.58 x 10^4	8.62 x 10^4	423.26
SIMCA	8.00	4.43 x 10^3	4.44 x 10^3	0.00
PLS-DA	8.00	488.00	488.00	0.00
SVM (Kernel Lineal)	8.00	488.00	488.00	0.00
SVM (Kernel RBF)	8.00	3.71 x 10^4	3.73 x 10^4	202.75
Random forest	2.24 x 10^4	0.00	0.00	0.00
LS-SVM (Kernel Lineal)	8.00	488.00	488.00	0.00
LS-SVM (Kernel RBF)	8.00	9.66 x 10^4	9.72 x 10^4	528.00
RVM (Kernel Lineal)	8.00	488.00	488.00	0.00
RVM (Kernel RBF)	8.00	1.12 x 10^4	1.12 x 10^4	61.13

Tabla 4.8: Comparativa de Operaciones para el Conjunto de Tintos con parámetros optimizados

En las tablas de resultados se han marcado en negrita las situaciones en las que la hipótesis nula no puede ser rechazada y por tanto las diferencias halladas en la media de la exactitud no son estadísticamente válidas.

Los resultados para el conjunto de medida de alcoholes se muestran en la tabla (4.15) donde el test de Wilcoxon valida casi todas las comparaciones entre medias, mientras que el t-test asegura que algunas de ellas no pueden ser consideradas. Sin embargo, ambos test validan estadísticamente que el mejor comportamiento lo presentan las LS-SVM seguidas de las SVM-RBF. El algoritmo de los k vecinos más cercanos se comporta peor estadísticamente que el resto de métodos reflejados en dicha tabla.

Respecto a los conjuntos de datos procedentes de espectrofotometría UV-VIS para discernir la denominación de origen de los vinos blancos y tintos, encontramos en las tablas (4.16) y (4.17) que no existen diferencias estadísticas entre los métodos lineales que mejor comportamiento han tenido, por lo que no se puede concluir cuál de ellos tiene un mejor comportamiento. Sin embargo, existe diferencia respecto a los otros métodos que mejor comportamiento presentan, en este caso las SVM con kernel no lineal y las fuzzy artmap.

Respecto al conjunto obtenido mediante técnica FIA para clasificar vinos, no existen diferencias entre las LS-SVM y las SVM ambas con kernel RBF, pero existen diferencias entre estos dos métodos y los demás.

Clasificador	Optimización Cross-Validation	Optimización Bootstrap .632	Optimización Bootstrap .632+
FLD	0.908	0.901	0.911
kNN	0.889	0.893	0.878
Perceptrón Multicapa	0.914	0.911	0.921
Fuzzy Artmap	0.565	0.498	0.568
RBF-NN	0.915	0.924	0.924
SIMCA	0.702	0.700	0.700
PLS-DA	0.891	0.885	0.893
SVM (Lineal)	0.913	0.914	0.914
SVM (Kernel RBF)	0.954	0.953	0.953
Random forest	0.931	0.934	0.934
LS-SVM (Lineal)	0.911	0.903	0.903
LS-SVM (Kernel RBF)	0.953	0.948	0.948
RVM (Kernel Lineal)	0.905	0.911	0.911
RVM (Kernel RBF)	0.922	0.924	0.924

Tabla 4.9: Comparativa de Resultados para el Conjunto de FIA

Clasificador	Compara-ciones	Sumas	Multiplica--ciones	No lineales
FLD	4.00	800.00	800.00	0.00
kNN	43.00	2.58×10^4	2.58×10^4	0.00
Perceptrón Multicapa	4.00	6.47×10^4	6.47×10^4	85.21
Fuzzy ArtMap	9.40×10^3	1.87×10^4	46.77	0.00
RBF-NN	3.69	7.02×10^3	7.03×10^3	11.69
SIMCA	4.00	6.82×10^3	6.82×10^3	0.00
PLS-DA	4.00	800.00	800.00	0.00
SVM (Kernel Lineal)	4.00	800.00	800.00	0.00
SVM (Kernel RBF)	4.00	4.76×10^4	4.77×10^4	79.38
Random forest	4.70×10^4	0.00	0.00	0.00
LS-SVM (Kernel Lineal)	4.00	800.00	800.00	0.00
LS-SVM (Kernel RBF)	4.00	1.03×10^5	1.03×10^5	172.00
RVM (Kernel Lineal)	4.00	800.00	800.00	0.00
RVM (Kernel RBF)	4.00	7.02×10^3	7.03×10^3	11.69

Tabla 4.10: Comparativa de Operaciones para el Conjunto de FIA con parámetros optimizados

Clasificador	Optimización Cross-Validation	Optimización Bootstrap .632	Optimización Bootstrap .632+
FLD	0.899	0.899	0.899
kNN	0.935	0.934	0.934
Perceptrón Multicapa	0.931	0.942	0.947
Fuzzy Artmap	0.736	0.683	0.683
RBF-NN	0.968	0.964	0.959
SIMCA	0.944	0.944	0.944
PLS-DA	0.865	0.865	0.865
SVM (Lineal)	0.917	0.917	0.917
SVM (Kernel RBF)	0.968	0.964	0.977
Random forest	0.967	0.967	0.967
LS-SVM (Lineal)	0.924	0.924	0.924
LS-SVM (Kernel RBF)	0.987	0.981	0.987
RVM (Kernel Lineal)	0.914	0.921	0.921
RVM (Kernel RBF)	0.963	0.961	0.965

Tabla 4.11: Comparativa de Resultados para el Conjunto de CV

Clasificador	Compara-ciones	Sumas	Multiplica--ciones	No lineales
FLD	4.00	5.10×10^3	5.10×10^3	0.00
kNN	65.00	2.49×10^5	2.49×10^5	0.00
Perceptrón Multicapa	4.00	8.36×10^4	8.36×10^4	41.70
Fuzzy ArtMap	6.05×10^4	1.21×10^5	47.40	0.00
RBF-NN	4.00	3.45×10^5	3.45×10^5	135.20
SIMCA	4.00	4.60×10^4	4.60×10^4	0.00
PLS-DA	4.00	5.10×10^3	5.10×10^3	0.00
SVM (Kernel Lineal)	4.00	5.10×10^3	5.10×10^3	0.00
SVM (Kernel RBF)	4.00	2.39×10^5	2.39×10^5	62.40
Random forest	9.99×10^4	0.00	0.00	0.00
LS-SVM (Kernel Lineal)	4.00	5.10×10^3	5.10×10^3	0.00
LS-SVM (Kernel RBF)	4.00	9.95×10^5	9.96×10^5	260.00
RVM (Kernel Lineal)	4.00	5.10×10^3	5.10×10^3	0.00
RVM (Kernel RBF)	4.00	4.48×10^4	4.48×10^4	11.70

Tabla 4.12: Comparativa de Operaciones para el Conjunto de CV con parámetros optimizados

Clasificador	Optimización Cross-Validation	Optimización Bootstrap .632	Optimización Bootstrap .632+
FLD	0.650	0.650	0.650
kNN	0.792	0.775	0.775
Perceptrón Multicapa	0.787	0.775	0.775
Fuzzy Artmap	0.503	0.477	0.477
RBF-NN	0.813	0.804	0.804
SIMCA	0.333	0.333	0.333
PLS-DA	0.650	0.683	0.683
SVM (Lineal)	0.712	0.712	0.712
SVM (Kernel RBF)	0.833	0.838	0.838
Random forest	0.767	0.775	0.785
LS-SVM (Lineal)	0.713	0.712	0.712
LS-SVM (Kernel RBF)	0.822	0.820	0.824
RVM (Kernel Lineal)	0.703	0.703	0.703
RVM (Kernel RBF)	0.798	0.785	0.793

Tabla 4.13: Comparativa de Resultados para el Conjunto de CSIC

Clasificador	Compara-ciones	Sumas	Multiplica--ciones	No lineales
FLD	3.00	360.00	360.00	0.00
kNN	29.00	1.04×10^4	1.05×10^4	0.00
Perceptrón Multicapa	3.00	1.26×10^4	1.26×10^4	32.44
Fuzzy ArtMap	3.44×10^3	6.82×10^3	28.40	0.00
RBF-NN	3.00	2.71×10^4	2.71×10^4	66.42
SIMCA	3.00	3.25×10^3	3.26×10^3	0.00
PLS-DA	3.00	360.00	360.00	0.00
SVM (Kernel Lineal)	3.00	360.00	360.00	0.00
SVM (Kernel RBF)	3.00	2.11×10^4	2.11×10^4	58.50
Random forest	1.43×10^4	0.00	0.00	0.00
LS-SVM (Kernel Lineal)	3.00	360.00	360.00	0.00
LS-SVM (Kernel RBF)	3.00	3.13×10^4	3.14×10^4	87.00
RVM (Kernel Lineal)	3.00	360.00	360.00	0.00
RVM (Kernel RBF)	3.00	1.04×10^4	1.04×10^4	24.51

Tabla 4.14: Comparativa de Operaciones para el Conjunto de CSIC con parámetros optimizados

Wilcox.	Método	SVM (RBF)	RBF-NN	RF	RVM	kNN
	LS-SVM (RBF)	24.13	19.98	26.21	28.37	26.87
	SVM (RBF)		6.50	13.86	26.58	20.83
	RBF-NN			**0.66**	15.36	10.27
	RF				3.30	15.18
	RVM					7.35
TTest	Método	SVM (RBF)	RBF-NN	RF	RVM	kNN
	LS-SVM (RBF)	0.00	0.30	0.00	0.00	0.00
	SVM (RBF)		14.59	0.33	0.00	0.00
	RBF-NN			**21.07**	**8.31**	1.04
	RF				**61.08**	3.66
	RVM					2.12

Tabla 4.15: Test de Wilcoxon y T-test para el conjunto Alcoholes

Wilcox.	Método	FLD	SVM (Lineal)	SVM (RBF)
	LS-SVM (Lineal)	**1.58**	**1.89**	14.99
	FLD		**0.75**	20.36
	SVM (Lineal)			13.20
TTest	Método	FLD	SVM (Lineal)	SVM (RBF)
	LS-SVM (Lineal)	**88.39**	**71.04**	0.36
	FLD		**82.80**	0.03
	SVM (Lineal)			0.21

Tabla 4.16: Test de Wilcoxon y T-test para el conjunto Blancos

Wilcox.	Método	SVM (Lineal)	FLD	F. Artmap	RF
	LS-SVM (Lineal)	4.43	7.35	8.96	11.31
	SVM (Lineal)		3.11	15.84	17.91
	FLD			12.35	18.19
	F. Artmap				9.52
TTest	Método	SVM (Lineal)	FLD	F. Artmap	RF
	LS-SVM (Lineal)	**79.61**	**78.12**	2.05	0.00
	SVM (Lineal)		**82.14**	0.03	1.04
	FLD			3.71	2.49
	F. Artmap				**89.52**

Tabla 4.17: Test de Wilcoxon y T-test para el conjunto Tintos

Wilcox.	Método	LS-SVM (RBF)	RF	RVM (RBF)
	SVM (RBF)	**1.13**	14.61	18.95
	LS-SVM (RBF)		17.16	18.10
	RF			15.93
TTest	Método	LS-SVM (RBF)	RF	RVM (RBF)
	SVM (RBF)	**94.13**	2.57	0.45
	LS-SVM (RBF)		0.29	0.00
	RF			2.14

Tabla 4.18: Test de Wilcoxon y T-test para el conjunto FIA

Wilcox.	Método	SVM (RBF)	RBF-NN	RF	RVM (RBF)
	LS-SVM (RBF)	29.50	31.20	30.35	31.39
	SVM (RBF)		21.30	23.19	29.98
	RBF-NN			23.75	25.17
	RF				**0.38**
TTest	Método	SVM (RBF)	RBF-NN	RF	RVM (RBF)
	LS-SVM (RBF)	0.00	0.00	0.00	0.00
	SVM (RBF)		0.25	0.00	0.00
	RBF-NN			0.00	0.00
	RF				**73.28**

Tabla 4.19: Test de Wilcoxon y T-test para el conjunto CV

Por último, para el conjunto de ciclovoltiamperogramas, se validan todas las diferencias obtenidas entre las medias a excepción de la existente entre random forest y las RVM.

En resumen, podemos apreciar cómo muchas de las conclusiones sobre qué método se comporta mejor han sido validadas mediante los test de significancia, necesarios por otra parte para aseverar la superioridad de un método sobre otros.

4.5.3. Conclusiones sobre los métodos de clasificación

Los diferentes métodos de clasificación empleados en la revisión bibliográfica han sido probados con diferentes conjuntos de datos, validando los resultados de forma estadística y analizando el número de operaciones necesarias. A partir de estos resultados se destacan una serie de conclusiones importantes:

- Existen conjuntos de datos donde los únicos métodos aceptables son algunos de los no lineales. Entre los métodos que mejores resultados han demostrado en este tipo de conjuntos de datos son los métodos kernel (SVM, RVM y LS-SVM), las

redes neuronales de base radial y random forest. Los resultados obtenidos por los algoritmos kNN y los perceptrones multicapa son ligeramente inferiores, mientras que los obtenidos por los métodos lineales no son aceptables para ciertos conjuntos de datos, como se ha demostrado para la clasificación de alcoholes basada en sensores termomodulados de SnO_2 y el conjunto de datos de sensores de gases tóxicos procedente del CSIC.

- La anterior conclusión no se puede generalizar para todos los conjuntos de datos, bien al contrario, existen conjuntos de datos en los que los clasificadores lineales demuestran su superioridad frente a los métodos no lineales, incluidos los citados como mejores clasificadores para problemas difíciles. Puesto que el número de operaciones requerido para analizar una muestra con un clasificador lineal está en dos o tres órdenes de magnitud por debajo del requerido por los clasificadores no lineales, la conclusión clara de este apartado es que son óptimos en alguno de los problemas expuestos como es la clasificación de vinos por espectrofotometría UV-VIS.

- Existe un tercer grupo de experimentos para los que se ha obtenido un resultado aceptable para los métodos lineales pero ligeramente inferior a los obtenidos con métodos no lineales. Sin embargo, atendiendo al número de operaciones necesarias, los métodos lineales superan a cualquier otro tipo de clasificador. En estos conjuntos de datos, se debería establecer qué tasa de error resulta admisible y los requisitos en número de operaciones.

- Algunos de los métodos más extendidos, sobre todo en el campo de la lengua electrónica, como son SIMCA y PLS-DA proporcionan resultados regulares para algunos de los conjuntos empleados y no aceptables para otros. Esta conclusión es especialmente relevante si se tiene en cuenta que en gran parte de los programas comerciales para el análisis de sistemas de nariz y lengua electrónica son los únicos métodos de clasificación considerados.

- En los métodos kernel considerados (SVM, RVM, LS-SVM y las redes neuronales de base radial), la optimización de parámetros juega un papel fundamental en la tasa de acierto final. Si esta optimización no se lleva a cabo, los resultados pueden ser peores que otros métodos de clasificación.

- Para los métodos kernel considerados, por norma general los mejores resultados se han obtenido con las LS-SVM, aunque en alguno de los casos no tiene una relevancia estadística dicha mejora. Por contra, el número de operaciones necesarias es muy superior a las necesarias para determinar una nueva muestra frente a las SVM y especialmente frente a las RVM. Este hecho es debido a que no se utiliza

ningún algoritmo de poda una vez entrenadas, si bien es cierto que se realizaron algunas pruebas con algoritmos de poda y las LS-SVM perdían su ventaja competitiva en tasa de acierto. Por otro lado, las RVM proporcionan tasas de acierto más bajas pero con un número de operaciones kernel significativamente reducido.

- Random forest es un método que funciona bastante bien, tanto para conjuntos de datos separables no linealmente como aquellos en los que prevalece el uso de clasificadores lineales. Las operaciones requeridas son todas comparaciones, por la propia naturaleza del algoritmo. Sin embargo, aunque este tipo de operaciones se viera reducido quedaría muy lejos del número de operaciones necesarias para un clasificador lineal. Por otro lado, en aquellas aplicaciones donde se requieren métodos no lineales y tasas de acierto altas, son ligeramente superiores los métodos kernel.

- El método propuesto de entrenamiento incremental de las LS-SVM no influye en los resultados obtenidos, ya que los clasificadores obtenidos son los mismos y el número de operaciones para la fase de test no varía. Sin embargo, es un método adecuado para no tener que volver a entrenar sobre todo el conjunto de datos. También proporciona una herramienta útil para la selección de los patrones que deben formar parte del conjunto de entrenamiento.

4.6. Selección de características

Como se describe en el capítulo 2, para definir formalmente la selección de características, supongamos que tenemos un conjunto $\boldsymbol{X}$ de l muestras, teniendo cada una de ellas inicialmente m características. El problema de la selección de características consiste en encontrar un subconjunto de características óptimas, de forma que los nuevos vectores tendrán d características con $d < m$. La selección de características se realiza mediante una función de criterio o función objetivo $J(\mathcal{D})$, donde $\mathcal{D}$ es una de las posibles combinaciones con d elementos seleccionados de las m características totales. Entre los métodos expuestos en la revisión bibliográfica destacan los siguientes:

- Sequential Fordward Floating Search. (SFFS)
- Sequential Backward Floating Search. (SBFS)
- Algoritmos genéticos
- Simulated annealing

Aunque en la revisión bibliográfica se expusieron más métodos, se indicó cómo los métodos enumerados tendían a solucionar los problemas de los anteriores, por lo que

en este apartado solo consideraremos estos métodos como *métodos clásicos* de selección de características. Este tipo de algoritmos proceden del campo del reconocimiento de patrones y no tienen en cuenta las posibles correlaciones existentes entre las muestras adyacentes. En el capítulo 3, se expusieron diversas medidas de extracción de parámetros de las señales dinámicas, de forma que puedan construirse clasificadores más sencillos. Cualquiera de los algoritmos expuestos anteriormente puede ser aplicado sobre este tipo de patrones procedentes de la extracción de características dinámicas, de forma que se determinen cuáles de dichos parámetros resultan relevantes para el problema de clasificación bajo estudio.

Cuando se trabaja con señales de tipo RAW, además de realizar clasificadores más sencillos, la selección de características puede tener una utilidad de extrema importancia, consistente en determinar dónde se encuentra la información útil, con el objetivo de mejorar la tecnología de la parte de los sensores. Así, dependiendo del tipo de señal que estemos considerando, podemos analizar:

Señales espectroscópicas En este caso estaremos interesados en determinar qué longitudes de onda consiguen una mayor discriminación. La instrumentación con un alto grado de precisión en todo el espectro es muy costosa, por lo que debemos concentrarnos en aquellas zonas del espectro realmente útiles.

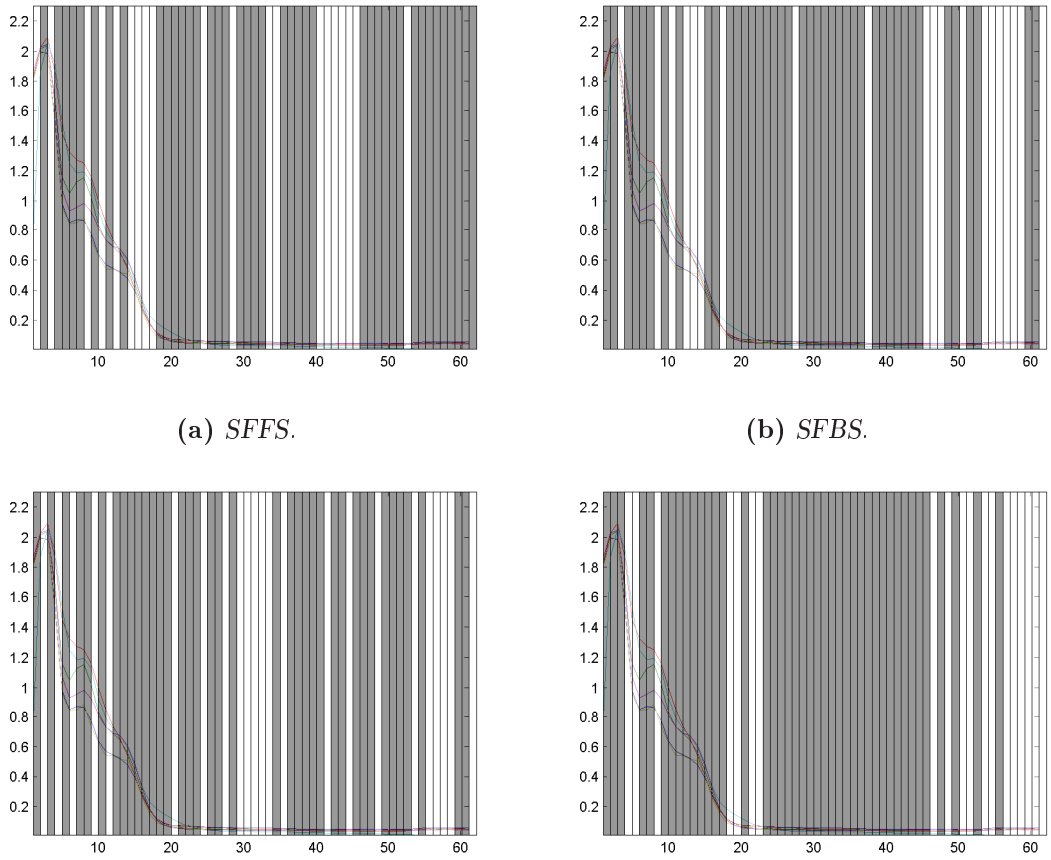
Señales termomoduladas Para este tipo de señales se buscan las temperaturas de trabajo que permiten una mejor clasificación del problema. El objetivo en este caso es doble: por un lado, se puede ajustar el sensor para que únicamente trabaje a las temperaturas de interés, no sometiendo al mismo a temperaturas más altas si no es preciso. Por otro lado, un barrido completo en temperatura está relacionado con una señal que ataca al heater que durará más tiempo cuanto mayor sea el rango de temperaturas a estudiar. El hecho de encontrar las temperaturas relevantes se traduce en disminuir el periodo de las señales de termomodulación.

Ciclovoltiamperogramas En este caso estamos interesados en conocer qué tensiones debemos aplicar a los sensores. Además de recortar el tiempo de barrido, conseguiremos no someter a los sensores a tensiones extremas si no es necesario, alargando la vida de los mismos.

Sistemas FIA El interés en este tipo de sistemas se centra en el tiempo de inyección necesario y en el periodo de recuperación hasta poder inyectar un nuevo bolo de analito. El ajuste de estos tiempos hace que la selección de características sea de especial relevancia.

Sin embargo, los métodos propuestos en la revisión bibliográfica, al tratar los patrones como vectores formados por elementos independientes, realizan su búsqueda

teniendo en cuenta la presencia o ausencia de cada característica individual. Adicionalmente, el número de combinaciones cuando se consideran las señales en formato RAW es tan sumamente elevado que las soluciones devueltas, aún mejorando el problema desde un punto de vista estricto de reconocimiento de patrones, aportan poco a la comprensión de las señales en el sentido arriba descrito. En la figura (4.7) se representa la solución de las características devuelta por cada uno de los métodos anteriormente mencionados para el problema de clasificación de vinos blancos, dibujando adicionalmente varias muestras representativas de las señalaes de cada clase y franjas oscuras para aquellas longitudes de onda que resultan descartadas en el proceso de selección de características. En las figuras (4.7(a)) y (4.7(b)) la búsqueda se ha realizado con los métodos SFFS y SBFS respectivamente, fijando el número de componentes a un valor de 20. En las figuras (4.7(c)) y (4.7(d)) se han utilizado algoritmos genéticos y simulated annealing, obteniendo el mejor resultado tras haber sido evaluadas un mínimo de 1000 combinaciones de forma guiada según el algoritmo utilizado.



(a) *SFFS.*

(b) *SFBS.*

(c) *Algoritmos genéticos.*

(d) *Simulated annealing.*

Figura 4.7: Selección de características para espectrofotogramas de vinos blancos.

Se puede apreciar cómo la información proporcionada por estos algoritmos no es concluyente respecto dónde se encuentra la información útil, incluso puede inducir a considerar información errónea. Así, el algoritmo SFBS tiende a premiar a las longitudes de onda más altas, mientras que si la búsqueda se realiza mediante el algoritmo SFFS tiende a premiar a las longitudes de onda más bajas. Hay que destacar que, aunque en la figura las longitudes de onda más alta parecen ser nulas, existen ligeras diferencias cuando se normalizan los datos respecto del máximo de cada característica. En los métodos aleatorios, además de no mostrar claramente la información sobre qué bandas de longitudes onda deben ser seleccionadas, surge el problema de desconocer el número de evaluaciones necesarias para detener el algoritmo.

4.6.1. Método propuesto

El objetivo del método de selección de características propuesto es obtener la información que mejor explique el problema, con un mínimo número de características y conseguir que dicha información pueda ser utilizada para determinar las zonas de información relevante del problema de clasificación.

En lugar de considerar cada característica como un elemento aislado del patrón, el método considera bloques de información que son descartados a medida que se va comprobando que no son útiles o se van refinando a medida que transcurre el algoritmo. Los conceptos de bloques de características y vecindad de los mismos son la parte más importante del algoritmo, coincidiendo con el objetivo de seleccionar zonas relevantes. El algoritmo consta de los siguientes pasos:

1. Se inicializan las variables $J_{best} \leftarrow 0$, $NC_{best} = \infty$, $\boldsymbol{B}_{best} \leftarrow \varnothing$, $n = 1$.

2. Se parte de una división de las señales en q bloques, de forma que se tendrá un conjunto inicial de bloques $\boldsymbol{B}_1 = \{b(1), b(2), ..., b(q)\}$. Cada uno de los bloques contendrá un conjunto de características asociados. En este paso se marcan todos los bloques con estado *activo*.

3. Para los bloques que no han sido descartados, se encuentra la mejor combinación C_n de bloques presentes/ausentes que maximice la función criterio $J(C_n)$. Dicha mejor combinación tendrá asociado un número de características no descartadas NC.

4. Si se da una de las siguientes condiciones:

   - $J_{best} < J(C_n)$
   - $J_{best} = J(C_n)$ y $NC < NC_{best}$

   Se almacena el resultado como mejor resultado hasta ahora:

- $J_{best} \leftarrow J(C_n)$
- $C_{best} \leftarrow \boldsymbol{C}_n$
- $NC_{best} \leftarrow NC$

5. Se comprueba si el bloque con más características del conjunto $\boldsymbol{B}_n$ ha alcanzado el mínimo deseado. Si dicho punto ha sido alcanzado, se devuelve la combinación de características C_{best}. Si no se ha alcanzado dicho punto el algoritmo prosigue.

6. Se genera un nuevo conjunto $\boldsymbol{B}_{n+1}$ a partir de la división de los bloques del conjunto $\boldsymbol{B}_n$. Los bloques b_n que ya tuvieran tamaño unidad no quedan divididos. Cada bloque $b_n(h)$ queda dividido en dos bloques $b^i_{n+1}(h)$ y $b^d_{n+1}(h)$, siendo el estado de estos nuevos bloques el calculado mediante las siguientes reglas:

   a) Si el bloque $b_n(h)$ tenía el estado de *eliminado*, sus descendientes $b^i_{n+1}(h)$ y $b^d_{n+1}(h)$ quedan marcados como eliminados.

   b) Si el bloque $b_n(h)$ tenía el estado de *activo*, sus descendientes $b^i_{n+1}(h)$ y $b^d_{n+1}(h)$ quedan marcados como activos si el bloque $b_n(h)$ estaba presente en la combinación C_n. En caso de que no estuviera presente, los descendientes tendrán el estado de *suspendidos*.

   c) Si el bloque $b_n(h)$ tenía el estado de *suspendido* sus descendientes $b^i_{n+1}(h)$ y $b^d_{n+1}(h)$ quedan marcados como activos si el bloque $b_n(h)$ estaba presente en la combinación C_n. En caso de que no estuviera presente:

      - El bloque $b^i_{n+1}(h)$ se marcará como suspendido si el bloque $b^d_{n+1}(h-1)$ está marcado como activo. En caso contrario o en el caso $h = 1$ el bloque $b^i_{n+1}(h)$ se marca como eliminado.
      - El bloque $b^d_{n+1}(h)$ se marca como suspendido si el bloque $b_n(h+1)$ está presente en la combinación C_n. En caso contrario, o cuando el bloque $b_n(h)$ es el último del conjunto $\boldsymbol{B}_n$, el bloque $b^d_{n+1}(h)$ se marca como eliminado.

   Una vez generada la división, se considera $n = n + 1$ y se procede con el nuevo conjunto de bloques creados. Se repite el algoritmo desde el paso 2.

En el paso 2 se busca la combinación de los bloques no eliminados que proporcione un mejor resultado según la función criterio $J(C_n)$. En nuestro caso, dicha función criterio medirá la media de acierto de un conjunto de datos dado al que se le aplica el procedimiento de validación cruzada con cinco divisiones. Además de la mejor tasa de acierto, en igualdad de condiciones, la función $J(C_n)$ selecciona aquella combinación que tenga asociadas un menor número de características seleccionadas. Cuando

el número de bloques considerado es inferior a nueve, se prueban todas las posibles combinaciones con excepción de la combinación nula por carecer de sentido. Si el número de bloques es superior, se ejecuta un algoritmo genético para estimar la mejor combinación. Dicho algoritmo genético tendrá los siguientes parámetros:

- Número de individuos por cada generación = 2 x Número de bloques considerados.
- Número máximo de generaciones = 20.
- Algoritmo de selección para el cruce mediante ruleta, donde cada individuo tendrá una probabilidad de selección proporcional al ranking que haya ocupado su solución en la generación evaluada.
- Probabilidad de mutación $P = 0{,}01$.
- Reemplazo con estrategia elitista, donde la siguiente población también contendrá las cinco mejores soluciones de la generación anterior.
- Para la formación de la población inicial, se consideran las siguientes combinaciones para formar la mitad de la población:
  - Todos los bloques no eliminados.
  - Solo los bloques con estado activo.
  - Todos los bloques con estado activo y los que tienen estado suspendido se incluirán o no mediante una función aleatoria.

  El resto de bloques, hasta completar el número de individuos de la generación se forman aleatoriamente, no permitiendo la combinación nula por carecer de sentido. En cualquier caso, un agente supervisa que en la primera generación no existan dos individuos iguales.

En la figura (4.8) se expone un ejemplo del método propuesto aplicado sobre el conjunto de separación de vinos blancos por espectrofotometría UV-VIS. Al igual que se hizo en la figura (4.7) se han representado una serie de muestras características de cada clase para mayor claridad del ejemplo. En la primera fase se divide el total de las 61 longitudes de onda que conforman la señal en 4 bloques. Se comprueban las 15 posibles combinaciones y se determina que la que mejor resultado proporciona es considerar únicamente el primer bloque (figura 4.8(a)). Dado que es la primera evaluación del algoritmo se guarda como ganadora la división de bloques actual. A continuación se dividen los bloques, marcando como activos los descendientes del primer bloque y como suspendidos los descendientes de los otros bloques, pues los bloques a dividir no

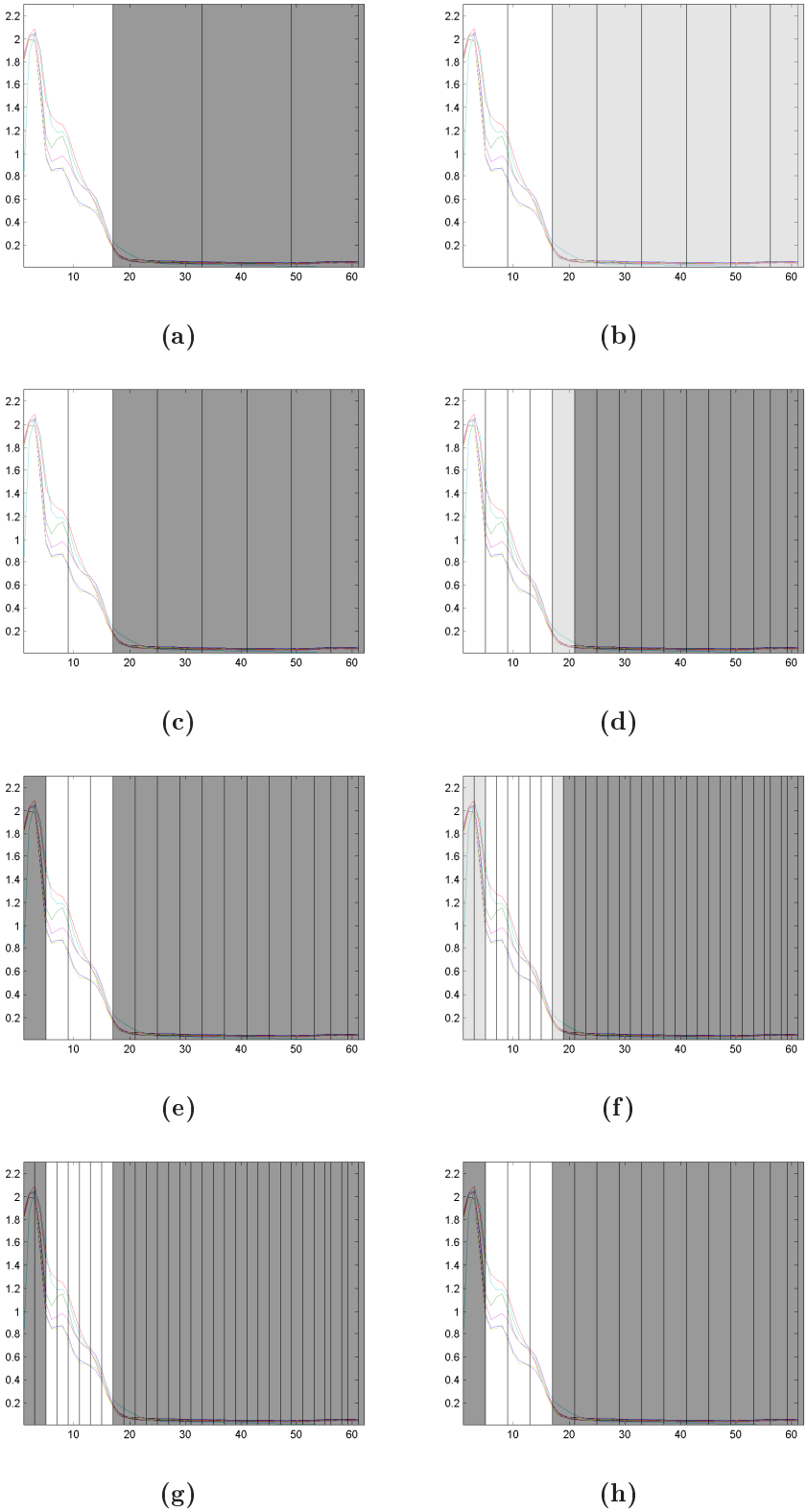


Figura 4.8: Ejemplo de aplicación del algoritmo de selección de características propuesto.

se encuentran en la combinación ganadora y estaban marcados como activos (figura 4.8(b)). Puesto que no hay bloques eliminados en la actualidad, probamos las siguientes 255 posibles combinaciones correspondientes a 8 bloques no eliminados. Como resultado de esta búsqueda encontramos presentes en la combinación ganadora solamente los dos primeros bloques (figura 4.8(c)). Esta combinación no ha mejorado ni la tasa de acierto ni el número de características anterior, por lo que no queda registrada como la mejor. Se realiza una nueva división de bloques, quedando como activos los descendientes de los primeros bloques y como eliminados los descendientes de los demás a excepción del bloque adyacente a los que han resultado activos, que mantiene su condición de suspendido, marcada con un tono de gris más claro (figura 4.8(d). Hay que considerar que solo tenemos cinco bloques activos en este momento, por lo que se siguen probando todas las combinaciones, resultando la combinación reflejada en la figura (4.8(e)) que mejora el resultado obtenido hasta ahora y además presenta menos características, por lo que se almacena como combinación ganadora hasta el momento. De nuevo se vuelven a dividir los bloques, pasando a estar suspendidos los dos primeros, ya que el bloque del cuál descienden estaba marcado como activo, pero no presente en la combinación ganadora. Los bloques descendientes de aquellos que estaban activos y que han resultado presentes en la combinación ganadora se marcan como activos. Sobre los descendientes del último bloque, que estaba marcado como suspendido por ser adyacente a un bloque activo, continua como suspendido el nuevo bloque adjunto a los que se han determinado como activos, mientras que el otro descendiente es marcado como eliminado (figura 4.8(f)). Dado que el número de bloques no eliminados ahora es de nueve, se utiliza un algoritmo genético para determinar la combinación ganadora, resultando ser la que se muestra en la figura (4.8(g)). Como esta combinación ganadora no ha mejorado a la anterior almacenada y se ha llegado al final del algoritmo se devuelve la combinación ganadora reflejada en la figura (4.8(h)). Se puede observar cómo el resultado, además de conseguir mejores tasas de acierto, aporta una información sobre las zonas a seleccionar mucho más rica que la mostrada en la figura (4.7).

En los casos en los que las señales proceden de diversos sensores, las divisiones se hacen de forma aislada a cada señal, no considerando la vecindad de los bloques iniciales de una señal con otra.

4.6.2. Resultados

En esta sección se muestran los resultados obtenidos en comparación con los algoritmos clásicos de selección de características, mostrando el número de características resultantes, el número de evaluaciones necesarias para llegar a la solución final, la tasa de acierto de la solución final y el parámetro φ que mide el nivel de vecindad de la solución propuesta. Este parámetro se ha definido mediante:

$$\varphi = \frac{\sum_{i=1}^{d-1} \delta(i, i+1)}{d-1} \tag{4.74}$$

donde d es el número de características seleccionadas y la función $\delta(i, i+1)$ será uno si las muestras $i, i+1$ de la solución final son contiguas y cero en caso contrario. Mediante este parámetro podemos identificar si el algoritmo de selección de características está dando zonas de información o por el contrario selecciona características de forma independiente.

Como método de clasificación se ha empleado en todos los casos SVM con kernel lineal o gaussiano, dependiendo del que consiguiera una mayor tasa de éxito en cada conjunto en la comparación de clasificadores. En los casos en los que se ha empleado un kernel gaussiano, se han utilizado los parámetros que fueron ajustados en la comparación de clasificación. Esto se realiza para poder utilizar el conjunto de entrenamiento como medida de la función criterio que guía los algoritmos. Finalmente, los resultados de tasa de acierto se realizan con un conjunto aislado de test.

El algoritmo propuesto en la sección anterior, que constituye una aportación de esta tesis, se ha denominado Multi-Block Feature Selection (MBFS) y en las siguientes tablas se compara con los algoritmos SFFS, SFBS, algoritmos genéticos y simulated annealing. En los dos primeros casos, se fija el número de características a obtener, por lo que se han realizado pruebas con el mismo número de características que obtenemos en nuestro algoritmo y con un 80 % de ese número (SFFS0.8 y SFBS0.8). Por otro lado, en el caso de los algoritmos genéticos y simulated annealing se fija el número de funciones a evaluar. En el caso de los algoritmos genéticos se han realizado pruebas con un número de evaluaciones correspondiente a poblaciones con el doble de individuos por generación que el número de características a considerar y un número de generaciones correspondientes a 20 y 30 generaciones (GA20G, SA30G). En el caso de simulated annealing, el algoritmo utilizado es el descrito en [Llobet07], donde para cada temperatura se realiza un número de evaluaciones igual a la dimensión del problema. Para tener las mismas evaluaciones que en el caso de los algoritmos genéticos se opta por la evaluación del doble de temperaturas, denominando como tal los algoritmos SA40G y SA60G.

En la tabla (4.20) se muestran los resultados para el conjunto de vinos blancos que ha sido expuesto en las figuras (4.7) y (4.8) como ejemplo en las secciones anteriores. Como se ha indicado, los resultados se obtienen al realizar el test con un conjunto de datos aislado que no ha sido utilizado para guiar ninguno de los algoritmos de selección de características. Los resultados muestran cómo el algoritmo propuesto obtiene una tasa de acierto tan buena como los algoritmos genéticos con menos evaluaciones y una agrupación de características muy superior a éstos. En [Baxter97] se señala cómo la información más relevante sobre el cultivo de vinos blancos está contenida en la banda

Algoritmo	Número de características	Número de evaluaciones	Accuracy	φ
MBFS	12	661	0.897	1
SFFS	12	873	0.867	0.545
SFBS	12	3548	0.852	0
SFFS0.8	10	749	0.867	0.33
SFBS0.8	10	3897	0.847	0
GA20G	20	2440	0.897	0.157
SA40G	14	2440	0.863	0.384
GA30G	25	3660	0.897	0.157
SA60G	13	3660	0.863	0.636

Tabla 4.20: Comparativa de métodos de selección para el conjunto de datos de separación de vinos blancos por espectrofotometría UV-VIS.

de 240-400 nm. por la presencia en esta banda de ésteres de ácidos hidroxicinámicos. La banda relevante encontrada para el algoritmo propuesto es de 260-380 nm. por lo que se puede comprobar la utilidad del algoritmo para encontrar bandas de información conjuntas.

Algoritmo	Número de características	Número de evaluaciones	Accuracy	φ
MBFS	22	533	0.917	0.854
SFFS	22	2230	0.901	0.571
SFBS	22	3012	0.881	0.472
SFFS0.8	18	1819	0.866	0.471
SFBS0.8	18	3260	0.821	0.352
GA20G	24	2440	0.921	0.304
SA40G	50	2440	0.921	0.76
GA30G	24	3660	0.921	0.304
SA60G	34	3660	0.923	0.76

Tabla 4.21: Comparativa de métodos de selección para el conjunto de datos de separación de vinos tintos por espectrofotometría UV-VIS.

En la tabla (4.21) se muestran los resultados para el conjunto de vinos tintos por espectrofotometría UV-VIS. En este caso, la tasa de acierto alcanzada por el algoritmo propuesto es ligeramente inferior a los algoritmos genéticos y simulated annealing es la misma, pero a diferencia de estos últimos, el número de características en el algoritmo propuesto es menor y la tasa de agrupamiento de características es muy superior,

indicando que la información relevante, además de las banda UV señalada para los vinos blancos, debe estar en las bandas superiores a 700nm donde se sitúa el infrarrojo cercano. El hecho de encontrar esta información relevante en esta zona se debe a la detección de componentes fenólicos, que como se demuestra en [Gomez04] son dependientes del tipo de uva utilizada y la forma de cultivar la vid, lo que está estrechamente relacionado con la denominación de origen.

Algoritmo	Número de características	Número de evaluaciones	Accuracy	φ
MBFS	22	969	0.985	0.904
SFFS	22	3819	0.975	0.523
SFBS	22	6207	0.981	0.4286
SFFS0.8	18	2616	0.932	0.4716
SFBS0.8	18	7364	0.941	0.4118
GA20G	25	2840	0.985	0.4
SA40G	30	2840	0.985	0.517
GA30G	24	4260	0.985	0.478
SA60G	30	4260	0.985	0.6897

Tabla 4.22: Comparativa de métodos de selección para el conjunto de datos de separación de Alcoholes.

En las tablas (4.22), (4.23) y (4.24) se muestran los resultados tras aplicar el algoritmo a los conjuntos de alcoholes, ciclovoltiamperogramas y técnicas FIA respectivamente. En todas las tablas podemos apreciar cómo se verifica un comportamiento muy similar del algoritmo propuesto respecto a la tasa de acierto considerada pero un número mucho menor de evaluaciones y factor de características contiguas muy superior, lo que significa que efectivamente se están seleccionando bloques de características.

Respecto a la tabla (4.25) podemos ver cómo en todos los casos selecciona una gran cantidad de muestras respecto del total de 124 posibles. Esto hace que el mejor comportamiento se encuentre en el algoritmo SFBS en cuanto a número de evaluaciones. Podemos ver cómo el grado de características contiguas es muy elevado en todos los casos, no destacando especialmente el algoritmo propuesto. Este resultado es lógico, ya que si los algoritmos clásicos seleccionan un número de características próximo a la totalidad es lógico que muchas de ellas sean contiguas.

Algoritmo	Número de características	Número de evaluaciones	Accuracy	φ
MBFS	475	32150	0.973	0.611
SFFS	475	38940	0.966	0.267
SFBS	475	61410	0.957	0.14
SFFS0.8	380	29210	0.959	0.07
SFBS0.8	380	85420	0.949	0.06
GA20G	523	51200	0.975	0.221
SA40G	616	51200	0.979	0.253
GA30G	510	76800	0.977	0.202
SA60G	616	76800	0.979	0.187

Tabla 4.23: Comparativa de métodos de selección para el conjunto de datos de separación de vinos por ciclovoltiamperometría.

Algoritmo	Número de características	Número de evaluaciones	Accuracy	φ
MBFS	68	6250	0.954	0.623
SFFS	68	5463	0.951	0.313
SFBS	68	10214	0.953	0.287
SFFS0.8	54	4723	0.945	0.202
SFBS0.8	54	14877	0.948	0.242
GA20G	82	8160	0.951	0.0.278
SA20G	139	8160	0.954	0.421
GA30G	79	12240	0.955	0.225
SA20G	124	12240	0.954	0.353

Tabla 4.24: Comparativa de métodos de selección para el conjunto de datos de separación de vinos mediante técnica FIA.

4.7. Resumen de las aportaciones realizadas en el capítulo

La primera aportación que se ha realizado en este capítulo ha sido la de establecer una metodología de comparación entre los diferentes métodos de clasificación encontrados en la revisión bibliográfica y los propuestos en este capítulo. Esta metodología es absolutamente necesaria para buscar la optimización de los parámetros mejores de cada clasificador y realizar una validación estadística de los resultados. En la metodología se propone ajustar los parámetros utilizando estimadores del error mediante el conjunto de datos de entrenamiento y probar con un conjunto de test externo, repitiendo este

Algoritmo	Número de características	Número de evaluaciones	Accuracy	φ
MBFS	96	4603	0.795	0.863
SFFS	96	6809	0.705	0.757
SFBS	96	2250	0.831	0.779
SFFS0.8	77	5235	0.726	0.772
SFBS0.8	77	4506	0.782	0.828
GA20G	90	4960	0.816	0.756
SA20G	98	4960	0.788	0.815
GA30G	87	7440	0.816	0.783
SA20G	94	7440	0.788	0.806

Tabla 4.25: Comparativa de métodos de selección para el conjunto de datos de gases tóxicos procedente del CSIC.

proceso un número de veces que permita garantizar los resultados. Como resultado de la aplicación de dicha metodología se ha visto que no existe un método superior a los demás, sino que dependiendo del conjunto de datos es mejor utilizar métodos lineales en algunos casos y en otros métodos no lineales. En cualquier caso, los métodos kernel nos permiten esta facilidad, quedando muy bien situados en todos los casos.

Como segunda aportación se han introducido en la nariz y lengua electrónica dos métodos nuevos en el estado del arte del reconocimiento de patrones como son las máquinas de vectores soporte por mínimos cuadrados y las máquinas de vectores relevantes. Estos métodos ya han sido probados en otros campos y sus características iniciales aconsejan su aplicación a los sistemas de nariz y lengua electrónica. En el primer caso demuestran alcanzar tasas de acierto superiores en algunos casos al resto de clasificadores, pero el número de operaciones que precisan en la parte de test es elevado. En el segundo caso, no consiguen tan buenas tasas de acierto como las anteriores o las SVM, pero el número de operaciones necesarias para la fase de test es sensiblemente más reducido.

Se ha propuesto un algoritmo de aprendizaje incremental para las LS-SVM, basado en las descomposiciones Cholesky y las propiedades de dicha descomposición. Este algoritmo puede ser muy útil para no tener que reentrenar todo el conjunto cada vez que se hace un nuevo experimento. Adicionalmente, mediante el aprendizaje incremental, se puede observar la relevancia de una nueva muestra en el conjunto.

Por último, se ha propuesto un nuevo algoritmo de selección de características por bloques con un doble objetivo. Por un lado trata de optimizar los clasificadores, eliminando la información que no es útil, de forma que se construyen clasificadores con menos carga computacional y más exactos. Por otro lado, proporciona una información

muy útil para determinar qué zonas de las señales de entrada son las que están proporcionando mejor información, siendo esto de gran importancia para mejorar la parte sensora.

Capítulo 5

Mejoras en las máquinas de vectores soporte

En el capítulo anterior, en el que se realizó una comparación entre los diferentes métodos de clasificación, una de las conclusiones obtenidas es la flexibilidad que ofrecen los métodos kernel para la clasificación de señales procedentes de sistemas de nariz y lengua electrónica, ya que permiten trabajar con kernels lineales cuando el problema de clasificación puede ser resuelto como tal o con kernels no lineales, de forma que para aquellos problemas más complejos se pueden conseguir tasas de acierto muy elevadas.

Entre los métodos kernel propuestos se introdujeron las LS-SVM o las RVM con las que, siguiendo los resultados obtenidos, podemos ganar en algunos casos en tasa de acierto media con el primer método o en número de operaciones a evaluar en otros casos cuando se utilizan RVM. Sin embargo, dichas ganancias suelen producirse incrementando el número de operaciones necesarias en el primer caso o deteriorando la tasa de acierto en el segundo. Las SVM se sitúan como una solución de compromiso, pero es conveniente reducir aún más el número de operaciones y profundizar en el ajuste de los hiperparámetros.

A lo largo de este capítulo se proponen diversas ideas para conseguir una mejor tasa de acierto de las SVM al tiempo que se reduce el número de operaciones necesario para evaluar nuevas muestras. Por otro lado, en el capítulo anterior se utilizó como estrategia para problemas multi-clase una estrategia uno contra todos. En este capítulo se proponen otras estrategias para cubrir los objetivos de maximización de la tasa de acierto y reducción del número de operaciones. Aunque en las ideas expuestas se habla de su aplicación para las SVM, la extrapolación a otros métodos kernel es en muchos casos inmediata.

5.1. Ajuste de hiperparámetros con métodos estadísticos

Una de las conclusiones más importantes del capítulo anterior era la importancia del ajuste de los parámetros del kernel a utilizar y de la constante C denominados habitualmente hiperparámetros de una SVM [Scholkopf01] y como se ha hecho en el anterior capítulo se utiliza la notación $\boldsymbol{\theta}$ para referirnos al vector de hiperparámetros asociado. En los capítulos anteriores de esta tesis, el procedimiento de optimización de estos parámetros ha sido optimizar cada uno de ellos por separado, lo que presupone la independencia de todos los hiperparámetros. Sin embargo, esta suposición no es correcta desde un punto de vista teórico, ya que la constante de regularización C controla la importancia en el funcional (2.57) de los errores y éstos toman más o menos valor dependiendo de los parámetros del kernel.

Hay que destacar que el uso de kernels más complejos requiere de una optimización de mayor número de variables. Así, supongamos que tenemos la información procedente compuesta de un sistema con j sensores. En lugar de utilizar la estrategia de concatenación del apartado anterior podemos decir que cada patrón estará formado por una matriz $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_j\}$, donde cada uno de los vectores $\boldsymbol{x}_i$ contiene la información del sensor i. La versión del kernel RBF necesario para poder aplicar la teoría de las SVM será:

$$\kappa(\boldsymbol{X}, \boldsymbol{Y}) = \prod_{i=1}^{j} \kappa_i(\boldsymbol{x}_i, \boldsymbol{y}_i) = \prod_{i=1}^{j} exp\left(\gamma_i \|\boldsymbol{x}_i - \boldsymbol{y}_i\|^2\right) \tag{5.1}$$

Además del anterior caso, podemos incluir también a los sistemas compuestos por un único sensor cuya información compone un patrón $\boldsymbol{x} \in \mathbb{R}^d$, pero ponderar cada característica por una sensibilidad diferente utilizando un kernel del estilo:

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = exp\left(\sum_{i=1}^{d} \gamma_i (x_i - y_i)^2\right) \tag{5.2}$$

Esta técnica, conocida como determinación de relevancia automática (ARD de *Automatic Relevance Determination*) [Grandvalet02], proporciona también un método para conocer la importancia de las características y componer un esquema de selección y extracción de las mismas.

Antes de proceder al proceso de optimización de los hiperparámetros $\boldsymbol{\theta}$, se definirán algunos de los estimadores del error propios de las SVM para poder optimizar éstos.

5.1.1. Estimadores del error con SVM

Además de los estimadores del error tradicionales que se vieron en el capítulo anterior como la estimación hold-out, el error cross-validation o el estimador bootstrap, se ve a continuación una serie de estimadores propios de las SVM que tratan de establecer cotas al error leave-one-out (LOO).

5.1.1.1. La cota *Radius-Margin*

Al estar calculando el error en una SVM, uno de los aspectos deseables sería relacionar la estimación del rendimiento con los conceptos de generalización del error y minimización del riesgo estructural expuestos y que sirven como base para la teoría de los clasificadores basados en vectores soporte. Una de las primeras aportaciones en este sentido se propone en [Bartlett98] indicando que existe una constante k para la cuál el error de generalización de una SVM con una probabilidad $1-\delta$ viene dado por:

$$\varepsilon = \frac{1}{l}\left(a + \sqrt{lk\left(\frac{R^2}{M^2}\log^2(l) + \log\left(\frac{1}{\delta}\right)\right)}\right) \tag{5.3}$$

donde R es el radio de la hiperesfera más pequeña que encierra todos los patrones de entrenamiento, M es el margen de separación entre clases, l es el número de patrones considerado y a es el número de muestras de entrenamiento incorrectamente clasificadas. Esta generalización del error se encuentra relacionada con la dimensión VC que sirve de base para la maximización del margen. Vapnik [Vapnik98] relaciona el error de generalización de la ecuación (5.3) con el error leave-one-out y establece que para una SVM con parámetro de bias b nulo y sin error de entrenamiento el error leave-one-out viene dado por

$$\varepsilon \leq \frac{1}{l}\frac{4R^2}{M^2} \tag{5.4}$$

Por lo que no solamente es importante maximizar el margen, sino minimizar el radio de la hiperesfera más pequeña que encierra todos los puntos. Si bien en el espacio de entrada de los patrones de entrenamiento no es posible actuar sobre el radio, cuando aplicamos un kernel se debe recordar que implícitamente se produce una transformación de los datos hacia un espacio de Hilbert. Los hiperparámetros propios del kernel hacen que el radio de la hiperesfera mencionada varíe en función de los mismos. El cálculo del radio de la hiperesfera se propone en [Shawe-Taylor04] como un problema de optimización cuya resolución puede ser efectuada con ligeras modificaciones por los algoritmos de solución del problema de optimización propio de las SVM.

La importancia de la cota *Radius-Margin* es que establece una relación entre la teoría de minimización del riesgo estructural, el rendimiento de una SVM y establece

la asociación con el error leave-one-out. Sin embargo, su formulación se hace suponiendo únicamente conjuntos separables con un parámetro $C = \infty$. Aunque existen estimadores como los descritos en [Duan03] o [Chung03] éstos requieren suponer que los patrones están distribuidos de forma esférica, por lo que hay que hacer una serie de ajustes que complican el uso de esta cota del error.

5.1.1.2. El estimador $\xi\alpha$

Al igual que la cota *Radius-Margin*, el estimador $\xi\alpha$ [Joachims00] trata de buscar una cota superior al error leave-one-out por las razones anteriormente expuestas. Partiendo de todo el conjunto de entrenamiento, se obtendrá una función de decisión

$$f^0(\boldsymbol{x}) = \sum_{i=1}^{nsv^0} \alpha_i^0 y_i \kappa\left(\boldsymbol{x}, \boldsymbol{sv}_i^0\right) + b^0 \tag{5.5}$$

Mientras que como resultado de eliminar la muestra p en el conjunto de entrenamientos, la función de decisión será

$$f^p(\boldsymbol{x}) = \sum_{i=1}^{nsv^p} \alpha_i^p y_i \kappa\left(\boldsymbol{x}, \boldsymbol{sv}_i^p\right) + b^p \tag{5.6}$$

El estimador $\xi\alpha$ se basa en la siguiente desigualdad:

$$y_p f^p(\boldsymbol{x}_p) \geq y_p \left(\sum_{i=1, i\neq p}^{nsv^0} \alpha_i y_i \kappa\left(\boldsymbol{x}_p, \boldsymbol{sv}_i^0\right) + b^0 \right) - \alpha_p B^2 \tag{5.7}$$

donde $B^2 \geq \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i)$ es una cota superior a cualquier valor de la diagonal de la matriz kernel. En el proceso leave-one-out, una muestra $\boldsymbol{x}_p$ producirá error si $y_p f^p(\boldsymbol{x}_p) < 0$ y por tanto, solo se puede producir error si

$$y_p \left(\sum_{i=1, i\neq p}^{nsv^0} \alpha_i y_i \kappa\left(\boldsymbol{x}_p, \boldsymbol{sv}_i^0\right) + b^0 \right) - \alpha_p B^2 \leq 0 \tag{5.8}$$

o de forma equivalente

$$y_p f^0(\boldsymbol{x}_p) - \alpha_p \kappa(\boldsymbol{x}_p, \boldsymbol{x}_p) - \alpha_p B^2 \leq 0 \tag{5.9}$$

Aprovecharemos que en el caso de las SVM, solo los vectores soporte pueden introducir error en el proceso leave-one-out. Los vectores soporte tipo 1 ($0 < \alpha_p^0 < C$) se encuentran sobre el margen y por tanto la desigualdad KKT para ellos se transforma en igualdad. Esto hace que $y_p f^0(\boldsymbol{x}_p) = 1$ y por tanto la desigualdad (5.9) implica que un vector soporte de tipo 1 solo puede cometer error si se cumple

$$\alpha_p^0 \kappa(\boldsymbol{x}_p, \boldsymbol{x}_p) + \alpha_p B^2 \geq 1, \quad \boldsymbol{x}_p \in X_{sv1}^0 \tag{5.10}$$

Dado que se ha impuesto la condición $B^2 \geq \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i)$ la expresión anterior puede ser simplificada como

$$2\alpha_p^0 B^2 \geq 1, \quad \boldsymbol{x}_p \in X_{sv1}^0 \tag{5.11}$$

Para vectores soporte tipo 2, $\alpha_p^0 = C$, se cumple la expresión $y_p f^0(\boldsymbol{x}_p) = 1 - \xi_p^0$ y siguiendo el mismo razonamiento que en el caso de los vectores soporte tipo 1, tendremos que los vectores soporte tipo 2 solo pueden cometer error en el proceso de leave-one-out si cumplen

$$2\alpha_p^0 B^2 \geq 1 - \xi_p^0, \quad \boldsymbol{x}_p \in X_{sv2}^0 \tag{5.12}$$

Puesto que para los vectores soporte tipo 1 se cumple que $\xi_p^0 = 0$ podemos agrupar las expresiones (5.11) y (5.12) en

$$\left(2\alpha_p^0 B^2\right) + \xi_p \geq 1^0, \quad \boldsymbol{x}_p \in X_{sv}^0 \tag{5.13}$$

y por tanto solo los vectores soporte que satisfagan la condición anterior podrán producir error en el proceso leave-one-out. Esto nos establece una cota superior del error LOO de la forma

$$\hat{\varepsilon}_{LOO} = \frac{\sum_{i=1}^{Nsv} u\left(\left(2\alpha_p^0 R^2\right) + \xi_p - 1\right)}{l}, \quad \boldsymbol{x}_i \in X_{sv} \tag{5.14}$$

La cota propuesta al error LOO, es un estimador muy sesgado del mismo, ya que es un estimador conservador en el sentido de que todos los posibles errores son considerados como tales. Sin embargo, proporciona un método rápido de descartar vectores soporte en el proceso leave-one-out total.

5.1.1.3. Estimador basado en Span

En [Vapnik00] se introduce un nuevo estimador del error leave-one-out basado en un nuevo concepto denominado *span* (S) de los vectores soporte. Para entender este concepto, supongamos que entrenamos un conjunto $\boldsymbol{X}$ con un kernel lineal y como resultado se obtiene un subconjunto de vectores soporte $\boldsymbol{X}_{sv}$ con coeficientes de Lagrange asociados $\boldsymbol{\alpha^0} = \{\alpha_1^0, \alpha_2^0, ..., \alpha_{Nsv}^0\}$. Dentro del subconjunto X_{sv} distinguiremos aquellos vectores soporte de tipo 1 $(\boldsymbol{X}_{sv1})$ para los cuales

$$0 < \alpha_i^0 < C, \quad i = 1, 2, ...Nsv1 \tag{5.15}$$

y aquellos soporte de tipo 2 $(\boldsymbol{X}_{sv2})$ para los cuales

$$\alpha_j^0 = C, \quad j = 1, 2, ..Nsv2 \tag{5.16}$$

Para cada uno de los vectores soporte se define el conjunto $\wedge_p$ como una combinación lineal del resto de vectores soporte sujeta a restricciones:

$$\wedge_p = \{ \sum_{i=1, i\neq p}^{Nsv} \lambda_i \boldsymbol{x}_i : \sum_{i=1, i\neq p}^{Nsv} \lambda_i = 1, \forall i \neq p, 0 \leq \alpha_i + y_i y_p \lambda_i \leq C\} \tag{5.17}$$

El conjunto $\wedge_p$ contiene todos los posibles planos de separación lineales candidatos a ser una solución del problema de optimización de las SVM supuesto que no cambia el conjunto de vectores soporte, dado que se tiene que cumplir la restricción:

$$\sum_{i=1}^{Nsv} \alpha_i y_i = 0 \tag{5.18}$$

Se define el *span* de un vector $\boldsymbol{x}_p \in \boldsymbol{X}_{sv}$ como la mínima distancia que existe entre el vector $\boldsymbol{x}_p$ y el conjunto $\wedge_p$.

$$S_p^2 = min\{\left(\sum_{i=1, i\neq p}^{n} \lambda_i \boldsymbol{x}_i \right)^2, i \in X_{sv1} : \lambda_p = -1, \sum_{i=1}^{n} \lambda_i = 0, 0 \leq \alpha_i + y_i y_p \lambda_i \leq C\} \tag{5.19}$$

Un vector $\boldsymbol{x}_p \in X_{sv1}$ introducirá error en el proceso leave-one-out si cumple

$$\alpha_p^0 S_p \operatorname{máx}\left(D, \frac{1}{\sqrt{C}} \right) \geq 1 \tag{5.20}$$

donde D es el diámetro de la hiperesfera más pequeña que encierra todos los puntos. Por tanto, existe una cota superior del error producido por el estimador leave-one-out,

$$\hat{\varepsilon}_{LOO} \leq \frac{Nsv2 + \sum_{i=1}^{n} u\left(\alpha_p^0 S_p \operatorname{máx}\left(D, \frac{1}{\sqrt{C}} \right) - 1 \right)}{l}, \quad \boldsymbol{x}_i \in X_{sv1} \tag{5.21}$$

La cota anterior presenta como ventaja un cálculo exacto del error LOO introducido por los vectores tipo 1. Sin embargo dicha cota considera que todos los vectores tipo 2 introducen error en el proceso leave-one-out, lo que no es correcto. Por otro lado, el cálculo del span definido según (5.19) es tan costoso como calcular el error leave-one-out de una forma exacta, además de tener que calcular el diámetro de la hiperesfera más pequeña que encierra todos los puntos. Por este motivo, en [Chapelle02a] supone que durante el proceso leave-one-out, al retirar un vector soporte $\boldsymbol{x}_p \in X_{sv}$ la solución del entrenamiento con el conjunto $\boldsymbol{X}^{\setminus p}$ hace que los conjuntos $\boldsymbol{X}_{sv1}$ y $\boldsymbol{X}_{sv2}$ sean iguales a la solución de entrenamiento con todos los vectores soporte, con la excepción del vector retirado en el proceso leave-one-out. Esta suposición hace que el error *leave-one-out* no sea exacto, pero el cálculo del span se pueda simplificar mucho. Denominando como K_{sv1} a la matriz kernel de los vectores soporte tipo 1, definimos la matriz,

$$\widetilde{\boldsymbol{K}}_{sv1} = \begin{pmatrix} \boldsymbol{K}_{sv1} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{pmatrix} \tag{5.22}$$

y su inversa $\boldsymbol{A} = \widetilde{\boldsymbol{K}}_{sv1}^{-1}$, se calcula el span de los vectores soporte tipo 1 como:

$$S_p^2 = \frac{1}{A(p,p)}, \quad 0 < \alpha_p < C \tag{5.23}$$

Bajo la aproximación realizada sobre el mantenimiento de conjuntos de vectores soporte, también es posible calcular el span de los vectores soporte tipo 2 como:

$$S_p^2 = \kappa(\boldsymbol{x}_p, \boldsymbol{x}_p) - \boldsymbol{v}_p^T \boldsymbol{A} \boldsymbol{v}_p \tag{5.24}$$

donde $\boldsymbol{v}_p$ es un vector de dimensión $n+1$ con componentes $\kappa(\boldsymbol{x}_p, \boldsymbol{x}_i), \quad i \in \boldsymbol{X}_{sv1}$ y cuyo último elemento es un 1.

Una vez estimado el span de un vector soporte, bajo la suposición de que el conjunto de vectores soporte tipo 1 y tipo 2 se mantiene igual en el proceso de leave-one-out, el error se estima como:

$$\hat{\varepsilon}_{LOO} = \frac{\sum_{i=1}^{Nsv} u\left(\alpha_i^0 S_i^2 - y_i f^0(\boldsymbol{x}_i)\right)}{l}, \quad \boldsymbol{x}_i \in X_{sv} \tag{5.25}$$

5.1.2. Función de optimización propuesta

En el marco de esta tesis se propone una función de estimación del error que además tendrá en cuenta el número de vectores soporte asociado a los hiperparámetros dados. La función de estimación se basa en el cálculo exacto del error leave-one-out realizado de forma eficiente mediante la aplicación de una serie de técnicas que reducen sensiblemente el número de operaciones a realizar.

Partiendo de la condición del estimador $\xi\alpha$, durante el proceso leave-one-out solo pueden producir error aquellos patrones que sean vectores soporte y cumplan la condición (5.13). Por otro lado, aquellos vectores soporte de tipo 2, en los que $\alpha_i = C$, que resulten incorrectamente clasificados en su entrenamiento producirán error seguro durante el proceso leave-one-out por lo que no es necesario realizar el entrenamiento sin cada una de estas muestras. Esto hace que solo se deba comprobar una pequeña parte del conjunto de entrenamiento.

Para los vectores soporte que no cumplan las condiciones anteriores y por tanto haya que realizar su proceso leave-one-out se debe comprobar si introducen error, de forma que se cumpla:

$$y_p f(\boldsymbol{x}_p) = y_p \left[\sum_{i=1}^{l-1} \alpha_i^p y_i \kappa(\boldsymbol{x}_p, \boldsymbol{x}_i) + b^p \right] < 0 \tag{5.26}$$

siendo $[\boldsymbol{\alpha}^p, b^p]$ el resultado del entrenamiento sin la muestra $\boldsymbol{x}_p$. La primera técnica que usaremos para reducir el coste computacional del proceso leave-one-out es la técnica conocida como la inserción de semilla del vector $\boldsymbol{\alpha}^p$ (*alpha seeding*) y fue propuesta en [DeCoste00] y [Lee04]. Sea el vector $\overline{\boldsymbol{\alpha}}^p$ una posible solución inicial del problema de optimización de las SVM, en un entrenamiento normal se parte del vector $\overline{\boldsymbol{\alpha}}^p$ nulo para llegar a la solución final $\boldsymbol{\alpha}^p$. En su lugar, en la técnica alpha seeding se parte de una posible solución del vector final $\overline{\boldsymbol{\alpha}}^p$ no nula, formada a partir del vector encontrado en el entrenamiento con todas las muestras $\boldsymbol{\alpha}^0$, de forma que el multiplicador de Lagrange α_p^0 se reparte asegurando que se cumple:

$$\sum_{i=1}^{l-1} \alpha_i^p y_i = 0 \tag{5.27}$$

En nuestro caso, la forma en cómo se reparte este multiplicador de Lagrange varía respecto de cómo se realizaba en los trabajos anteriormente citados. El algoritmo propuesto para la técnica de inserción de semilla consta de los siguientes pasos:

1. Inicializar $\lambda_p = 1$, $\overline{\alpha}_i^p = \alpha_i^0 \quad \forall i \neq p$.

2. Buscar el vector más próximo que cumpla:

$$\begin{aligned} &\min_i \|\boldsymbol{x}_p - \boldsymbol{x}_i\| \\ &\text{Sujeto a:} \\ &0 < \overline{\alpha}_i^p < C \\ &y_p = y_i \end{aligned} \tag{5.28}$$

3. Si se cumple la condición

$$\overline{\alpha}_i^p + \alpha_p^0 \lambda_p < C \tag{5.29}$$

actualizamos los valores

$$\begin{aligned} &\overline{\alpha}_i^p = \overline{\alpha}_i^p + \alpha_p^0 \lambda_p \\ &\lambda_p = 0 \end{aligned} \tag{5.30}$$

Si no se cumple la condición (5.29) se actualizan los valores mediante:

$$\overline{\alpha}_i^p = C \quad \lambda_p = \frac{C - \alpha_p^0}{\alpha_p^0} \tag{5.31}$$

4. Si $\lambda_p = 0$ se termina el algoritmo. En caso contrario se repite desde el paso 2 hasta que $\lambda_p = 0$. Si no existen vectores que cumplan la condición impuesta en (5.28) se relaja la condición y se buscan entre los vectores que cumplan:

$$\overline{\alpha}_i^p < C \quad y_p = y_i \tag{5.32}$$

De esta forma, se parte de una solución inicial próxima al vector solución final $\boldsymbol{\alpha}_p$ reduciendo el número de iteraciones medio necesarias para llegar a esta última.

La segunda técnica aplicada para acelerar el proceso es la reducción del conjunto de búsqueda o grupo de patrones que puede cambiar el valor de la semilla inicial. Esta técnica es conocida como reducción del conjunto (*shrinking*) [Fan05] y se emplea habitualmente en el entrenamiento de las SVM. Definimos la función

$$g(\boldsymbol{x}_j) = -\sum_{i=1}^{l-1} \overline{\alpha}_i^p y_i \kappa(\boldsymbol{x}_j, \boldsymbol{x}_i) + y_j \tag{5.33}$$

y los conjuntos:

$$I_{up}(\overline{\boldsymbol{\alpha}}^p) = \begin{cases} \overline{\alpha}_i^p < C \text{ Si } y_i = 1 \\ \overline{\alpha}_i^p > 0 \text{ Si } y_i = -1 \end{cases} \tag{5.34}$$

$$I_{low}(\overline{\boldsymbol{\alpha}}^p) = \begin{cases} \overline{\alpha}_i^p < C \text{ Si } y_i = -1 \\ \overline{\alpha}_i^p > 0 \text{ Si } y_i = 1 \end{cases} \tag{5.35}$$

Las únicas muestras que pueden cambiar el valor de su multiplicador α_i^p respecto del valor de la semilla inicial $\overline{\alpha}_i^p$ deben estar contenidas en el conjunto

$$\boldsymbol{\Lambda} = \{i | M(\overline{\boldsymbol{\alpha}}^p) \leq g(\boldsymbol{x}_i) \leq m(\overline{\boldsymbol{\alpha}}^p)\} \tag{5.36}$$

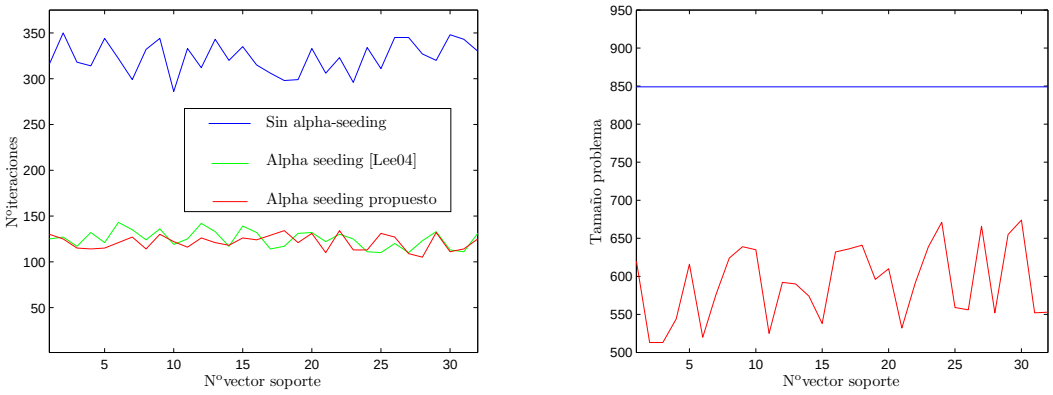
donde

$$m(\overline{\boldsymbol{\alpha}}^p) = \max_{k \in I_{up}} g(\boldsymbol{x}_k) \quad M(\overline{\boldsymbol{\alpha}}^p) = \min_{k \in I_{low}} g(\boldsymbol{x}_k) \tag{5.37}$$

Con los pasos anteriores se consigue reducir el tiempo de cómputo sobre el cálculo de aquellas muestras que precisan del entrenamiento del conjunto con excepción de la muestra $\boldsymbol{x}_p$.

Para mostrar las cualidades del método propueto, supongamos que del conjunto de datos de clasifiación de alcholes nos centramos en el clasificador que separa la clase etanol contra el resto. Si utilizáramos un clasificador que no fuera una SVM estimar el error leave one out supondría realizar $l = 849$ entrenamientos. Sin embargo, tras aplicar las condiciones de estimador $\xi\alpha$ y obviar aquellos vectores soporte tipo 2 que producen error, solo hay que comprobar 32 vectores soporte mediante un entrenamiento completo. En la figura (5.1(a)) se aprecia el número de iteraciones necesarias para cada uno de los 32 clasificadores en los que hay que entrenar y probar con el vector soporte que se ha dejado fuera del proceso. Dicho número de iteraciones para tres casos distintos: en el primero no se considera la técnica alpha-seeding, en el segundo se considera el método alpha-seeding propuesto en [Lee04] y en el tercer caso se considera el método

de alpha-seeding propuesto en esta tesis. Se puede observar cómo el método propuesto proporciona resultados ligeramente mejores que el método propuesto en [Lee04], lo que ha sido contrastado para varios clasificadores. Además, el método propuesto tiene la ventaja de aplicar la técnica de reducción del conjunto, de forma que para cada uno de los entrenamientos de la figura anterior no es necesario trabajar con las $l-1$ muestras. En la figura (5.1(b)) se muestra el tamaño del conjunto de entrenamiento para cada uno de los 32 clasificadores. Para entender mejor la comparación, se muestra en línea continua con $l-1$ muestras con las que se trabaja en otros métodos diferentes del propuesto.



(a) *Reducción en número de iteraciones por alpha-seeding.* **(b)** *Reducción en el conjunto de entrenamiento.*

Figura 5.1: Aplicación del método de cálculo leave-one-out propuesto.

Además de estimar el error, en el capítulo anterior se señalaba la importancia de reducir el número de operaciones en la fase de test, lo que se consigue en el caso de las SVM teniendo que evaluar menos vectores soporte. Por tanto, el objetivo de la función propuesta no solo es optimizar los hiperparámetros respecto al error estimado, sino también en cuanto al número de vectores soporte proporcionado. La función propuesta de optimización tiene la forma:

$$J(\boldsymbol{\theta}) = \begin{cases} Nsv(\boldsymbol{\theta})/l^2 & \text{Si} \quad \varepsilon_{LOO}(\boldsymbol{\theta}) \leq \mu \\ \varepsilon_{LOO}(\boldsymbol{\theta}) + Nsv(\boldsymbol{\theta})/l^2 & \text{Si} \quad \varepsilon_{LOO}(\boldsymbol{\theta}) > \mu \end{cases} \tag{5.38}$$

de forma que si el error leave-one-out ε_{LOO} es menor que un umbral μ, se considera el vector de hiperparámetros que menor número de vectores soporte presenta, mientras que para valores superiores se trata de minimizar en primer lugar el error y solo ante dos combinaciones con el mínimo error se seleccionará aquella con menor número de vectores soporte. Para formar la función a optimizar desarrollada en (5.38) se ha aprovechado que la función de error leave-one-out es escalonada con una diferencia mínima

de $1/l$ entre dos errores consecutivos y que la cantidad $Nsv/l \leq 1$, motivo por el que se suma la cantidad Nsv/l^2 que será siempre menor que el escalón mínimo de la función de error leave-one-out.

La aplicación de esta función se realiza para cada clasificador binario que compone una SVM, de forma que por cada uno de ellos se prueba en primer lugar si es aplicable el kernel lineal. Si el error obtenido con dicho clasificador no supera el valor μ establecido se seleccionará este kernel para ese clasificador binario, ya que solo se necesita el equivalente de un vector soporte para evaluarlo. Si procede, se prueba con el kernel RBF aplicando y se obtiene el error asociado. En cualquier caso, si como resultado de la optimización del kernel RBF se obtuviera un resultado peor que con el kernel lineal, aún habiendo superado el valor μ, se utilizaría el kernel lineal.

5.1.3. Métodos descritos de optimización de hiperparámetros

Una vez que se han revisado los estimadores del error, tanto los vistos en el capítulo anterior como las cotas del error leave-one-out y su cálculo eficiente descritos en la sección anterior, se describe a continuación los métodos de búsqueda de los hiperparámetros que minimizan dichos errores, analizando en cada uno de ellos sus ventajas y sus inconvenientes. Esta búsqueda puede realizarse considerando la variación de los parámetros bien de forma lineal o de forma logarítmica.

5.1.3.1. Método de rejilla

La búsqueda por rejilla o *grid search* es el método más extendido cuando no se considera cada parámetro de forma independiente. Para su ejecución se discretiza el espacio de hiperparámetros $\boldsymbol{\theta}$ de forma que solo se permite a cada parámetro tomar una serie de valores m_i dentro de un rango definido. Así el nuevo espacio de hiperparámetros contendrá un número de combinaciones N definido por:

$$N = \prod_{i=1}^{h} m_i \tag{5.39}$$

donde h es el número total de hiperparámetros a ajustar incluyendo el parámetro de regularización C. Para cada una de estas combinaciones se deberá estimar el error de generalización, de forma que si consideráramos solo dos hiperparémtros a optimizar se formaría una rejilla de valores en dos dimensiones. La ventaja que presenta este método es que la función de evaluación no tiene requisitos de derivabilidad y además puede incorporar términos interesantes como la inclusión del número de vectores soporte propuesta en la sección anterior. La desventaja de este método es el número de combinaciones necesarias que hay que ejecutar para tener una buena resolución. Existen variantes del mismo como la búsqueda iterativa de rejilla [Gestel04], mostrado

en la figura (5.2), donde se parte de un número de divisiones pequeño y se encuentra la combinación que mejor resultado obtenga. A continuación se establece otra rejilla cuyos límites son los de la división ganadora en el paso anterior. Este proceso se puede seguir iterando hasta conseguir una buena resolución, pero puede caer fácilmente en un mínimo local y su utilización para kernels con un número de hiperparámetros elevados, como los propuestos en las ecuaciones (5.1) y (5.2), adquiere un grado de complejidad elevado.

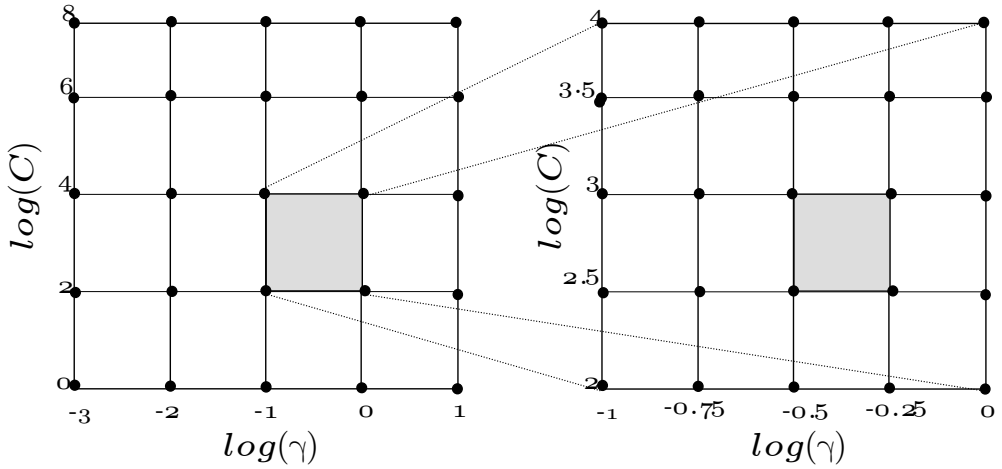


Figura 5.2: Búsqueda por rejilla iterativa con dos hiperparámetros.

5.1.3.2. Optimización por descenso del gradiente

Para evitar los problemas planteados en la optimización por rejilla, algunos autores han sugerido utilizar una optimización por descenso del gradiente. Para poder emplear este método, la función de evaluación del error debe cumplir dos premisas:

- La función debe ser derivable al menos en un rango establecido para los hiperparámetros.
- La derivada de la función con respecto a los hiperparámetros debe ser conocida de antemano.

En [Chapelle02b] se utiliza este método con las cotas del span y radius-margin, pero solamente puede ser aplicado a las L2-SVM para poder incluir el parámetro de regularización C como parte del kernel. Este truco no se puede realizar en las L1-SVM que son las seleccionadas para esta tesis dado que proporcionan menos vectores soporte. Para poder aplicar el método del descenso del gradiente a las L1-SVM en [Chung03] se utiliza una cota derivada del estimador radius-margin que sí es derivable respecto a

sus hiperparámetros. No obstante, esta expresión presenta el inconveniente de no ser convexa, por lo que pueden presentarse múltiples mínimos locales. Además, como se señala en [Chapelle99], la cota radius-margin es efectiva si los puntos se sitúan en el espacio transformado en una hiperesfera, mientras que si se sitúan en una hiperelipsoide debido a la mayor varianza de uno de los ejes la cota fallará.

Adicionalmente a los problemas anteriores, ninguno de los métodos basados en descenso por gradiente controla el número de vectores soporte, por lo que no seleccionará los hiperparámetros que en igualdad de condiciones proporcionen un número menor de éstos y por tanto traten de reducir el número de operaciones en la fase de test.

5.1.4. Algoritmos genéticos

El uso de algoritmos genéticos ya fue introducido en esta tesis aplicado a la selección de características. En dicha aplicación cada cromosoma o posible solución se codificaba como una cadena de bits en la que la presencia de la característica se codificaba como un 1, mientras que la ausencia de la misma se caracteriza por un cero. En este caso la aplicación de los algoritmos genéticos tratará de buscar la combinación de los elementos de $\boldsymbol{\theta}$ que optimice la ecuación (5.38) que será considerada como *función de fitness*. Recientes trabajos como [Lessmann06] y [Zhang08] han intentado aplicar un esquema evolutivo al problema de selección de hiperparámetros de un kernel optimizando el error proporcionado por el estimador $\xi\alpha$ y la validación cruzada respectivamente como funciones de fitness. Además de los problemas ya descritos sobre el uso de dichos estimadores para la selección de hiperparámetros, el problema fundamental consiste en la codificación de cada uno de los elementos de $\boldsymbol{\theta}$ a un espacio discreto con n_i bits cada uno. Si se codifican con pocos bits el método será muy similar al método de la rejilla sin proporcionar ventajas sobre éste. Por contra, si el número de bits es excesivo el algoritmo genético requerirá de un número elevado de generaciones para converger a una solución óptima. Además, la influencia de la operación de mutación es fuertemente dependiente del número de bits empleado [Miller04].

El algoritmo que se propone a continuación se basa también en el *algoritmo simple genético* (SGA) pero los cromosomas son las variables θ_i en espacio continuo sin necesidad de ser discretizadas. Para poder trabajar con estas variables continuas, las operaciones de mutación y recombinación o cruce deben ser redefinidas según:

Recombinación. La operación de recombinación en el espacio continuo se forma mediante una combinación lineal de los padres. Así, supongamos que tenemos dos individuos $\boldsymbol{\theta}_1$ y $\boldsymbol{\theta}_2$ que han sido seleccionados para tener descendientes. La operación de recombinación viene dada por

$$\boldsymbol{\theta}_h = \boldsymbol{\theta}_1 + \eta R\left(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1\right) \tag{5.40}$$

donde $\eta \in [0,1]$ es un número aleatorio con distribución uniforme y R es un número que limita la zona de descendencia. Si $R \in [0,1]$, los descendientes se encontrarán dentro del hipercubo cuyos vértices opuestos forman los padres.

Mutación. Para poder aplicar la operación de mutación nos basaremos en dos conceptos:

- Mutación en dos pasos. En este caso, tenemos una primera etapa donde se decide sobre cada variable si existirá mutación o no.
- Ruido Gaussiano. Si se determina que una variable tiene que mutar, se cambia su valor por otro generado mediante ruido gaussiano de media cero a cada una de las variables que forman el cromosoma o solución del individuo. La varianza de dicho ruido puede ser constante, tomando un número muy reducido o dependiente de la generación en la cuál nos encontremos de la forma.

$$\sigma_{k+1} = \sigma_k \left(1 - \lambda \frac{k}{N_{Generaciones}}\right) \tag{5.41}$$

De esta forma, en las primeras generaciones la varianza es mayor, tratando de explorar todas las zonas del espacio de búsqueda para evitar caer en un mínimo local. En las últimas generaciones nos centramos en el mínimo calculado hasta el momento, con poca o nula variación sobre las soluciones. El parámetro $\lambda \in [0,1]$ controla cómo afecta el paso de las generaciones en la variación del ruido.

5.1.5. Simulated annealing

Al igual que los algoritmos genéticos, esta técnica ya fue introducida para la selección de características. En este caso también extenderemos su uso para trabajar con variables continuas. Casi todos los autores que han trabajado con esta técnica coinciden en señalar que la característica más importante es el considerado esquema de enfriamiento. En nuestro caso, hemos utilizado una variante del algoritmo denominada *Adaptive Simulated Annealing* (ASA) [Ingber93] . El algoritmo aplicado a nuestro problema se resume en los siguientes pasos:

1. Inicializar la temperatura a un valor T_0. Seleccionar de formar aleatoria una posible combinación del vector $\boldsymbol{\theta}$ y calcular el estimador del error descrito en (5.38). Guardar el vector como $\boldsymbol{\theta}^g$ y el valor del error estimado como $\hat{\varepsilon}(\boldsymbol{\theta}^g)$.

2. Seleccionar un vector $\boldsymbol{\theta}^n$ vecino a $\boldsymbol{\theta}^g$ mediante una función de vecindad en la que cada componente de dicho vector se modifica según:

$$\theta_i^n = \theta_i^g + sgn(u_i - 0{,}5)T\left[\left(1+\frac{1}{T}\right)^{|2u_i-1|} - 1\right]R_j \tag{5.42}$$

donde $u_j \in [0,1]$ es una variable aleatoria uniforme y R_j representa el rango admitido de la variable θ_j. En la generación de un vecino se comprobará que la nueva componente θ_i^n no supera los límites de los rangos establecidos. Mediante esta función de vecindad, conseguimos que a medida que se va enfriando la temperatura, además de ser más improbable que pueda ser aceptada una solución de mayor error, la probabilidad de buscar por zonas más alejadas de la solución actual es menor. Mediante este esquema se consigue que al principio se evite caer en mínimos locales, mientras que según se va enfriando la temperatura, el algoritmo buscará con alta probabilidad el mínimo dentro de una región determinada.

Una vez calculado el nuevo vector $\boldsymbol{\theta}^n$ se calcula la estimación de su error $\hat{\varepsilon}(\boldsymbol{\theta}^n)$.

3. Si se produce la condición $\hat{\varepsilon}(\boldsymbol{\theta}^n) < \hat{\varepsilon}(\boldsymbol{\theta}^g)$, entonces se guarda el cambio:

$$\begin{aligned} \boldsymbol{\theta}^g &\leftarrow \boldsymbol{\theta}^n \\ \hat{\varepsilon}(\boldsymbol{\theta}^g) &\leftarrow \hat{\varepsilon}(\boldsymbol{\theta}^n) \end{aligned} \tag{5.43}$$

4. Si no se produce la condición anterior aceptamos el cambio expuesto en la expresión (5.43) con una probabilidad:

$$p = exp\left(-\left(\hat{\varepsilon}(\boldsymbol{\theta}^n) - \hat{\varepsilon}(\boldsymbol{\theta}^g)\right)/T\right) \tag{5.44}$$

5. Si hemos alcanzado el número máximo de iteraciones devolvemos la solución de $\boldsymbol{\theta}$ que mejor resultado hubiera obtenido, no teniendo necesariamente que coincidir con $\boldsymbol{\theta}^g$. Si no se ha alcanzado el número máximo de iteraciones, se actualiza la temperatura y se vuelve a repetir el algoritmo.

El esquema de enfriamiento utilizado responde a la siguiente ecuación:

$$T(k) = T_0 exp(-\eta \upsilon k) \tag{5.45}$$

donde η es una constante calculada para que la temperatura en la última iteración coincida con la temperatura final deseada y υ es una constante de enfriamiento que controlará cómo decrece la temperatura a medida que avanzamos en cada iteración. Debe notarse que simulated annealing es un algoritmo no poblacional [Holger04]. Sin embargo, en esta tesis se propone utilizar un conjunto de soluciones vecinas por cada iteración o temperatura bajo estudio. De esta forma, el vector $\boldsymbol{\theta}^n$ será el que obtenga un mínimo error del conjunto de soluciones vecinas creadas para cada temperatura.

5.1.6. Optimización por colonia de hormigas

La optimización por colonia de hormigas (ACO acrónimo de *Ant Colony Optimization*) (ACO) es un método meta-heurísitco reciente basado en la denominada inteligencia de los insectos, por lo que se considera un método bioinspirado. Este método proporciona buenos resultados para solucionar problemas combinatorios difíciles, por lo que se ha utilizado en esta tesis para el ajuste de hiperparámetros cuando el número de éstos a ajustar es elevado. Este caso, ocurre cuando queremos ajustar los denominados factores de escala de la ecuación (5.2) para determinar cada uno de los coeficientes del vector $\boldsymbol{\gamma}$ y el parámetro C.

En el caso de la optimización ACO, el algoritmo cuenta con los siguientes elementos:

- Un número de hormigas artificiales. Cada hormiga viaja a través de un camino asociado a una solución del vector $\boldsymbol{\theta} = [\boldsymbol{\gamma}, C]$.
- Se discretiza el rango de valores de cada uno de los elementos del vector $\boldsymbol{\theta}$. Así, el parámetro C solo puede tomar una serie de valores establecidos y cada uno de los parámetros γ_i tomarán una serie de m valores.
- Existe un estado inicial S^0 desde donde todas las hormigas comienzan su viaje. Además, nos referimos al estado S_j^i como el estado j-ésimo asociado a la característica i, siendo $i < d$ y d la dimensión del problema. Por otro lado, nos referimos al estado S_j^C como el estado j-ésimo asociado con el parámetro C.
- Desde cada estado S_j^i a otro estado S_z^{i+1} existe un camino con los siguientes elementos:
  - La probabilidad P_{jz} de que ese camino pueda ser seleccionado por una hormiga.
  - Un valor de feromona τ_{jz} que depende del número de hormigas que hayan viajado por ese enlace hace poco tiempo.

Para explicar mejor el algoritmo nos fijamos en la figura (5.3) correspondiente a un problema con dimensión $d = 3$, a la que se ha añadido una columna de estados adicional para encontrar el valor del parámetro óptimo C. Cada hormiga comienza su viaje en el estado S^0 donde puede elegir entre los m valores de γ_1, tomando en el ejemplo de la figura un valor $m = 5$. En este movimiento inicial se ha seleccionado el estado S_j^1 con una probabilidad P_{0j}. Desde el estado S_j^i solamente puede moverse a otro estado donde seleccionará el valor γ_2 para su solución. Este movimiento se hará con una probabilidad P_{jz}^1 y proseguirá su camino. Una vez que llega al estado S_h^C seleccionará el valor del parámetro C y habrá concluido su viaje que constituirá una posible solución al vector $\boldsymbol{\theta}$.

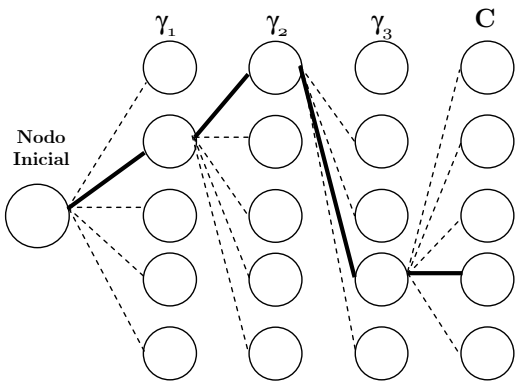


Figura 5.3: Esquema de funcionamiento del algoritmo ACO aplicado a la selección de hiperparámetros.

Para cada instante de tiempo t, hay k hormigas concurrentes viajando a través de los estados. En la figura (5.3) se ha representado el viaje, asociado a una solución del problema, de una hormiga y los posibles caminos que puede tomar en cada momento.

Una vez que se ha explicado la relación entre la selección de hiperparámetros y los caminos seguidos por cada hormiga, se describen a continuación los pasos del algoritmo diseñado:

1. Crear todos los posibles estados discretizando el espacio de variables. Asignar una distribución uniforme para todos los posibles saltos de un estado a otro.

2. Asignar la misma cantidad de feromona τ_{jz} para todos los posibles caminos.

3. Asignar k hormigas al nodo inicial. Cada hormiga seleccionará una solución j para el valor de γ_1 dependiendo del valor de la probabilidad P_{0j}. Una vez que ha hecho esta selección, el siguiente estado será elegido dependiendo de la probabilidad de los diferentes caminos hacia el siguiente estado.

4. Una vez que las k hormigas han completado su viaje, y por tanto cada una de ellas tiene asociada una posible solución, se calcula alguno de los estimadores del error descritos previamente.

5. Se seleccionan las B hormigas que hayan conseguido valores más bajos del estimador de error.

6. Se incrementa el valor de cada enlace mediante:

$$\Delta\tau_{jz} = \begin{cases} \frac{Q}{\widehat{\varepsilon_k}} \; si \; k \in B \\ 0 \; si \; k \notin B \end{cases} \quad . \tag{5.46}$$

donde Q es una constante que debe ser seleccionada a priori.

7. Conservar la solución cuyo error sea el menor hasta el momento.

8. Recalcular todas las probabilidades del camino:

$$P_{jz} = \frac{\tau_{jz}^{\lambda}}{\sum \tau_{j}^{\lambda}} \tag{5.47}$$

   donde λ es una constante a ser ajustada.

9. Realizar la evaporación del nivel de feromona:

$$\tau_{jz}^{it+1} = (1-\rho)\,\tau_{jz}^{it} \tag{5.48}$$

   donde ρ, es un coeficiente de evaporación.

10. Si se ha alcanzado el número máximo de iteraciones, finalizar el algoritmo y devolver la solución cuyo estimador del error haya resultado menor. Si no se ha alcanzado el número de iteraciones máximas, se repite desde el paso 3.

El algoritmo ACO expuesto solo tiene sentido cuando se desea utilizar sobre kernels multiexponenciales.

5.1.7. Particle swarm optimization

La optimización por enjambre de partículas o PSO [Clerc02] es un método reciente de optimización muy utilizado para la minimización de funciones. Este método está inspirado en el movimiento emergente de una bandada de pájaros en búsqueda de comida. Al igual que los anteriores métodos de búsqueda meta-heurística descritos hasta ahora, la búsqueda del mínimo se realiza mediante un proceso iterativo en el que cada vez una población o combinación de m posibles soluciones busca mediante un proceso aleatorio dirigido el óptimo de la función de evaluación. En nuestro caso, el algoritmo puede resumirse en los siguientes pasos:

1. El espacio de búsqueda de los hiperparámetros se acota entre unos valores mínimos y máximos.

2. Cada una de las partículas $\boldsymbol{\theta}_i$ está formada por una posible solución del vector de hiperparámetros.

3. En la primera generación, se asigna para cada partícula una posición inicial $\boldsymbol{\theta}_i(0)$ consistente en posibles soluciones del vector de hiperparámetros tomadas de forma aleatoria mediante una distribución uniforme del vector del espacio acotado de hiperparámetros. Además, a cada partícula le asignaremos una velocidad inicial $\boldsymbol{v}_i(0)$, siendo este valor también aleatorio.

4. Se evalúa cada una de las partículas mediante la función de error descrita en (5.38). Para cada partícula, guardaremos el valor $pbest_i$ como el mínimo alcanzado por cada partícula, además de la posición asignada a dicho valor $\boldsymbol{\theta}_{pbest_i}$. Si el valor asociado a la posición actual de cada partícula no es menor que el mínimo alcanzado hasta ahora, no se guardará nada.

5. De entre los valores calculados hasta el momento $pbest_i$ y sus posiciones asociadas $\boldsymbol{\theta}_{pbest_i}$, se guarda la mínima de todas ellas como $gbest$ o mínimo global alcanzado y su posición asociada como $\boldsymbol{\theta}_{gbest}$.

6. Si el número de generaciones es el máximo permitido se devuelven los valores $\boldsymbol{\theta}_{gbest}$ y $gbest$. Si por el contrario, el número de generaciones no ha alcanzado el máximo, las posiciones y velocidades de las partículas se modifican según:

$$\begin{aligned} \boldsymbol{v}_i(t+1) &= \phi(t)\,\boldsymbol{v}_i(t) + c_1 r_1\left(\boldsymbol{\theta}_{pbest} - \boldsymbol{\theta}_i(t)\right) + c_2 r_2\left(\boldsymbol{\theta}_{gbest} - \boldsymbol{\theta}_i(t)\right) \\ \boldsymbol{\theta}_i(t+1) &= \boldsymbol{\theta}_i(t) + v_i(t+1) \quad . \end{aligned} \tag{5.49}$$

   donde $v_i(t+1)$ es la nueva velocidad de la partícula i, c_1 y c_2 son coeficientes relacionados con la atracción a las posiciones $\boldsymbol{\theta}_{pbest_i}$ y $\boldsymbol{\theta}_{gbest}$ respectivamente, $r_{1,2} \in [0,1]$ son números aleatorios con distribución uniforme y $\phi(t)$ es la función de inercia. En este trabajo se ha seguido la función de inercia descrita en [Trelea03], con la siguiente expresión:

$$\phi(t) = \alpha_i + \frac{(\alpha_f - \alpha_i)(t)}{t_{InerciaFinal}} \tag{5.50}$$

   donde α_i es el valor de la inercia inicial, α_f es el valor de la inercia final, $t \in [1,2,...k]$ es la generación actual y $t_{InerciaFinal}$ es la generación para la cuál la inercia pasa a ser un valor constante α_f.

La ventaja del algoritmo PSO es que cada partícula o posible solución del problema se encamina usando su propia información pasada y la de sus vecinos. Este hecho hace que cada partícula vuele hacia una posición mínima, pero pueden escapar de la misma si resulta ser un mínimo local.

5.1.8. Ejecución distribuida

Todos los algoritmos expuestos en la sección anterior son fácilmente realizables en un entorno de programación paralela, por lo que se utilizó la toolbox de Matlab *Distributed Computing*. En la figura (5.4) se muestra un esquema del funcionamiento de esta herramienta. En un servidor principal se ejecuta el algoritmo de optimización

bajo consideración. En todos los algoritmos expuestos, en cada iteración hay que evaluar una serie de posibles soluciones del vector de hiperparámetros, por lo que en este momento el programa servidor contacta con el *job manager* o gestor de trabajos, de forma que se le manda evaluar la población de soluciones actual. El gestor de trabajos, tiene una serie de *workers* o trabajadores asociados encargados de recibir los posibles hiperparámetros y evaluar la función (5.38) con ellos. Hay que señalar que en una primera instancia los trabajadores deberán ser notificados sobre qué conjunto de datos se requiere la optimización, ya que los patrones de entrenamiento no se pasan por la red para evitar demoras en el cálculo. Cada ordenador puede ejecutar varios workers en paralelo, pero en nuestros experimentos hemos observado que el mejor rendimiento se alcanza lanzando un trabajador por cada núcleo disponible del procesador. Para la realización de las pruebas se dispuso de cinco ordenadores en paralelo con dos trabajadores en cada uno, por lo que supone tener diez trabajadores totales. Este hecho hace que las poblaciones de soluciones para cada algoritmo sean, en la medida de lo posible, múltiplos de diez.

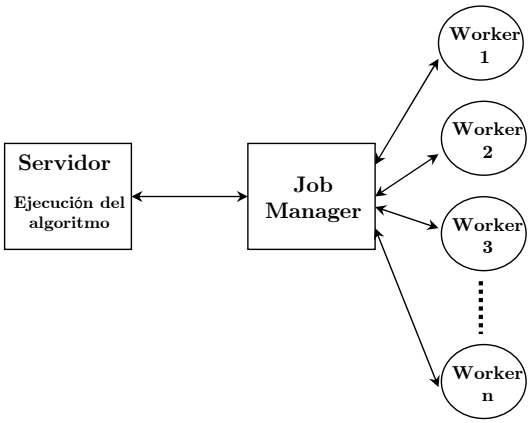


Figura 5.4: Esquema de ejecución distribuida.

5.1.9. Resultados y comparaciones

Los métodos anteriormente mostrados se aplican a cada uno de los clasificadores binarios utilizados para resolver un problema multiclase mediante SVM. Es importante destacar que todos los métodos de búsqueda propuestos en esta sección, como alternativa al método de la rejilla, se basan en ejecuciones aleatorias y por tanto no tienen por qué ofrecer los mismos resultados cada vez que se ejecutan. Uno de los conceptos más importantes en la aplicación de algoritmos meta heurísticos es la probabilidad de alcanzar una solución óptima para cada uno de ellos para un número de generaciones determinado. Sin embargo, en los problemas bajo estudio, es imposible conocer la so-

lución óptima exacta, ya que se trata de variables continuas. Para poder aproximar la solución óptima, se ha optado por aplicar el método de la rejilla con un número de puntos elevado, de forma que obtendremos el vector $\boldsymbol{\theta}_{grid}$ y un número de vectores soporte Nsv_{grid}. Se considera que un método de búsqueda ha encontrado una solución óptima al problema si cumple las siguientes condiciones:

- La desviación del vector encontrado $\boldsymbol{\theta}_{Metodo}$ es menor que un $1\,\%$ en cada parámetro respecto a la encontrada en $\boldsymbol{\theta}_{grid}$.
- El error leave-one-out encontrado coincide o es menor que el error encontrado con el método de la rejilla.
- El número de vectores soporte es inferior o no supera más de un $2\,\%$ respecto al encontrado con el método de la rejilla.

De esta forma, se pueden realizar curvas de probabilidad de éxito contra el número de generaciones empleadas, como las mostradas en la figura (5.5). Para la realización de cada curva se comienza ejecutando el algoritmo permitiendo muy pocas evaluaciones, de forma que la probabilidad de encontrar una solución cercana a la óptima será muy baja. A medida que aumenta el número de evaluaciones permitidas, tanto por ampliar el número de generaciones como por ampliar el número de individuos por generación, la probabilidad de acierto irá creciendo, siendo esta subida más rápida conforme sea más eficiente el algoritmo. La figura (5.5) muestran curvas realizadas para diferentes clasificadores binarios que separan una clase del resto, cuando se optimiza un vector de hiperparámetros $\boldsymbol{\theta} = [C, \gamma]$. Para cada uno de estos clasificadores se ejecutó el método de la rejilla con las siguientes características:

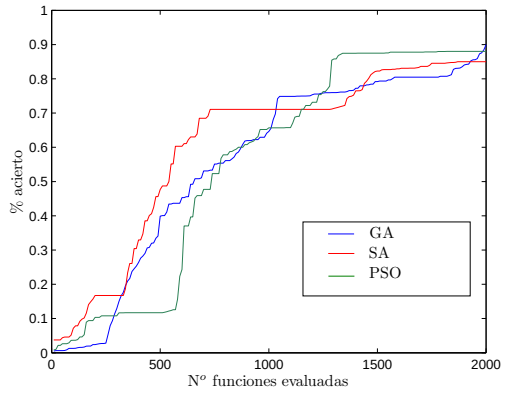
- En lugar de optimizar el parámetro C se trabaja con el logaritmo de dicha variable.
- El espacio de búsqueda se acota de la siguiente manera:

  Parámetro C . El espacio de búsqueda se acota entre $log\,(C) \in [0, 9]$.

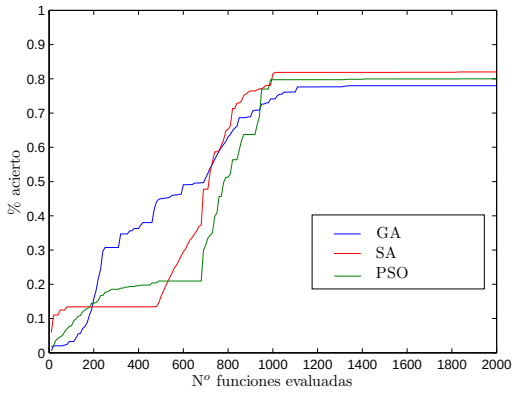
  Parámetro γ . El espacio de búsqueda queda acotado mediante $\gamma \in [\gamma_1, \gamma_2]$, calculando estos valores mediante

  $$\begin{aligned} \gamma_1 &= \tfrac{1}{d} \\ \gamma_2 &= \tfrac{50}{d} \end{aligned} \tag{5.51}$$

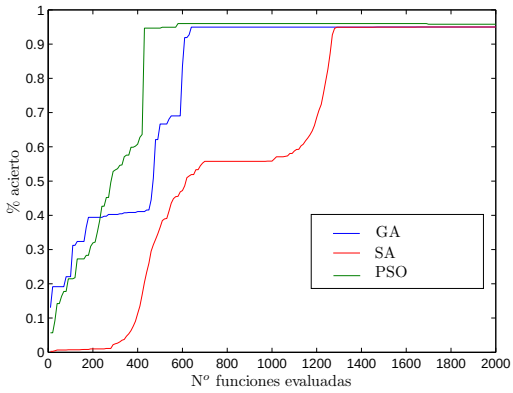
  donde d es la dimensión de los patrones de entrenamiento.
- En cada paso, se realiza la búsqueda sobre rejillas de tamaño 20×20 con un número de 4 iteraciones, lo que supone un total de 2000 evaluaciones para cada clasificador.



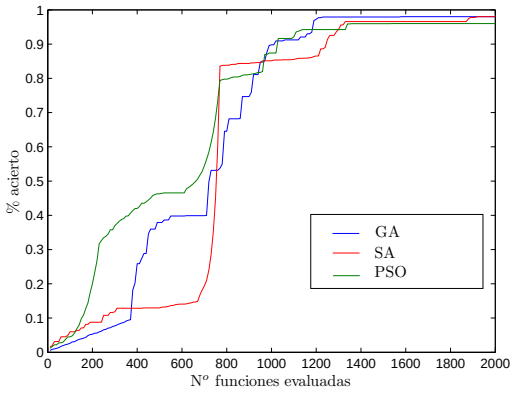
(a) *Optimización sobre conjunto de alcoholes.*



(b) *Optimización sobre conjunto CSIC.*



(c) *Optimización sobre conjunto FIA.*



(d) *Optimización sobre conjunto Ciclovoltiamperogramas.*

Figura 5.5: Curvas de porcentaje de éxito al encontrar hiperparámetros en función del número de funciones evaluadas.

Los algoritmos propuestos acotan su espacio de búsqueda de igual forma a como se acota con el algoritmo de la rejilla. Para el cálculo de la probabilidad en cada punto se ejecuta el algoritmo durante 50 repeticiones y se mide cuales de ellas obtuvieron éxito según las condiciones descritas anteriormente. En las curvas mostradas, se puede apreciar que no existe ningún método que sea claramente superior, aunque PSO destaca en algunos casos.

En la tabla (5.1) se muestran los resultados obtenidos, utilizando PSO como método de optimización de la función de error propuesta, sobre la aplicación de los diferentes conjuntos de datos con los que se trabajó en el capítulo anterior. Hay que indicar que para los conjuntos de separación de vinos por espectrofotometría UV-VIS no se produce ganancia alguna, ni en precisión ni en número de vectores soporte, porque se ha seleccionado un kernel lineal. Para el resto de conjuntos de datos vemos cómo

Dataset	Método Clásico		Optimización PSO	
	ACC	$\overline{Nsv}$	ACC	$\overline{Nsv}$
Alcoholes	0.985	288.83	0.992	180.12
Vinos Blancos	0.889	6	0.889	6
Vinos Tintos	0.917	8	0.917	8
FIA	0.954	79.38	0.971	63.12
Ciclovoltiamperogramas	0.968	62.4	0.985	57.24
CSIC	0.833	58.5	0.856	55.3

Tabla 5.1: Comparativa de resultados de aplicación entre el método clásico y la optimización por PSO propuesta.

mejora tanto la tasa de acierto como el número de vectores soporte medio necesario, tras realizar múltiples experimentos sobre cada conjunto de datos.

El ajuste de hiperparámetros con métodos meta-herurísticos constituye una aportación original de esta tesis, habiéndose publicado parte de las ideas en [Acevedo07b] y [Acevedo06].

5.2. Estrategias multiclase para las SVM

Uno de los aspectos que más critican los detractores del uso de las SVM es su concepción para resolver problemas binarios, quedando abierto cómo se solucionan los problemas multiclase. Existen varios trabajos ([Bredensteiner99], [Weston99] o [Crammer01]) que han intentado reformular el problema de optimización de las SVM para abordar su estudio partiendo de un problema multiclase. Sin embargo, aunque la solución propuesta en estos trabajos es matemáticamente atractiva, su implementación es compleja y no proporcionan tan buenos resultados como la división del problema mediante clasificadores binarios.

Hasta ahora, la estrategia seguida en esta tesis ha sido la denominada uno contra todos, consistente en dividir el problema en M problemas binarios. Para las futuras muestras de test, el método clásico asigna a la muestra $\boldsymbol{x}$ la clase $\mathcal{C}_i$ según:

$$\boldsymbol{x} \in \mathcal{C}_i \quad \max_i \; f_i(\boldsymbol{x}) \tag{5.52}$$

donde $f_i(\boldsymbol{x})$ es la salida del clasificador i-ésimo. De esta forma se asigna la clase a aquél clasificador que proporcione una mayor salida de las SVM. Sin embargo, el valor de la función de decisión de una SVM es un valor no calibrado sobre la pertenencia o no a una clase, por lo que comparar las salidas directamente de dichas SVM no es correcto

desde un punto de vista matemático. A lo largo de esta sección se verá cómo solventar este problema y otro tipo de estrategias para la resolución de problemas multi-clase.

5.2.1. Probabilidades de Platt

Para poder utilizar la estrategia uno contra todos, en [Platt99] se propone traducir la salida $f(\boldsymbol{x})$ de cada SVM a una probabilidad $P(\boldsymbol{x} \in \mathcal{C}_i / f(\boldsymbol{x}))$. Supongamos que tenemos un problema de clasificación como el mostrado en la figura (5.6), para el que se establece una frontera de decisión creada, sin pérdida de generalidad, con un kernel lineal. La tasa de acierto del propio problema, como muestra la figura, es de un 86,81 %.

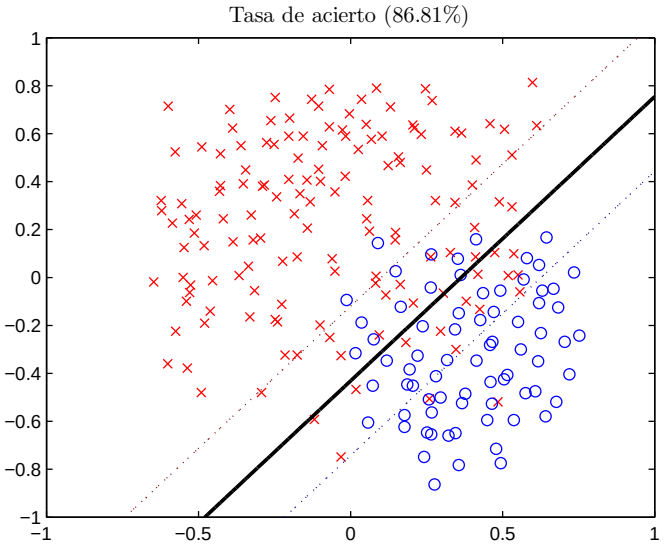
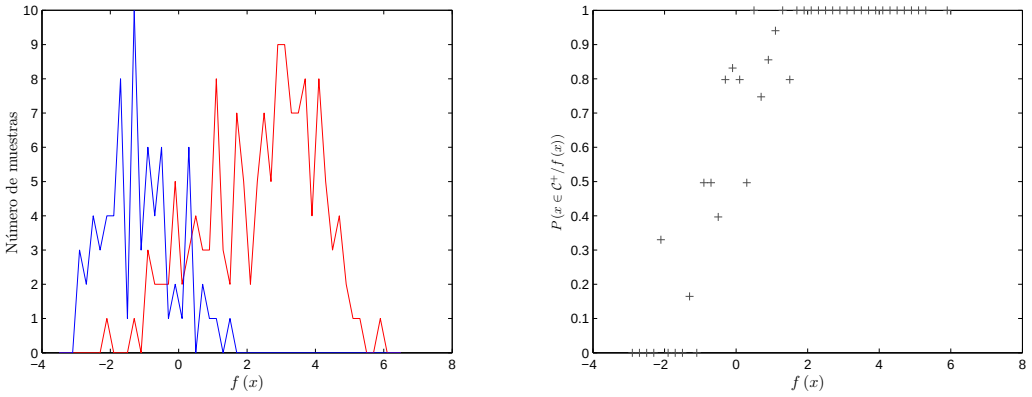


Figura 5.6: Problema de clasificación artificial y frontera creada por una SVM.

En la figura (5.7(a)) se han representado los histogramas estimados de la distribución de las muestras positivas y negativas según el valor de $f(\boldsymbol{x})$. Si se utiliza el histograma de muestras positivas para estimar la $P(f(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{C}_+)$, podemos aplicar el teorema de Bayes para estimar la probabilidad a posteriori:

$$P(\boldsymbol{x} \in \mathcal{C}_+ / f(\boldsymbol{x})) = \frac{P(f(\boldsymbol{x}) / \boldsymbol{x} \in \mathcal{C}_+) P(\boldsymbol{x} \in \mathcal{C}_+)}{P(f(\boldsymbol{x}) / \boldsymbol{x} \in \mathcal{C}_+) P(\boldsymbol{x} \in \mathcal{C}_+) + P(f(\boldsymbol{x}) / \boldsymbol{x} \in \mathcal{C}_-) P(\boldsymbol{x} \in \mathcal{C}_-)} \quad (5.53)$$

donde $P(\boldsymbol{x} \in \mathcal{C}_+)$ es la probabilidad de las muestras positivas, que puede ser estimada a partir del conjunto de entrenamiento. Los resultados de aplicar Bayes sobre el problema artificial considerado se pueden ver en la figura (5.7(b)), donde se puede apreciar que para valores muy negativos de $f(\boldsymbol{x})$ la probabilidad de que la muestra pertenezca a



(a) *histogramas de $f(\boldsymbol{x})$ del problema artificial.* **(b)** *Aplicación del teorema de Bayes a las muestras positivas.*

Figura 5.7: Estimación de la $P(\boldsymbol{x} \in \mathcal{C}_+ / f(\boldsymbol{x}))$ para el problema artificial.

la clase positiva es prácticamente nula, mientras que para valores muy positivos la probabilidad tiende a la unidad. A partir de esta gráfica se puede apreciar que la $P(\boldsymbol{x} \in \mathcal{C}_+ / f(\boldsymbol{x}))$ se puede aproximar por una función sigmoidal de la forma:

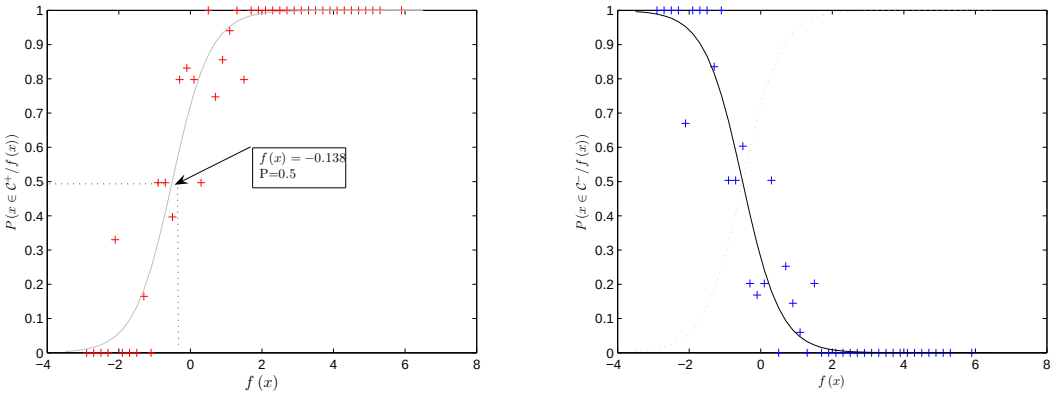
$$P(\boldsymbol{x} \in \mathcal{C}_+ / f(\boldsymbol{x})) \approx \frac{1}{1 + exp(-Af(\boldsymbol{x}) - B)} \tag{5.54}$$

donde los coeficientes A y B son parámetros que necesitan optimizarse para ajustar la sigmoide. Como resultado de esta optimización, se obtiene la gráfica representada en la figura (5.8(a)), en la que se muestra también el umbral para una $P(\boldsymbol{x} \in \mathcal{C}_+ / f(\boldsymbol{x})) = 0{,}5$ como nuevo umbral Thr, que puede ser calculado a partir de los parámetros de la sigmoide mediante:

$$Thr = \frac{-B}{A} \tag{5.55}$$

En la figura (5.8(b)) se ha realizado el cálculo considerando el proceso para las muestras negativas. Se ha superpuesto la sigmoide de las muestras positivas, dando como resultado la complementaria, de forma que ambas probabilidades suman la unidad para todos los valores de $f(\boldsymbol{x})$.

En la figura (5.9) se muestra el problema de clasificación con el nuevo umbral encontrado, consiguiendo una mejor tasa de acierto que con el umbral fijado a cero. Sin embargo, el objetivo no es tanto mejorar la tasa de acierto de cada clasificador binario, sino permitir la comparación entre diferentes SVM. De esta forma, en la estrategia uno contra todos se debe añadir para cada clasificador una etapa que convierta la salida del clasificador $f_i(\boldsymbol{x})$ en la salida de una sigmoide $\sigma_i(\boldsymbol{x})$ con coeficientes ajustados A_i



(a) *Ajuste de sigmoide para las muestras positivas.*

(b) *Ajuste de sigmoide para las muestras negativas.*

Figura 5.8: Ajuste de la $P(\boldsymbol{x} \in \mathcal{C}_{\pm} / f(\boldsymbol{x}))$ mediante sigmoides.

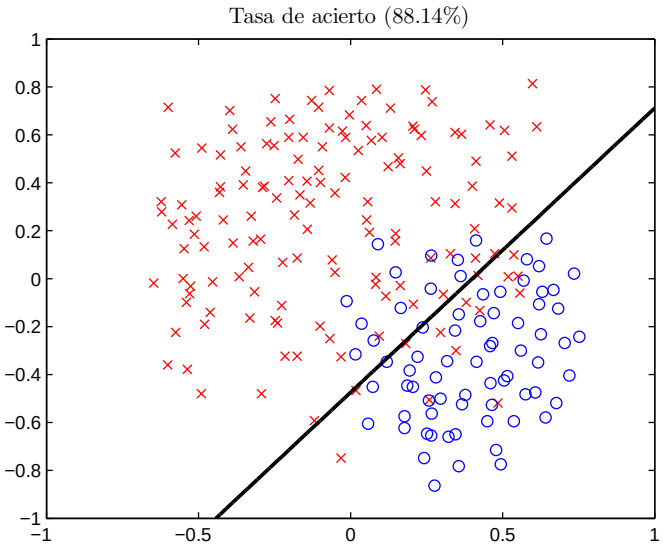


Figura 5.9: Problema de clasificación artificial con la nueva frontera.

y B_i. El ajuste de estos parámetros se realiza mediante la optimización del siguiente problema:

$$\min_{A,B} \; F(A,B) = -\sum_{j=1}^{l} t_j \log\left(\sigma\left(\boldsymbol{x}_j\right)\right) + \left(1-t_j\right)\left(1-\log\left(\sigma\left(\boldsymbol{x}_j\right)\right)\right)$$

$$t_j = \begin{cases} \frac{N_+ +1}{N_+ +2} \text{ si } y_j = 1 \\ \\ \frac{1}{N_- +2} \text{ si } y_j = -1 \end{cases} \tag{5.56}$$

donde N_+ y N_- son la cantidad de muestras positivas y negativas respectivamente. Para resolver la ecuación (5.56) se siguió la implementación descrita en [Lin07], que utiliza un método de optimización de Newton con secuencia de búsqueda *bactracking*. Para las futuras muestras de test, en lugar de utilizar el criterio descrito en (5.52), ahora se busca:

$$\boldsymbol{x} \in \mathcal{C}_i \;\; \max_i \; P\left(\boldsymbol{x} \in \mathcal{C}_i / f_i\left(\boldsymbol{x}\right)\right) = \frac{1}{1+exp\left(-A_i f_i\left(\boldsymbol{x}\right) - B_i\right)} \tag{5.57}$$

En la figura (5.10) se pueden apreciar las sigmoides encontradas para el conjunto de vinos tintos mediante espectrofotometría UV-VIS. En esta figura podemos apreciar, para un caso real, cómo cambia el criterio de decisión. Para una muestra determinada se obtienen los valores de las funciones de decisión correspondientes a las denominaciones de origen La Mancha y Valdepeñas $f_1\left(\boldsymbol{x}\right) = 1{,}53$ y $f_2\left(\boldsymbol{x}\right) = 0{,}97$ respectivamente, por lo que siguiendo el criterio clásico la muestra sería asignada a la denominación de origen La Mancha. Sin embargo, si observamos la figura (5.10) encontramos que la sigmoide pintada en color rojo corresponde al clasificador La Mancha contra los demás, mientras que la clase Valdepeñas tiene asignada la sigmoide dibujada en color azul. Así, los valores de salida de las sigmoides serán $\sigma_1\left(\boldsymbol{x}\right) = 0{,}81$ y $\sigma_1\left(\boldsymbol{x}\right) = 0{,}93$, por lo que finalmente es asignada la clase Valdepeñas.

En la tabla (5.2) se muestran los resultados encontrados utilizando el método clásico y el método de Platt para los conjuntos bajo estudio de esta tesis. Para realizar la tabla, se realizaron $L > 30$ experimentos de división de los conjuntos de entrenamiento y test, calculando adicionalmente las sigmoides para cada uno de los clasificadores binarios. En dicha tabla se muestra la media de la tasa de acierto para ambos métodos y los test de significancia estadística de Wilcoxon y t-test, exponiendo sus resultados como se vio en el capítulo anterior. Se marcan en negrita los resultados que plantean diferencias con significancia estadística. Para el entrenamiento se utilizó el método de ajuste de hiperparámetros descrito en la sección anterior, empleando en cada caso el kernel y sus parámetros asociados que mejor resultado dieron con búsqueda mediante PSO. Debe destacarse que el método expuesto requiere para el entrenamiento de la sigmoide muestras que resulten mal clasificadas con el método clásico. Hay que hacer notar que si el problema es claramente separable, los parámetros de la sigmoide tenderán a $A_i \to \infty$ y $B_i \to 0$. De esta premisa se puede entender que cuando todos los parámetros

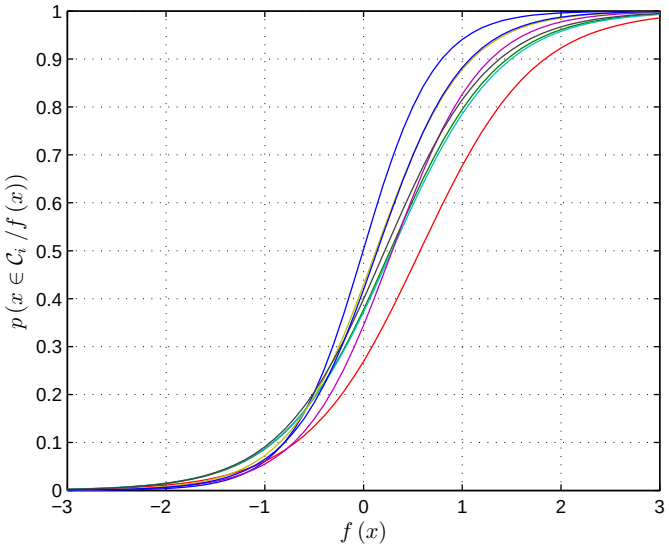


Figura 5.10: Sigmoides encontradas para el conjunto de clasificación de vinos tintos mediante espectrofotometría UV-VIS.

Dataset	Método Clásico	Método Platt	Test Wilcoxon	T-test
Alcoholes	0.992	0.991	0.35	60.32
Vinos Blancos	**0.917**	**0.931**	12.36	1.02
Vinos Tintos	**0.889**	**0.911**	7.23	2.04
FIA	0.971	0.974	1.35	35.4
Ciclovoltiamperogramas	0.985	0.978	1.86	8.12
CSIC	**0.856**	**0.888**	20.3	0.85

Tabla 5.2: Comparativa de resultados con la aplicación de las probabilidades de Platt.

de las sigmoides tienden a estos valores, el método de Platt convergería con el método clásico, ya que las sigmoides se convertirían en funciones escalón sin desplazamiento y será la ganadora la que proporcione un mayor valor de $f(\boldsymbol{x})$, motivo por el que este último suele dar buenos resultados aunque matemáticamente no sea correcto comparar directamente las salidas de las SVM.

Por último, es necesario comentar la reflexión que se hace en [Tipping01] sobre la capacidad del método propuesto para medir la probabilidad $P(\boldsymbol{x} \in \mathcal{C}_+ / f(\boldsymbol{x}))$. En el caso del trabajo citado se pone de manifiesto que, partiendo de dos distribuciones uniformes solapadas en una zona, el cálculo de la probabilidad encontrado con el método de Platt no puede ajustarse bien al cálculo de la probabilidad real. La respuesta a la crítica que hace Tipping es que se han utilizado funciones sigmoidales dependientes de

la salida de una SVM y ésta salida es creciente a medida que la muestra se aleja del plano de separación en el espacio de Hilbert inducido por la operación kernel. Por tanto, no puede acercarse al modelado de distribuciones uniformes, sino que debe cumplirse que la función de distribución de probabilidad de pertenencia a una clase sea creciente a medida que aumenta la separación al hiperplano. En cualquier caso, esta suposición es contraria al principio de no presuponer un modelo de distribución de los datos utilizado en las SVM. Por tanto, el método de Platt es un esfuerzo para realizar una comparación entre diferentes clasificadores, pero no podemos afirmar que el resultado proporcionado sea un reflejo exacto de la probabilidad de pertenencia a una clase.

La aplicación de la teoría de las probabilidades de Platt en el campo de la nariz electrónica constituye una aportación original de esta tesis y en concreto su uso para discriminar el conjunto de datos de alcoholes fue publicado en [Acevedo07a].

5.2.2. Estrategias de grafos ordenados

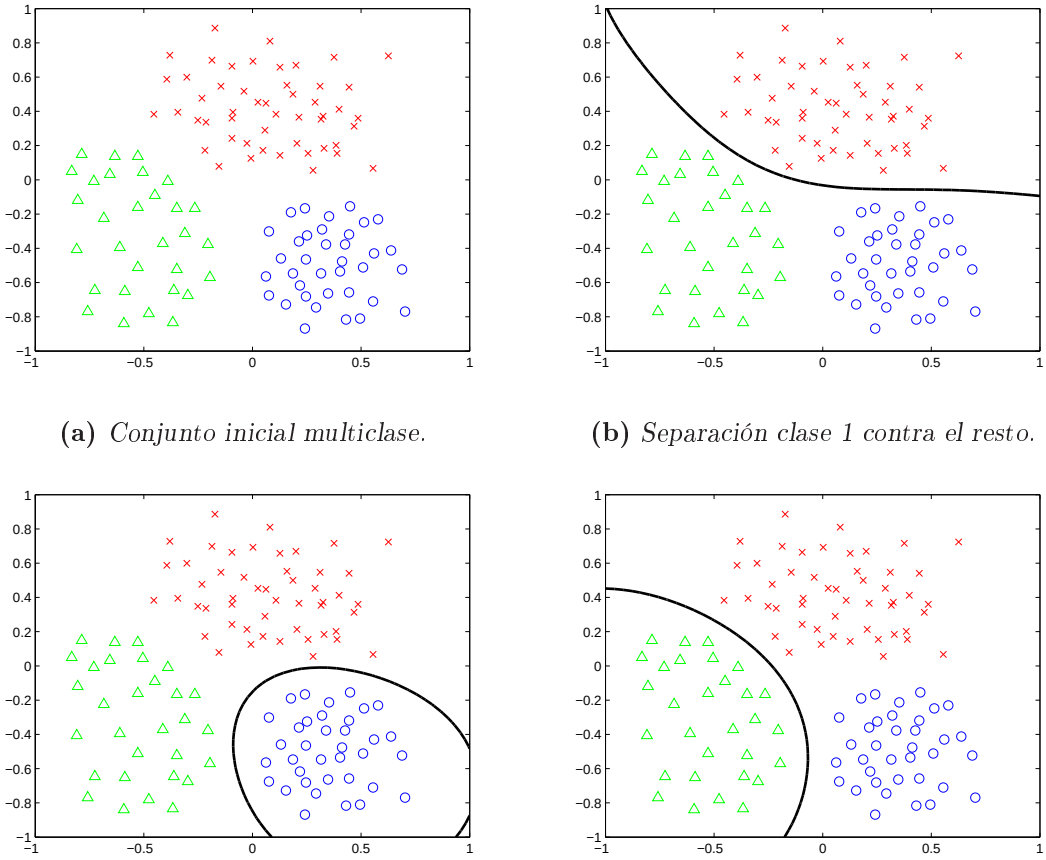
En la sección anterior se ha seguido una estrategia uno contra todos en la división de los problemas multiclase. Esta estrategia es la más común frente a otros tipos de estrategias como son la uno contra uno [Kreßel98]. Entre los motivos por los que esta estrategia no es comúnmente utilizada podríamos destacar los siguientes:

- Es necesario crear $M(M-1)/2$ clasificadores frente a los M necesarios en la estrategia uno contra todos. Esto hace que a priori la fase de test pueda requerir más operaciones para clasificar una muestra.

- Es complicado determinar a qué clase pertenece una muestra a partir de las salidas de clasificadores binarios. Aún aplicando el concepto de las probabilidades de Platt visto en la sección anterior, se ha establecido que éstas no reflejan fielmente la probabilidad de pertenencia a una clase.

Sin embargo, no se suele tener en cuenta que los clasificadores resultantes de una estrategia uno contra uno pueden ser más sencillos, y por tanto requerir menos vectores soporte que los obtenidos con una estrategia uno contra todos. En la figura (5.11) se muestra un problema de clasificación artificial de tres clases y las fronteras requeridas por cada uno de los tres clasificadores binarios siguiendo una estrategia uno contra todos. Como se puede observar del problema artificial, cada uno de estos clasificadores debe ser no lineal para responder al problema de separación de clases. Por contra, en la figura (5.12) se muestra el mismo problema de clasificación y las fronteras requeridas por cada uno de los clasificadores binarios usando una estrategia uno contra uno. Podemos ver cómo en este caso cada uno de los problemas de separación puede ser resuelto mediante clasificadores lineales. Este ejemplo hace que en la separación de los problemas

bajo estudio de esta tesis pueda resultar interesante seguir una estrategia uno contra uno. Por ejemplo, en el problema de clasificación de alcoholes se espera un clasificador más sencillo al separar entre las clases etanol y cualquier alcohol aromático como son el veratril, isoamil o el amil, frente a un clasificador más complejo al separar el etanol de un licor.



(a) *Conjunto inicial multiclase.*

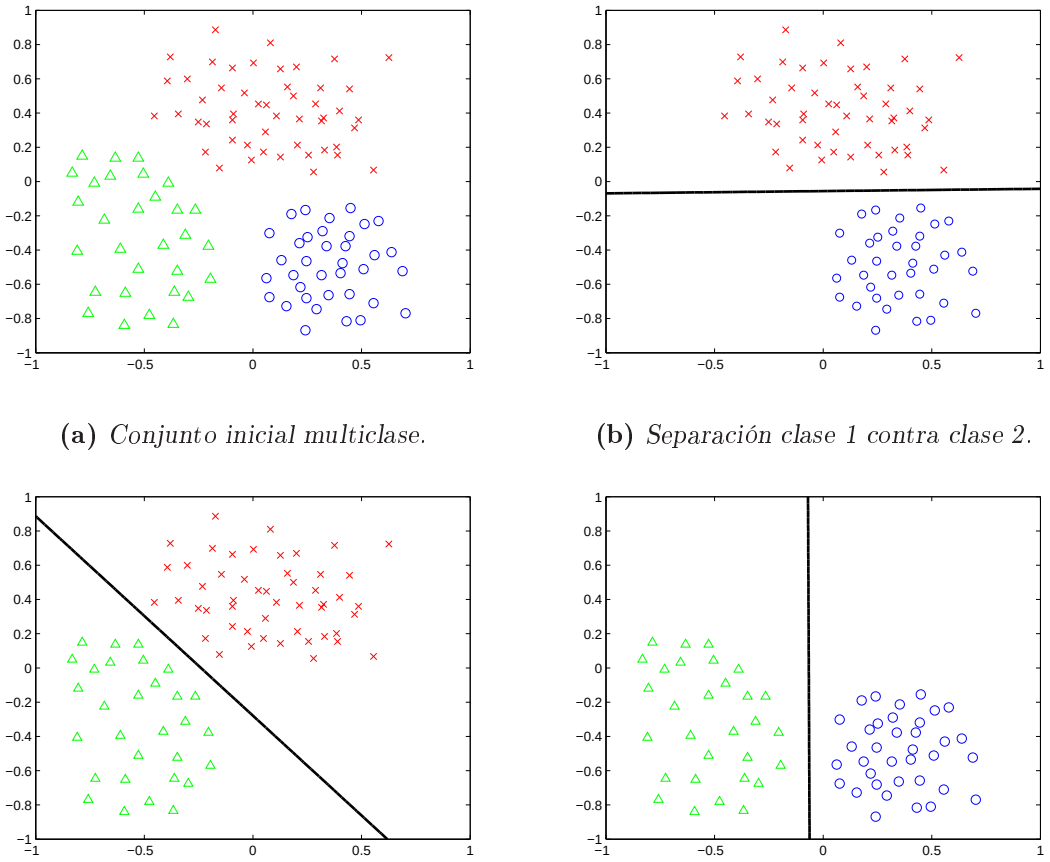
(b) *Separación clase 1 contra el resto.*

(c) *Separación clase 2 contra el resto.*

(d) *Separación clase 3 contra el resto.*

Figura 5.11: Problema multiclase y fronteras con estrategia uno contra todos.

En [Platt00] se propone el método denominado *Directed Acyclic Graphs* based on SVM (DAGSSVM) que consiste en construir los clasificadores binarios mediante la estrategia uno contra uno pero se forma un grafo de decisión, de forma que en la fase de test solo es necesario utilizar M clasificadores para asignar una clase. La idea de esta propuesta se puede ver en la figura (5.13) donde para una nueva muestra $\boldsymbol{x}$ se somete primero a un clasificador que compara las clases uno contra cuatro. Supuesto que, como resultado del primer clasificador, obtenemos que la muestra es más parecida a la clase uno que a la cuatro, se descarta la clase cuatro y se somete a una comparación entre



(a) *Conjunto inicial multiclase.*

(b) *Separación clase 1 contra clase 2.*

(c) *Separación clase 1 contra clase 3.*

(d) *Separación clase 2 contra clase 3.*

Figura 5.12: Problema multiclase y fronteras con estrategia uno contra uno.

las clases uno contra tres. Si se decide que se parece más a la clase tres, teniendo en cuenta que la clase cuatro ha sido descartada, solo queda la comparación con la clase dos para determinar la clase a la que pertenece.

Con el algoritmo DAGSVM se solventan los problemas de la estrategia uno contra uno pero los clasificadores binarios deben tener una elevada tasa de acierto individual, ya que cada equivocación lleva a una rama del grafo diferente, de forma que equivocará la clase a la que pertenece la muestra a clasificar. Para solventar este problema se propone en esta tesis utilizar los métodos de selección de kernel y ajuste de hiperparámetros propuestos, de forma que además de tener una mayor exactitud en cada uno de ellos, se minimiza el número de operaciones necesarias por cada clasificador. Realizando diversas pruebas sobre los conjuntos de datos empleados, se observó cómo el orden del grafo no influye apenas en la tasa de acierto total, pero sí tiene una gran influencia sobre el número de operaciones medio necesario para evaluar una muestra. Debe en-

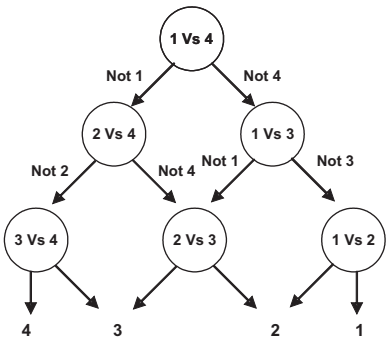


Figura 5.13: Grafo propuesto para un problema de 4 clases con el método DAGSVM.

tenderse que con este método dicho número de operaciones dependerá del camino que se recorra del grafo. Supongamos el grafo de la figura (5.13) y que el clasificador que más operaciones requiere es el de la clase uno contra la cuatro. Independientemente de la clase final asignada, con el grafo propuesto obligamos a que todas las muestras sean evaluadas con este clasificador.

La propuesta de esta tesis es crear un nuevo algoritmo de ordenación del grafo que minimice el número medio de operaciones necesarias para la fase de test. Para ello, en primer lugar se estiman las probabilidades de cada clase mediante:

$$\begin{aligned} P(\mathcal{C}_i) &= \sum_{h=1}^{l} \delta(y_h, i)/l \\ \delta(j,i) &= \begin{cases} 1 \text{ si } j = i, \\ 0 \text{ si } j \neq i \end{cases} \end{aligned} \tag{5.58}$$

Por otro lado, establecemos el número de vectores soporte N_{sv}^i como el número de vectores soporte suma de todos los clasificadores que evalúen la clase $\mathcal{C}_i$ contra otra. En aquellos casos en los que solo se requiere un clasificador lineal el número de vectores soporte a tener en cuenta será de uno, ya que el cálculo se puede realizar como un producto escalar con el plano creado, que a su vez puede ser calculado previamente.

Los pasos del algoritmo propuesto para formar el *Grafo Ordenado SVM (GOSVM)* son los siguientes:

- Realizar una lista L con la información de todas las clases incluidas. Cada elemento de L se calcula mediante:

$$L(i) = \frac{P(\mathcal{C}_i)}{N_{sv}^i} \tag{5.59}$$

- Ordenar la lista L de menor a mayor, de forma que el elemento $L(1)$ contiene la clase con una mayor probabilidad estimada y un menor número de operaciones para ser evaluada.

- Se construye el grafo tal como se muestra en la figura (5.14). El primer clasificador es el que discrimina entre las dos primeras clases de la lista ordenada L. La siguiente capa se compone de dos clasificadores que comparan las clases anteriores de la lista con la nueva clase contenida en $L(i)$. De esta forma se van a añadiendo capas al grafo hasta que se hayan sido incluidos todos los niveles.

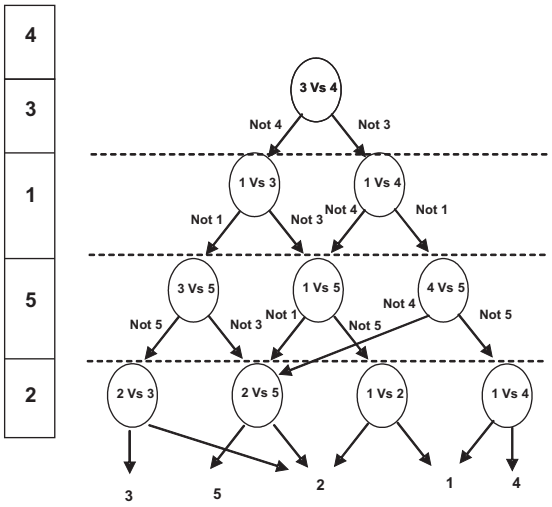


Figura 5.14: Grafo propuesto para un problema de 4 clases con el método GOGSVM.

En la tabla (5.3) se muestra una comparación de tasa de acierto entre el método DAGSVM y el método GOSVM. Como se ha hecho en la presentación de anteriores resultados se han incluido los resultados de los test estadísticos para determinar la significancia de los resultados. Se observa cómo no existen diferencias significativas en cuanto a la tasa de acierto entre ambos métodos y son muy similares a las obtenidas con el método uno contra todos. En la tabla (5.4) se muestra la diferencia de vectores soporte medios necesarios para evaluar una nueva muestra entre el método DAGSVM y GOSVM, donde el método DAGSVM se ha descompuesto en dos columnas: Por un lado se presentan los resultados del método original, mientras que en el segundo caso se presentan las operaciones necesarias con los clasificadores binarios optimizados. Hay que destacar que no existe ganancia en la aplicación del método en los conjuntos de separación de vinos por espectrofotometría UV-VIS, ya que para todos los clasificadores binarios se emplea un kernel lineal.

La propuesta del método GOSVM es una de las aportaciones originales de esta tesis y fue publicada en [Acevedo07c].

Dataset	ACC. DAGSVM	ACC. GOSVM	Test Wilcoxon	T-test
Alcoholes	0.987	0.988	0.35	30.26
Vinos Blancos	0.927	0.925	0.17	46.5
Vinos Tintos	0.913	0.914	0.46	26.3
FIA	0.973	0.973	0.31	38.9
Ciclovoltiamperogramas	0.971	0.970	0.95	12.3
CSIC	0.875	0.875	0.16	86.4

Tabla 5.3: Comparativa de resultados de aplicación entre el método DAGSVM y GOSVM.

Dataset	Número vectores soporte medio		
	DAGSVM	DAGSVM Opt	GOSVM
Alcoholes	250.12	190.42	174.12
Vinos Blancos	6	6	6
Vinos Tintos	8	8	8
FIA	78.12	68.4	64.12
Ciclovoltiamperogramas	63.4	56.1	54.3
CSIC	56.2	53.8	52.1

Tabla 5.4: Comparativa de vectores soporte medios necesarios en los métodos DAGSVM y GOSVM.

5.3. Agrupación de vectores soporte

Uno de los objetivos de este capítulo es encontrar formas de reducir la complejidad computacional para evaluar nuevas muestras cuando se utilizan kernels no lineales. En el ajuste de hiperparámetros se ha incorporado a la función de optimización la capacidad de seleccionar aquellos parámetros que, además de optimizar la tasa de acierto, reducen el número de vectores soporte. Sin embargo, es necesario profundizar en la reducción del número de vectores soporte necesarios para minimizar los requerimientos de memoria y ganar en velocidad de cálculo para la fase de test. Recordemos que los vectores soporte se pueden dividir en dos clases, aquellos que caen en los límites de las fronteras de decisión y que cumplen $(0 < \alpha_i < C)$ o aquellas muestras del conjunto de entrenamiento que caen dentro de la zona del margen o son candidatos a estar mal clasificados $(\alpha_i = C)$. Se puede apreciar a partir de los conjuntos de datos con los que se trabaja en esta tesis, cómo es altamente probable que los vectores soporte no aparezcan como muestras aisladas, sino como grupos de muestras con una reducida distancia euclídea entre ellos. La propuesta de esta parte de la tesis es concentrar esos grupos

de vectores soporte que son de la misma clase y están suficientemente cerca en un único vector soporte. El algoritmo se aplica sobre cada clasificador binario construido. Así, los pasos del algoritmo a aplicar una vez se ha entrenado el clasificador, son los siguientes:

1. Crear dos matrices con las distancias entre los vectores soporte positivos y negativos respectivamente.

$$\begin{array}{l} D^{+}\left(SV_i, SV_j\right), i \neq j, SV_{i,j}/y_i, y_j = +1 \\ D^{-}\left(SV_i, SV_j\right), i \neq j, SV_{i,j}/y_i, y_j = -1 \\ D^{+}\left(SV_i, SV_i\right) = D^{-}\left(SV_i, SV_i\right) = \beta \end{array} \tag{5.60}$$

2. Buscar la distancia mínima de los vectores positivos y verificar si $D^{+}_{min}(i,j) < \beta$ donde β es un umbral dado a priori. Si esta condición no se cumple, el agrupamiento de vectores soporte ha concluido. Por otro lado se comprueba si $D^{-}_{min}(i,j) < \beta$ y si no se cumple también se concluye con el agrupamiento de vectores soporte negativos.

3. Si la condición del punto anterior se ha cumplido, se elimina uno de los vectores soporte (por ejemplo SV_i). La razón por la cuál se elimina uno de los vectores soporte y no se crea un pseudo-vector es permitir que el mismo vector soporte pueda estar presente en varios clasificadores, ahorrando memoria.

4. Se reconstruye la matriz $\boldsymbol{D}$ en la que se ha realizado el agrupamiento, o ambas matrices si hubo agrupamiento tanto en los vectores positivos como los negativos. La nueva matriz se calcula $\boldsymbol{D}^{\setminus i}$, esto es, la matriz $\boldsymbol{D}$ sin la fila y columna i.

5. Se repite el proceso desde el paso 2 hasta que no sea posible realizar ningún agrupamiento.

6. Se reentrena la SVM con los vectores que no han sido descartados.

Se debe destacar que el agrupamiento de vectores soporte solo tiene sentido en aquellos clasificadores en los que se ha seleccionado un kernel RBF, ya que para kernels lineales el plano de decisión se puede calcular como una combinación lineal de los vectores soporte y por tanto, reducir su número no implica reducción alguna en las operaciones de test. También es importante destacar que realizamos la búsqueda de vectores soporte que estén próximos en cuanto a su distancia euclídea. Podría pensarse que es más correcto buscar la relación de proximidad en el espacio de Hilbert transformado para el que estamos calculando el producto interno mediante el truco del kernel. Pero si tenemos en cuenta que podemos calcular la distancia entre dos vectores en dicho espacio transformado mediante [Shawe-Taylor04]:

$$\| \phi(SV_i) - \phi(SV_j) \|^2 = \kappa(SV_i, SV_i) + \kappa(SV_j, SV_j) - 2\kappa(SV_i, SV_j) \tag{5.61}$$

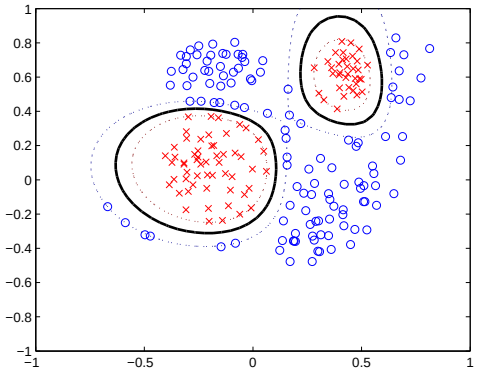
Cuando se usa un kernel RBF la ecuación anterior se convierte en:

$$\| \phi(SV_i) - \phi(SV_j) \|^2 = 2 - 2exp\left(-\gamma \| (SV_i) - (SV_j) \|^2\right) \tag{5.62}$$

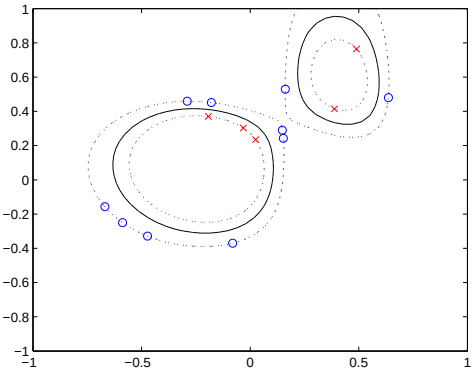
y por tanto, buscar el mínimo $\| \phi(SV_i) - \phi(SV_j) \|^2$ implica buscar la distancia euclídea mínima en el espacio de entrada para dicho kernel.

En la figura (5.15) se muestra un ejemplo del agrupamiento de vectores soporte por distancia para un conjunto de datos artificial de dos dimensiones. En la figura (5.15(a)) se presenta el conjunto de datos inicial con la frontera creadas por una SVM en el entrenamiento de dicho conjunto. En la figura (5.15(b)) se pueden apreciar con detalle los vectores soporte del conjunto de datos original. Como puede verse, existen varios vectores soporte próximos entre sí. En la figura (5.15(c)) se muestra la agrupación de vectores soporte cuando se establece el umbral de distancia a un valor $\beta = 0.08$, pasando de tener 15 vectores soporte a tener solo 10. Se reentrena una SVM únicamente con los 10 vectores que han quedado y se obtienen las fronteras que también aparecen en dicha figura. Para ver con mayor claridad el efecto de este agrupamiento, en la figura (5.15(d)) se vuelve a mostrar el conjunto de datos original con la nueva frontera y la antiguas frontera dibujada en gris. Vemos cómo se superponen ambas fronteras, por lo que se ha conseguido reducir el número de vectores soporte un 33 % sin cambiar la exactitud del clasificador. Si aumentamos el umbral de distancia hasta $\beta = 0.2$, como se muestra en la figura (5.15(e)), conseguimos bajar el número de vectores soporte hasta tener únicamente 8. Sin embargo, en la figura (5.15(f)) se muestra la nueva frontera representada en negro y la frontera original representada en gris. En este caso se puede apreciar cómo cambia la exactitud del clasificador quedando alguna de las muestras mal clasificadas. Este último caso demuestra que la aplicación de este método puede requerir de un compromiso entre la tasa de reducción de vectores soporte y la pérdida de tasa de acierto del clasificador. Aunque el ejemplo se ha mostrado para un clasificador binario, el método se aplica en el caso multiclase sobre cada uno de los clasificadores en los que ha sido dividido el problema.

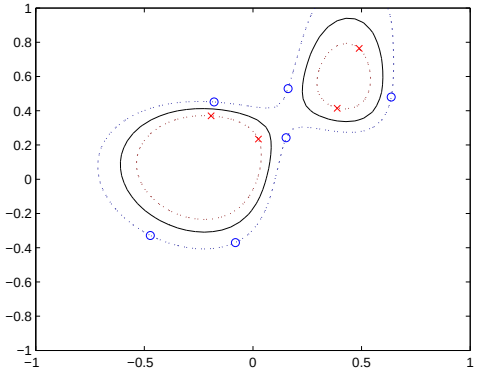
Para visualizar los resultados del método propuesto se realizaron barridos del umbral de distancia tolerado β, de forma que se pueda apreciar cómo se comporta la tasa de acierto a medida que va creciendo dicho umbral y por tanto, se va reduciendo el número de vectores soporte. En la figura (5.16) se muestra el resultado de aplicar dicho barrido al conjunto de alcoholes. Una vez ha sido entrenado el modelo, se tiene el punto inicial de la gráfica con una tasa de acierto de 0,981 y un número de vectores soporte igual a $Nsv = 406$. Se puede observar cómo tomando esos vectores soporte y agrupando aquellos que se sitúan más cerca se consigue una sensible reducción del número de



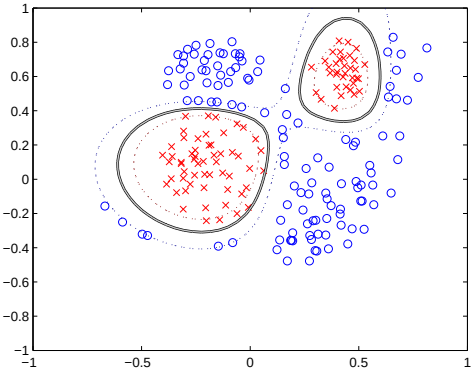
(a) *Conjunto inicial con frontera SVM*



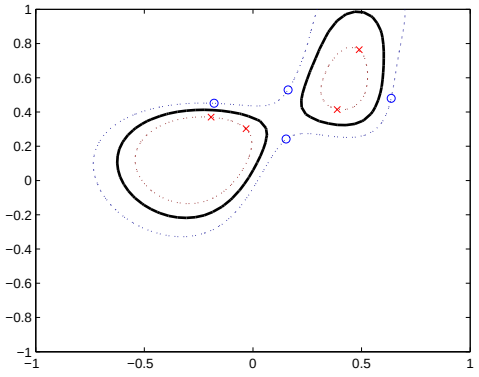
(b) *Vectores soporte del conjunto original.*



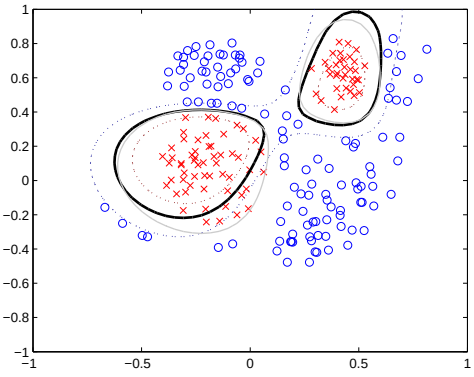
(c) *Agrupación para* $\beta = 0{,}08$.



(d) *Conjunto inicial con nuevas frontera* ($\beta = 0{,}08$)



(e) *Agrupación para* $\beta = 2$.



(f) *Conjunto inicial con nuevas frontera* ($\beta = 2$)

Figura 5.15: Ejemplo de la aplicación de la reducción de vectores soporte por distancia.

vectores soporte necesarios. A medida que se va incrementando el umbral β se puede apreciar en dicha gráfica cómo se produce una disminución en términos generales de la tasa de acierto. Sin embargo, existe una zona donde la tasa de acierto es similar a la conseguida con el modelo completo y sin embargo el número de vectores soporte se reduce notablemente. Se ha identificado un segundo punto en esta gráfica donde se puede apreciar cómo la tasa de acierto es muy similar a la original y sin embargo el número de vectores soporte necesarios solo es la tercera parte del total.

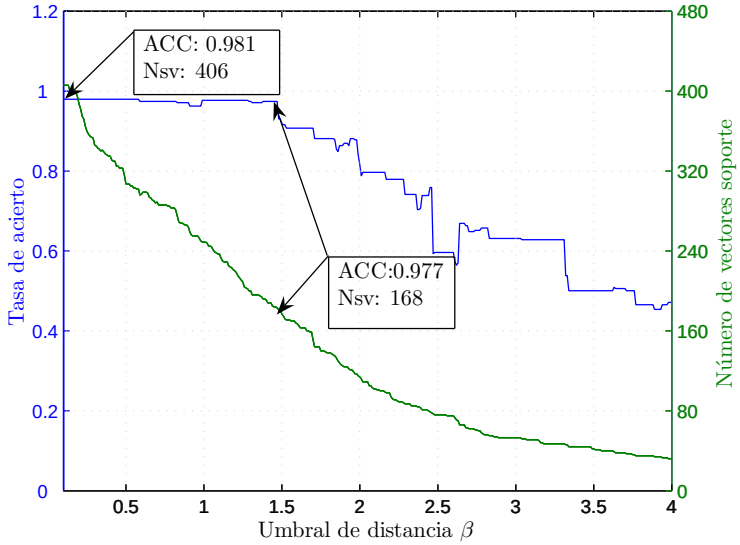
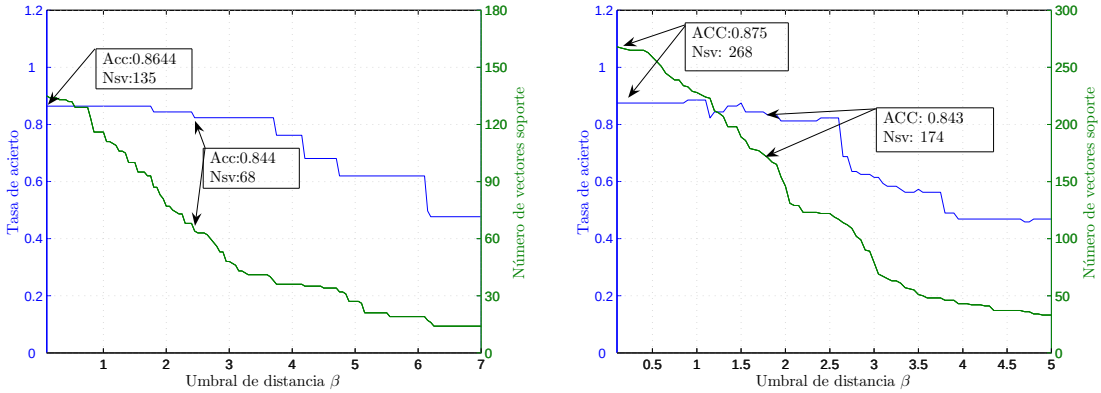


Figura 5.16: Comparativa de tasa de acierto y número de vectores soporte aplicando agrupamiento para el conjunto de alcoholes.

En la figura (5.17) se muestra el resultado de aplicar el algoritmo a los conjuntos de separación de vinos por su denominación de origen a partir de los espectrofotogramas UV-VIS. Al igual que en la sección anterior, se han incluido estos conjuntos de datos utilizando un kernel RBF, aunque como se vio en el capítulo anterior se obtiene mejor resultado con métodos lineales. Sin embargo, como se ha explicado anteriormente, el método aquí propuesto no tiene aplicación para un kernel lineal. Al igual que para el caso de la clasificación de alcoholes se ha marcado el punto inicial de cada barrido, que correspondería con la tasa de acierto y el número de vectores soporte sin aplicar el agrupamiento de los mismos. En el caso de los vinos blancos se logra reducir casi a la mitad el número de vectores soporte con una tasa de acierto similar. En el caso de los vinos tintos la reducción en el número de vectores soporte no es tan sensible pero el método sigue demostrando su utilidad.

En las figuras (5.18) y (5.19) se puede apreciar el método aplicado a los conjuntos de clasificación de vinos mediante análisis de flujo por inyección y ciclovoltiamperome-



(a) *Conjunto de vinos blancos.* **(b)** *Conjunto de vinos tintos.*

Figura 5.17: Comparativa de tasa de acierto y número de vectores soporte aplicando agrupamiento para el conjunto de clasificación de espectrofotogramas UV-VIS de vinos.

trías respectivamente. En ambos casos, se mantiene una tasa de acierto muy similar a la original con una reducción de aproximadamente la mitad de vectores soporte necesarios. Si embargo, en la figura (5.20) se muestra la reducción del conjunto de datos procedente del CSIC. En este caso, se puede apreciar cómo la reducción de vectores soporte apenas tiene efecto si se intentan mantener tasas de acierto similares. El motivo de este comportamiento es el reducido volumen de este conjunto, que además está formado por muestras no similares entre sí. Esta falta de buenos resultados para este conjunto nos reafirma en que el buen comportamiento de los otros casos se debe a que están formados por muestras tomadas en intervalo de tiempo consecutivo y por tanto son muy similares entre sí. Esto hace, que al igual que se mostraba en el problema artificial de la figura (5.15), los vectores soporte aparecen en grupos de muestras. Sin embargo, si la realización de ensayos se hace en intervalos de tiempo separados, cambiando las condiciones ambientales, y se toma únicamente una muestra no aparecerán probablemente este tipo de agrupamientos. Sin embargo, esta situación no es habitual en el entrenamiento de sistemas de nariz y lengua electrónica, por lo que el método propuesto puede ser un instrumento eficaz para reducir el número de vectores soporte necesarios y así, reducir los requerimientos de memoria y aumentar la velocidad de cálculo en la etapa de test.

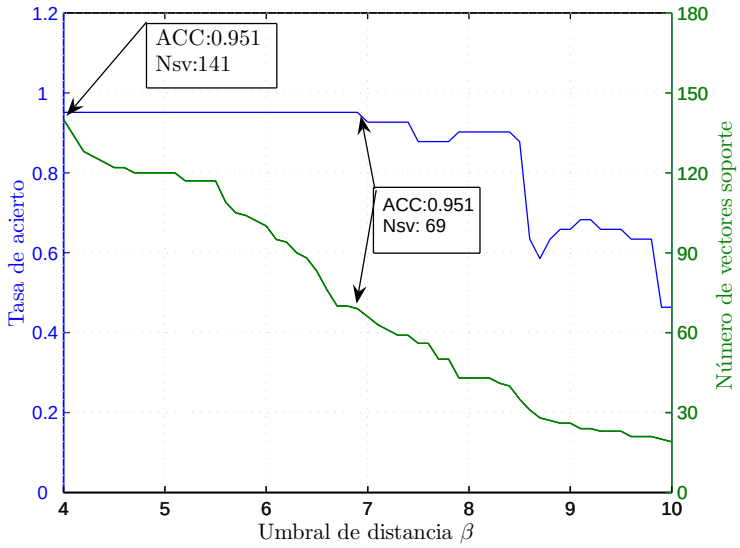


Figura 5.18: Comparativa de tasa de acierto y número de vectores soporte aplicando agrupamiento para el conjunto FIA.

5.4. Reducción de operaciones en la función de decisión

Uno de los aspectos en los que se ha incidido más a lo largo del capítulo es la necesidad de reducir al máximo posible el número de operaciones en la fase de test. Hasta ahora, esta idea ha sido abordada desde la perspectiva de reducir el número de vectores soporte, bien mediante la optimización de parámetros o bien mediante la agrupación de dichos vectores soporte.

En esta sección la idea es reducir el número de operaciones una vez que se ha optimizado el número de vectores soporte. Para conseguir este objetivo se aprovecharán las relaciones geométricas de los kernels basados en la distancia euclídea, como es caso del kernel RBF utilizado en esta tesis. A medida que se va calculando la función de decisión, utilizando este sentido geométrico, vamos a encontrar cotas de la misma, deteniendo su cálculo si no existe posibilidad de cambiar de signo. Esto hace que podamos ahorrar operaciones de test manteniendo la misma tasa de acierto. Aunque el método se aplica a las SVM, puede ser también aplicado a las RVM o las LS-SVM.

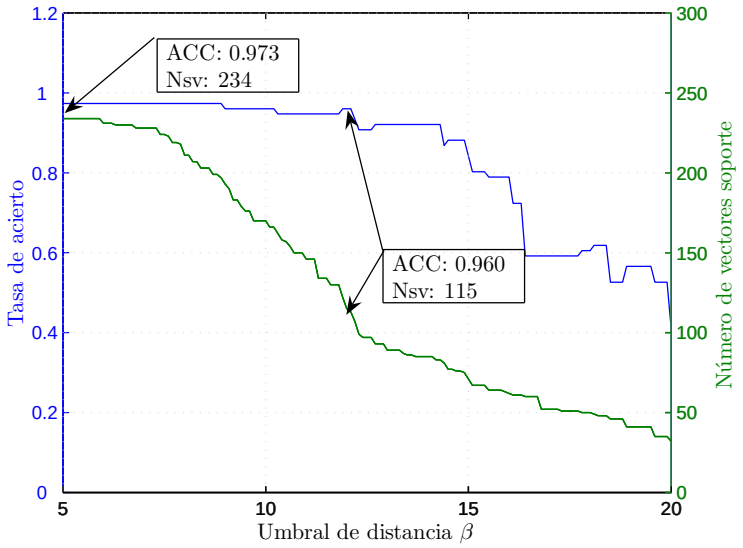


Figura 5.19: Comparativa de tasa de acierto y número de vectores soporte aplicando agrupamiento para el conjunto de ciclovoltiamperometría.

5.4.1. Kernel RBF

Como se expuso en la revisión bibliográfica, una vez entrenadas las SVM proporcionan una función de decisión de la forma:

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{Nsv} \alpha_i y_i \kappa(\boldsymbol{x}, \boldsymbol{sv}_i) + b\right) \tag{5.63}$$

El kernel no lineal utilizado en esta tesis es el de base radial, por sus buenos resultados en aquellas aplicaciones que requieren clasificadores no lineales:

$$\kappa(\boldsymbol{x}, \boldsymbol{sv}_i) = exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{sv}_i\|^2}{2\sigma^2}\right) \tag{5.64}$$

Vamos a seleccionar un vector soporte cualquiera que denominaremos $\boldsymbol{sv}_1$ y formamos una lista que contenga al resto de los vectores soporte ordenados de menor a mayor distancia del vector seleccionado $\boldsymbol{sv}_1$:

$$L = \{sv_1, sv_2, ...sv_{Nsv}\}\, d(\boldsymbol{sv}_q, \boldsymbol{sv}_1) \leq d(\boldsymbol{sv}_{q+1}, \boldsymbol{sv}_1)\ \ q = 2, ..., Nsv \tag{5.65}$$

Consideremos ahora una nueva muestra $\boldsymbol{x}$ que debe ser clasificada. La distancia de dicha muestra al primer vector soporte de la lista creada será $d(\boldsymbol{x}, \boldsymbol{sv}_1)$. Utilizando la desigualdad de Cauchy-Schwarz la siguiente condición siempre se cumple:

$$d(\boldsymbol{sv}_1, \boldsymbol{sv}_q) \leq d(\boldsymbol{sv}_1, \boldsymbol{x}) + d(\boldsymbol{x}, \boldsymbol{sv}_q) \tag{5.66}$$

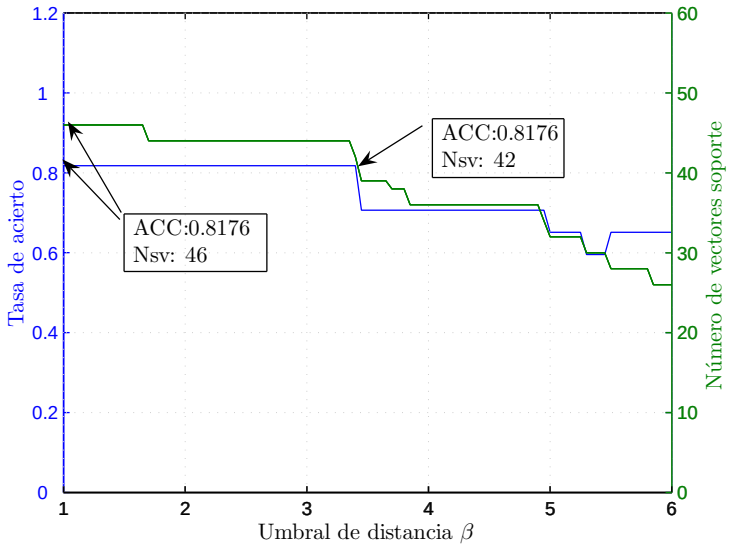


Figura 5.20: Comparativa de tasa de acierto y número de vectores soporte aplicando agrupamiento para el conjunto de separación de gases procedente del CSIC.

Así, es posible obtener una cota inferior de la distancia $d(\boldsymbol{x}, \boldsymbol{sv}_q)$ que existe entre la nueva muestra y un vector soporte cualquiera $\boldsymbol{sv}_q$. Dicha cota viene establecida por:

$$d\left(\boldsymbol{x}, \boldsymbol{sv}_q\right) \geq d\left(\boldsymbol{sv}_1, \boldsymbol{sv}_q\right) - d\left(\boldsymbol{sv}_1, \boldsymbol{x}\right) \tag{5.67}$$

Además, dado que los vectores están ordenados de forma ascendente en una lista y se cumple $d(\boldsymbol{sv}_q, \boldsymbol{sv}_1) \leq d(\boldsymbol{sv}_{q+1}, \boldsymbol{sv}_1)$, la ecuación (5.67) supone una cota inferior para todos los vectores que se sitúen en la lista después de q:

$$d_n^{low}\left(\boldsymbol{x}, \boldsymbol{sv}_n\right) \geq d\left(\boldsymbol{sv}_1, \boldsymbol{sv}_q\right) - d\left(\boldsymbol{sv}_1, \boldsymbol{x}\right) \;\; \forall n \geq q \tag{5.68}$$

Por otro lado, utilizando de nuevo la desigualdad de Cauchy-Schwarz la distancia máxima desde el vector de entrada $\boldsymbol{x}$ a cualquier vector soporte de la lista satisface:

$$d_n^{upp}\left(\boldsymbol{x}, \boldsymbol{sv}_n\right) \leq d\left(\boldsymbol{sv}_1, \boldsymbol{sv}_{Nsv}\right) + d\left(\boldsymbol{sv}_1, \boldsymbol{x}\right) \;\; \forall n \leq Nsv \tag{5.69}$$

lo que supone una cota superior de la distancia entre el vector de entrada y cualquier vector soporte. Ahora utilizaremos las cotas encontradas para evitar el cálculo de operaciones innecesarias en la fase de test. Primero se reescribe la ecuación (5.63) de forma que queda:

$$
\begin{aligned}
f(\boldsymbol{x}) &= f_n(\boldsymbol{x}) + f_n^+(\boldsymbol{x}) - f_n^-(\boldsymbol{x}) \\
f_n^+(\boldsymbol{x}) &= \sum_{i=1}^{Nsv_n^+} \alpha_i^{n+} exp(-\|\boldsymbol{x} - \boldsymbol{sv}_i^{n+}\|^2/2\sigma^2) \\
f_n^-(\boldsymbol{x}) &= \sum_{i=1}^{Nsv_n^-} \alpha_i^{n-} exp(-\|\boldsymbol{x} - \boldsymbol{sv}_i^{n-}\|^2/2\sigma^2)
\end{aligned}
\tag{5.70}
$$

donde $f_n(\boldsymbol{x})$ es la función de decisión evaluada hasta el vector n de la lista creada, incluyendo en dicha evaluación el parámetro de bias b, $f_n^+(\boldsymbol{x})$ es la contribución restante de todos los vectores soporte positivos, siendo sv^{n+} dicho conjunto de vectores soporte positivos cuya clase está determinada mediante $y_i = 1$ y con multiplicadores α_i^{n+} asociados al conjunto de vectores soporte positivos restantes. Por otro lado $f_n^-(\boldsymbol{x})$ representa la contribución restante del conjunto de vectores soporte negativos. Definimos ahora una cota B para el conjunto de vectores restantes de forma que se cumpla:

$$
\|f_n(\boldsymbol{x})\| > B > \|f_n^-(\boldsymbol{x}) - f_n^+(\boldsymbol{x})\| \tag{5.71}
$$

Si existe esta cota B entonces no hay posibilidad de que cambie el signo que tenga la función evaluada hasta ahora $f_n(\boldsymbol{x})$ y por tanto la muestra a clasificar $\boldsymbol{x}$ no requiere de más operaciones kernel. Si consideramos, como se ha expuesto, que los vectores soporte están agrupados en una lista, teniendo en cuenta las cotas establecidas en las Ecuaciones (5.68) y (5.69) y considerando que se cumplen las siguientes condiciones:

$$
\begin{aligned}
&f_n(\boldsymbol{x}) \geq 0 \\
&f_n(\boldsymbol{x}) \geq exp\left(-\tfrac{d_n^{low^2}}{2\sigma^2}\right) \sum_{i=1}^{Nsv^{n-}} \alpha_i^{n-} - exp\left(-\tfrac{d_n^{upp^2}}{2\sigma^2}\right) \sum_{i=1}^{Nsv^{n+}} \alpha_i^{n+}
\end{aligned}
\tag{5.72}
$$

la muestra $\boldsymbol{x}$ solo puede ser clasificada como positiva, por lo que no es necesario seguir evaluando la función de test con el consiguiente ahorro de $Nsv - n$ evaluaciones del kernel. Por otro lado, si las condiciones que se dan son:

$$
\begin{aligned}
&f_n(\boldsymbol{x}) \leq 0 \\
&f_n(\boldsymbol{x}) \leq -exp\left(-\tfrac{d_n^{low^2}}{2\sigma^2}\right) \sum_{i=1}^{Nsv^{n+}} \alpha_i^{n+} + exp\left(-\tfrac{d_n^{upp^2}}{2\sigma^2}\right) \sum_{i=1}^{Nsv^{n-}} \alpha_i^{n-}
\end{aligned}
\tag{5.73}
$$

la muestra $\boldsymbol{x}$ solo puede ser clasificada como negativa, con el consiguiente ahorro de operaciones de evaluación. Hay que hacer notar que los sumatorios de la segunda parte de las Ecuaciones (5.72) y (5.73) pueden ser precalculados y almacenados en memoria en la fase de entrenamiento. De esta forma, las operaciones necesarias para comprobar las mencionadas condiciones son una suma y dos exponenciales. Estas operaciones extra necesarias, no suponen coste computacional comparadas con la evaluación de un kernel.

El algoritmo propuesto se divide en dos partes. En la primera etapa, después de realizar el entrenamiento del clasificador, se ejecutan los siguientes pasos:

1. Encontrar K centroides sobre el conjunto de vectores soporte. Estos centroides pueden ser calculados mediante el algoritmo k-means.

2. Seleccionar los K vectores soporte más cercanos a los centroides encontrados en el paso anterior. Por cada vector soporte seleccionado se crea una lista con el resto de vectores soporte ordenados de menor a mayor distancia.

3. Además de contener el índice al vector soporte, cada elemento de la lista K_i debe guardar la distancia al primer vector soporte, la suma de todos los multiplicadores positivos restantes α_i^{n+} y la suma de todos los multiplicadores negativos α_i^{n-}.

En la fase de test, se ejecutan los siguientes pasos:

1. Seleccionar una lista L_h de forma que:

$$h = \min_h d\left(\boldsymbol{x}, \boldsymbol{sv}_1^h\right) \tag{5.74}$$

donde $d\left(\boldsymbol{x}, \boldsymbol{sv}_1^h\right)$ es la distancia desde la muestra hasta el primer vector soporte de la lista h.

2. Calcular $f_0\left(x\right) = b$ y asignar $n = 1$.

3. Para cada vector de la lista $f_n\left(x\right)$ se actualiza verificando si alguna de las condiciones descritas en las Ecuaciones (5.72) y (5.73) se verifica. Si dichas condiciones no se dan, se incrementa el valor de n y se sigue evaluando hasta que se evalúen todos los vectores soporte.

5.4.2. Kernel exponencial

Las cotas mostradas en las Ecuaciones (5.68) and (5.69) nos permiten calcular las contribuciones máxima y mínima de las funciones restantes $f_n^-\left(\boldsymbol{x}\right)$ y $f_n^+\left(\boldsymbol{x}\right)$. Sin embargo, las Ecuaciones (5.72) y (5.73) comparten los mismas cotas de distancia para todos los vectores soporte restantes. En esta parte se buscarán límites más estrechos para las contribuciones restantes de las funciones que quedan por evaluar.

En lugar del kernel descrito en la ecuación (5.64), consideremos el kernel exponencial definido por:

$$\kappa\left(\boldsymbol{x}, \boldsymbol{sv}_i\right) = exp\left(-\frac{\left\|\boldsymbol{x} - \boldsymbol{sv}_i\right\|}{2\sigma^2}\right) = exp\left(-\gamma d\left(\boldsymbol{x}, \boldsymbol{sv}_i\right)\right) \tag{5.75}$$

Por otro lado, cada vector soporte tendrá las siguientes cotas al vector a evaluar $\boldsymbol{x}$:

$$\begin{aligned} d\left(\boldsymbol{x}, \boldsymbol{sv}_i\right) &\leq d\left(\boldsymbol{sv}_1, \boldsymbol{sv}_i\right) + d\left(\boldsymbol{x}, \boldsymbol{sv}_1\right) \\ d\left(\boldsymbol{x}, \boldsymbol{sv}_i\right) &\geq d\left(\boldsymbol{sv}_1, \boldsymbol{sv}_i\right) - d\left(\boldsymbol{x}, \boldsymbol{sv}_1\right) \end{aligned} \tag{5.76}$$

y por tanto, la contribución de cada vector soporte estará acotada por:

$$\begin{aligned}\alpha_i exp\left(-\gamma d\left(\boldsymbol{sv}_i, \boldsymbol{x}\right)\right) &\geq \alpha_i exp\left(-\gamma d\left(\boldsymbol{sv}_1, \boldsymbol{sv}_i\right)\right) exp\left(-\gamma d\left(\boldsymbol{sv}_1, \boldsymbol{x}\right)\right) \\ \alpha_i exp\left(-\gamma d\left(\boldsymbol{sv}_i, \boldsymbol{x}\right)\right) &\leq \alpha_i exp\left(-\gamma d\left(\boldsymbol{sv}_1, \boldsymbol{sv}_i\right)\right) exp\left(+\gamma d\left(\boldsymbol{sv}_1, \boldsymbol{x}\right)\right)\end{aligned} \tag{5.77}$$

Así, las condiciones de parada con el kernel exponencial pueden ser reformuladas mediante:

$$\begin{aligned}&f_n\left(\boldsymbol{x}\right) \geq 0 \\ &f_n\left(\boldsymbol{x}\right) \geq exp\left(+\gamma d\left(\boldsymbol{x}, \boldsymbol{sv}_1\right)\right) \sum_{i=1}^{Nsv^{n-}} \alpha_i^{n-} exp\left(-\gamma d\left(\boldsymbol{sv_i}, \boldsymbol{sv}_1\right)\right) - \\ &\qquad exp\left(-\gamma d\left(\boldsymbol{x}, \boldsymbol{sv}_1\right)\right) \sum_{i=1}^{Nsv^{n+}} \alpha_i^{n+} exp\left(-\gamma d\left(\boldsymbol{sv_i}, \boldsymbol{sv}_1\right)\right)\end{aligned} \tag{5.78}$$

y

$$\begin{aligned}&f_n\left(\boldsymbol{x}\right) \leq 0 \\ &f_n\left(\boldsymbol{x}\right) \leq -exp\left(+\gamma d\left(\boldsymbol{x}, \boldsymbol{sv}_1\right)\right) \sum_{i=1}^{Nsv^{n+}} \alpha_i^{n+} exp\left(-\gamma d\left(\boldsymbol{sv_i}, \boldsymbol{sv}_1\right)\right) + \\ &\qquad exp\left(-\gamma d\left(\boldsymbol{x}, \boldsymbol{sv}_1\right)\right) \sum_{i=1}^{Nsv^{n-}} \alpha_i^{n-} exp\left(-\gamma d\left(\boldsymbol{sv_i}, \boldsymbol{sv}_1\right)\right)\end{aligned} \tag{5.79}$$

De nuevo, los sumatorios incluidos en las condiciones de parada pueden ser precalculados en la fase de entrenamiento y la distancia al primer vector soporte se calcula para seleccionar la lista a utilizar. La diferencia entre el kernel exponencial y el kernel RBF es que en este último caso aparece un término exponencial adicional debido al cuadrado de la distancia y por tanto, no es posible precalcular los sumatorios de la expresión (5.79). En la figura (5.21) se muestra una comparación de las cotas obtenidas con el kernel exponencial y las obtenidas en el apartado anterior. En dicha figura se ha representado la diferencia normalizada de la función restante de decisión real, que representa la línea del cero, y las cotas calculadas para ambos casos. Se puede observar cómo las nuevas cotas se sitúan mucho más próximas a la línea de cero, indicando que se aproximan mucho más al valor de la función restante de decisión real.

5.4.3. Resultados

En esta parte se exponen los resultados de ahorro de operaciones, caracterizadas mediante el ahorro de evaluaciones de funciones kernel. A medida que la dimensión del problema de clasificación sea mayor, cada evaluación kernel representará un mayor número de operaciones en sumas y restas, por lo que su ahorro será más importante. Para establecer dichos resultados, se realizaron diversas divisiones de cada conjunto de datos para formar los conjuntos de entrenamiento y test. En la tabla (5.4.3) se muestran los resultados aplicando el kernel RBF, reflejando el número de evaluaciones

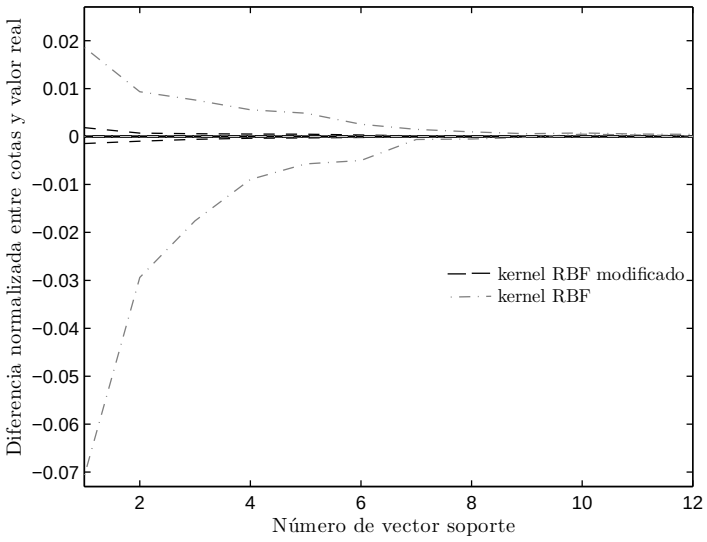


Figura 5.21: Comparación normalizada entre las cotas.

medio necesario para evaluar una muestra con el método clásico, en el que se evalúan todos los vectores soporte, y con el método propuesto para dicho kernel, deteniendo la evaluación si con las cotas calculadas no existe posibilidad de cambiar de signo. Es importante destacar que, aunque se han incluido los conjuntos de separación de vinos por denominación de origen procedentes de espectrofotometría UV-VIS, éstos alcanzaban mejores resultados con kernels lineales. Sin embargo, el método propuesto solo es de aplicación con kernels RBF. A partir de esta última columna se puede visualizar una comparación entre los métodos. Si bien es cierto que el método basado en el kernel RBF modificado consigue en la mayor parte de los casos mejores resultados, el hecho que empleando este kernel se produzcan más vectores soporte hace que en global los ahorros no sean tan significativos en la mayoría de los conjuntos, llegando a ser necesarias más operaciones medias en los casos en los que el ahorro ha sido marcado como negativo.

En la tabla (5.4.3) se reflejan los resultados utilizando el kernel exponencial y sus cotas asociadas. El procedimiento seguido ha sido el mismo que en el kernel RBF, es decir, se realizan diversos experimentos dividiendo el conjunto de datos en entrenamiento y test. Además, se siguió un proceso de ajuste de los parámetros mediante el procedimiento de validación cruzada para este kernel. Si comparamos el número medio de evaluaciones completas necesarias con este kernel con el número medio de evaluaciones completas necesarias con el kernel RBF, nos damos cuenta que el kernel exponencial, por lo general, requiere de más vectores soporte para una tasa de acierto similar. Sin embargo, este aumento en el número de vectores soporte y por tanto en el número de

Conjunto de datos	Número de características	Evaluaciones kernel		
		Evaluación completa	Método propuesto	Ahorro (%)
Alcoholes	71	1881.45	323.12	82.82
Vinos Blancos UV-VIS	61	54.16	24.23	55.26
Vinos Tintos UV-VIS	61	124.27	34.22	72.46
CV	1276	157.65	39.24	75.1
FIA	204	117.25	72.13	38.48
CSIC	124	38.26	16.23	57.58

Tabla 5.5: Resultados de ahorro de evaluaciones kernel utilizando el kernel RBF

Conjunto de datos	Evaluaciones kernel			
	Evaluación completa	Método propuesto	Ahorro (%)	Ahorro RBF (%)
Alcoholes	2688	113.73	95.77	64.8
Vinos Blancos UV-VIS	92.45	19.83	78.54	18.16
Vinos Tintos UV-VIS	184.27	35.45	80.76	-3.59
CV	232	36.9	84.09	5.96
FIA	108.41	69.82	35.6	3.2
CSIC	40.12	18.15	54.76	-11.83

Tabla 5.6: Resultados de ahorro de evaluaciones kernel utilizando el kernel exponencial

evaluaciones necesarias, se ve compensado por el manejo de cotas propuestas para este kernel, que como se mostró en la figura (5.21) son mucho más próximas a la función de decisión real. Se ha añadido una columna adicional de ahorro respecto a las operaciones obtenidas utilizando las cotas del kernel RBF. En aquellos casos en los que se muestra un porcentaje negativo, significa que es mejor utilizar el kernel RBF.

La propuesta de este método, tanto utilizando el kernel exponencial como el gaussiano, es una aportación original de esta tesis y esta parte del trabajo ha sido publicada en [Acevedo09].

5.5. Resumen de las aportaciones realizadas en el capítulo

Es importante destacar que la mayoría de las aportaciones realizadas en este capítulo, además de ser de aplicación en el marco de esta tesis, son de aplicación en otros

campos de investigación donde se requiera el uso de SVM como método de aprendizaje. Se exponen a continuación las principales aportaciones

Propuesta de cálculo rápido del error leave-one-out en SVM Se propone un nuevo método para evaluar el error leave-one-out, relacionado con el error de generalización de las SVM, sin necesidad de utilizar cotas de estimación. El método parte de las condiciones dadas por estos estimadores para solo tener que evaluar unas pocas muestras del conjunto de entrenamiento a las que se aplica las técnicas de inicialización por semilla y reducción para facilitar su cálculo. Además se incorpora el número de vectores soporte al criterio a minimizar de forma que se atienda esencialmente a este parámetro cuando el error obtenido es suficientemente bajo y en caso contrario se optimizará por mínimo error primero y ante la igualdad de éste se seleccionarán los hiperparámetros que minimicen los vectores soporte.

Búsqueda de hiperparámetros por algoritmos metaheurísticos Dado que la función propuesta no es derivable, se propone el uso de los siguientes algoritmos para optimizar los parámetros de una SVM:

- Algoritmos genéticos (GA) de optimización continua.
- Simulated annealing (SA) con múltiples soluciones por temperatura.
- Adaptación del algoritmo de optimización por colonia de hormigas (Ant Colony).
- Particle swarm optimization (PSO).

Aplicación de probabilidades a la estrategia uno contra todos La forma habitual de comparar las salidas de una SVM para problemas multiclase no es correcta desde un punto de vista matemático. Mediante este método se pretende estimar la probabilidad de pertenencia a una clase dado un valor de la función de decisión, optimizando los parámetros de una sigmoide que será colocada como salida de cada uno de los clasificadores binarios. Las salidas de estas sigmoides sí pueden ser comparadas matemáticamente. En este caso, el método se encontraba descrito de una forma general y la aportación se encuadra en la aplicación del método a los sistemas de nariz y lengua electrónica.

Propuesta de grafos ordenados Además de la estrategia uno contra todos, la estrategia uno contra uno puede proporcionar clasificadores más sencillos. Si se elabora un grafo de decisión solo es necesario comprobar un número de clasificadores igual al número de clases. La propuesta en este sentido es el algoritmo de ordenación del grafo para minimizar el número de operaciones medio de la fase de test.

Agrupación de vectores soporte El algoritmo propuesto tiene su hipótesis en que cuando aparece un vector soporte, es muy probable que encontremos una muestra cercana debido a la toma de muestras como resultado de un ensayo efectuado en las mismas condiciones. Esto hace que nos planteemos el reentrenamiento de las SVM agrupando los vectores soporte de la misma clase que se encuentren cercanos.

Reducción de operaciones en la función de decisión. Esta propuesta se aplica sobre kernels basados en la distancia euclídea, como son algunos de los utilizados en esta tesis. El método establece en cada momento las cotas de las contribuciones restantes positivas y negativas y detiene el cálculo de la función de decisión cuando no existe posibilidad de cambiar el signo de dicha función.

Capítulo 6

Conclusiones y futuras líneas de investigación

Este capítulo recoge las principales conclusiones de esta tesis haciendo un recorrido por los objetivos propuestos para la misma, hace un resumen de las aportaciones originales realizadas y señala las posibles líneas de investigación futuras que surgen a partir del trabajo realizado.

6.1. Conclusiones

Las redes de excelencia sobre los sistemas de nariz y lengua electrónica establecen tres líneas de actuación para la mejora de dichos sistemas. Esta tesis se centra en las técnicas de preprocesado y reconocimiento de patrones, en especial cuando se trata de información dinámica. En la revisión bibliográfica se estudian las principales técnicas que se aplican sobre los sensores para poder entender mejor la naturaleza de las señales con las que se va a trabajar a lo largo de la tesis.

Respecto a la extracción de información de las señales dinámicas, a partir de la revisión bibliográfica se ha podido dilucidar que los métodos de extracción encontrados son válidos para un determinado tipo de señales muy concretas, como es el caso de Padé-Z o la aproximación por exponenciales, o funcionan de forma muy irregular como el caso del espacio de fase o los modelos ARMA. El uso de la transformada wavelet ha sido ampliamente utilizado con éxito en diversas aplicaciones de la nariz y lengua electrónicas, pero solo se han considerado wavelets ortogonales. Se ha propuesto en esta tesis la ampliación del uso de la transformada wavelet con extensión periódica simétrica, para lo que se requieren waveletes biortogonales y también se propone el uso de esquemas de descomposición wavelet packet. Tanto la descomposición como la selección de qué coeficientes wavelet deben formar parte de los patrones a clasificar debe realizarse de forma común, para lo que se propone un método que resuelve este

problema. Además de la extensión de la transformada wavelet se propone un método para parametrizar las señales con un número fijo de señales base, por lo que se ha utilizado un método de regresión kernel conocido como *kernel fixed regression* (KFR). Este método se ha extendido para poder utilizar funciones kernel de tipo spline y se ha establecido cómo seleccionar las funciones base necesarias. Estas técnicas han sido aplicadas a los conjuntos de señales bajo estudio de esta tesis indicando que, cuando el número de coeficientes seleccionado es bajo, el método KFR proporciona buenos resultados aplicando el tipo de kernel más adecuado, que para algunos casos es el gaussiano y para otros es el kernel de tipo spline. Sin embargo, cuando el número de coeficientes es elevado, este método de extracción de la información confunde al método de clasificación y no proporciona buenos resultados. Por el contrario, la transformada wavelet demuestra su mejor comportamiento cuando el número de coeficientes es más elevado y se demuestra que para determinado tipo de señales como son las procedentes de espectrofotometría o ciclovoltiamperogramas la reflexión periódica simétrica, que implica el uso de wavelets biortogonales, suele proporcionar mejores resultados que el uso de wavelets ortogonales.

El uso de máquinas de vectores relevantes (RVM) y de máquinas de vectores soporte por mínimos cuadrados (LS-SVM) como métodos de clasificación que no habían sido probados en los sistemas de nariz y lengua electrónica. En el primer caso la propuesta viene motivada por proporcionar soluciones de menor densidad que en el caso de las SVM o las LS-SVM, ya que su entrenamiento parte de conceptos completamente diferentes al principio de maximización del margen y utilización del denominado truco del kernel. Las LS-SVM se ha propuesto por los buenos resultados mostrados en otras áreas de aplicación en cuanto a tasa de acierto. Además, las LS-SVM suelen ser utilizadas por la velocidad de su entrenamiento. En este sentido se ha propuesto un método de aprendizaje incremental, desarrollado en el marco de esta tesis, para evitar tener que recalcular todo el proceso de entrenamiento ante un nuevo ensayo. La comparación de resultados aplicados sobre los conjuntos de datos utilizados en esta tesis avala las hipótesis planteadas sobre el uso de estos clasificadores, proporcionando normalmente mejores resultados las LS-SVM pero con un número muy elevado de operaciones necesarias para la fase de test, mientras que las RVM reducen dicho número de operaciones a costa de tener normalmente una menor tasa de acierto.

Respecto a la comparación de clasificadores se ha establecido una metodología de comparación adecuada a los conjuntos de datos que manejamos y se ha ampliado también al número de operaciones necesarias para evaluar una futura muestra. Para realizar esta comparación se implementaron todos los métodos de clasificación en lenguaje MATLAB por lo que se han descrito los detalles de implementación utilizados. Una de las conclusiones más importantes de esta comparación es la dependencia en la tasa de acierto del correcto ajuste de los parámetros de la mayoría de los clasificado-

res no lineales. Los resultados de dicha comparación muestran cómo existen conjuntos de datos donde la mejor solución es aplicar métodos no lineales, ya que los obtenidos con métodos lineales no son admisibles. Sin embargo, existen otros conjuntos de datos donde la mejor opción es utilizar clasificadores lineales respecto a su tasa de acierto. Un último grupo de conjuntos de datos indica que se obtienen ligeramente mejores resultados de la aplicación de métodos no lineales pero los métodos lineales requieren varios órdenes menos de magnitud en las operaciones necesarias para evaluar futuras muestras. Así, podemos ver que los métodos kernel de clasificación son adecuados para los sistemas de nariz y lengua electrónica, ya que proporcionan la flexibilidad suficiente para trabajar con ellos de forma lineal o no lineal, dependiendo del kernel seleccionado. Como conclusiones adicionales se destaca el buen comportamiento en general que tiene el método random forest propuesto recientemente para este tipo de sistemas y cómo para algunos conjuntos de datos los resultados proporcionados por métodos como SIMCA o PLS-DA son absolutamente insatisfactorios. Esta conclusión es relevante en cuanto a que estos métodos se encuentran muy extendidos en los sistemas de lengua electrónica.

Se ha desarrollado un nuevo método que trabaja con bloques de características y que proporciona tan buenos resultados en cuanto a tasa de acierto que los algoritmos en el estado del arte aplicados a los sistemas de nariz y lengua electrónica con un número muy inferior de evaluaciones necesarias. Sin embargo, la gran utilidad de este método consiste en proporcionar la selección por bloques en lugar de sobre muestras aisladas lo que hace posible la relación entre las magnitudes físicas (temperatura, tensión o longitud de onda) con la explicación de la discriminación del problema.

En el último capítulo de la tesis se han abordado diversas técnicas de mejora sobre las SVM. Todas ellas vienen derivadas de los resultados obtenidos en la comparación de métodos de clasificación, en la que se vio la necesidad de ajustar correctamente los parámetros del kernel y la constante de regularización, así como reducir el número de operaciones necesarias para evaluar futuras muestras. En primer lugar se ha desarrollado un método para el cálculo eficiente del error leave-one-out y se establece una función de optimización que tiene presente el número de vectores soporte resultado. Para buscar los valores de los hiperparámetros se ha propuesto una serie de algoritmos metahurísticos que optimizan el proceso de búsqueda como son los algoritmos genéticos, simulated annealing y particle swarm optimization.

En este capítulo de mejoras sobre las SVM, se ha profundizado en la aplicación de este método de entrenamiento a los problemas multiclase. En primer lugar se destaca que la comparación de diferentes salidas, como se realiza en los trabajos publicados de aplicación de las SVM al campo de la nariz y lengua electrónica, no es correcta desde un punto de vista matemático. Para evitar este problema se añade una etapa que transforma la salida en una sigmoide para poder comparar probabilidades a posteriori.

Los resultados no varían sensiblemente por la aplicación de este método, pero se da explicación matemática en que los valores encontrados para las sigmoides se aproximan a la comparación del valor más elevado, siendo más correcto desde un punto de vista matemático. Adicionalmente, se ha utilizado la teoria de grafos para construir un clasificador SVM basado en clasificadores uno contra uno. En este caso la propuesta es el método de construcción de dicho grafo para conseguir minimizar el número medio de operaciones necesarias para evaluar una nueva muestra. Los resultados obtenidos indican que no se obtienen variaciones sensibles de la tasa de acierto pero sí se consigue minimizar el número de vectores soporte.

Las últimas dos ideas sobre la mejora de las máquinas de vectores soporte tienen como objetivo reducir el número de operaciones necesarias para evaluar una nueva muestra, una vez que el clasificador está entrenando. La primera de ellas se basa en la hipótesis de que algunas de las muestras definidas como vectores soporte se encuentran muy próximas entre sí, por lo que se puede conseguir una reducción en el número de vectores soporte si agrupamos ambas muestras. Los resultados obtenidos muestran que dicha reducción tiene lugar manteniendo la tasa de acierto. La segunda idea comprueba en todo momento la cota máxima restante de la parte positiva y negativa de la función de evaluación con el objeto de no seguir realizando evaluaciones de kernel cuando no existe posibilidad de cambiar de signo. Para dicha idea se utilizó un kernel gaussiano y una modificación del mismo. Los resultados muestran cómo el número de evaluaciones de kernel queda reducido con ambos métodos sensiblemente.

Se cumplen de esta forma todos los objetivos planteados para la realización de esta tesis, con excepción de la propuesta de nuevas líneas de investigación que se realizará más adelante.

6.2. Resumen de aportaciones

Las aportaciones de la tesis se pueden resumir en los siguientes puntos:

Ampliación de la transformada wavelet. Esta aportación aplica métodos conocidos en el campo de la compresión a los sistemas de nariz y lengua electrónica, de forma que se aplica la extensión periódica simétrica y se ha propuesto un método para seleccionar globalmente la descomposición wavelet packet y los coeficientes wavelet. Se demuestra que en aquellas señales como las espectrofotogramas UV-VIS o las ciclovoltiamperometrías, la propuesta mejora significativamente los resultados utilizando las wavelet tradicionales.

Propuesta de kernel fixed regression. Esta propuesta para extraer información de las señales dinámicas se basa en una técnica utilizada en otros campos como método de regresión multivariante. Como aportación de esta tesis se propone

utilizar esta técnica como método de parametrización de las señales obtenidas de los sensores y utilizar los parámetros obtenidos como entradas del clasificador. Además, se ha ampliado la técnica Kernel Fixed Regression para utilizar kernels de tipo spline, con los que se consiguen mejores resultados en algunos casos. El método demuestra ser eficaz frente a la transformada wavelet o PCA cuando consideramos un número pequeño de coeficientes.

Aplicación de clasificadores con éxito en otros campos. Se proponen la aplicación de clasificadores como son las LS-SVM y las RVM que son métodos kernel en el estado del arte y no han sido probados sobre los sistemas de nariz y lengua electrónicas. La propuesta de estos clasificadores viene dada en el primer caso por su facilidad de entrenamiento y resultados proporcionados, mientras que en el segundo aplicamos una filosofía de entrenamiento diferente que hace que el número de vectores considerado para la función de decisión sea mucho menor que en las SVM.

Comparación entre los métodos de clasificación. Se ha propuesto una metodología de comparación entre los métodos de clasificación adaptada a los sistemas de nariz y lengua electrónica. Además, se da validez a los resultados medios mediante la aplicación de test de significancia estadística como son el t-test y el test de Wilcoxon. Esta metodología es absolutamente necesaria a la hora de obtener conclusiones sobre si un método de clasificación es superior a otro. Además de la comparación en tasas de acierto, se realiza un estudio de las operaciones necesarias para evaluar futuras muestras. Para poder llevar a cabo dicha comparación fue necesaria la implementación de todos los clasificadores descritos en lenguaje MATLAB. Como resultado de esta parte del trabajo, se determina que no siempre los clasificadores más complejos son los que mejor funcionan. La hipótesis de partida inicial se ve así confirmada indicando que la flexibilidad de los métodos kernel es adecuada para adaptarse a los problemas planteados en los conjuntos de datos recopilados.

Aprendizaje incremental en las LS-SVM. El aprendizaje incremental supone una herramienta útil cuando se están adquiriendo muestras a partir de ensayos y no se quiere reentrenar todo el conjunto pasado. Para tal fin, en esta tesis se aporta un método de aprendizaje incremental basado en las propiedades de la descomposición Cholesky de la matriz de Gram transformada.

Propuesta de un método de selección de características Se ha propuesto un nuevo método de selección de características por bloques que proporciona, en cuanto a tasa de acierto, tan buenos resultados como los obtenidos con los algoritmos en el estado del arte con un número mucho menor de evaluaciones. Pero la gran

ventaja de este algoritmo es que proporciona zonas de información que dan una información valiosa sobre qué valores de la variable independiente se debe trabajar para conseguir una mejor clasificación o para razonar a nivel físico y químico que componentes de las sustancias están contribuyendo a la discriminación.

Propuesta de cálculo rápido del error leave-one-out en SVM Se propone un nuevo método para evaluar el error leave-one-out, relacionado con el error de generalización de las SVM, sin necesidad de utilizar cotas de estimación. El método parte de las condiciones dadas por estos estimadores para solo tener que evaluar unas pocas muestras del conjunto de entrenamiento a las que se aplica las técnicas de inicialización por semilla y reducción para facilitar su cálculo. Además se incorpora el número de vectores soporte al criterio a minimizar de forma que se atienda esencialmente a este parámetro cuando el error obtenido es suficientemente bajo y en caso contrario se optimizará por mínimo error primero y ante la igualdad de éste se seleccionarán los hiperparámetros que minimicen los vectores soporte.

Búsqueda de hiperparámetros por algoritmos meta-heurísticos Dado que la función propuesta no es derivable, se propone el uso de los siguientes algoritmos para optimizar los parámetros de una SVM:

- Algoritmos genéticos (GA) de optimización continua.
- Simulated annealing (SA) con múltiples soluciones por temperatura.
- Adaptación del algoritmo de optimización por colonia de hormigas (Ant Colony).
- Particle swarm optimization (PSO).

Aplicación de probabilidades a la estrategia uno contra todos La forma habitual de comparar las salidas de una SVM para problemas multiclase no es correcta desde un punto de vista matemático. En esta tesis se utiliza el método de Platt para estimar la probabilidad de pertenencia a una clase dado un valor de la función de decisión, optimizando los parámetros de una sigmoide que será colocada como salida de cada uno de los clasificadores binarios. Las salidas de estas sigmoides sí pueden ser comparadas matemáticamente.

Propuesta de grafos ordenados Además de la estrategia uno contra todos, la estrategia uno contra uno puede proporcionar clasificadores más sencillos. Si se elabora un grafo de decisión solo es necesario comprobar un número de clasificadores igual al número de clases. Se aporta un nuevo algoritmo de ordenación del grafo para minimizar el número de operaciones medio de la fase de test.

Agrupación de vectores soporte El algoritmo aportado tiene su hipótesis en que cuando aparece un vector soporte, es muy probable que encontremos una muestra cercana debido a la toma de muestras como resultado de un ensayo efectuado en las mismas condiciones. Esto hace que nos planteemos el reentrenamiento de las SVM agrupando los vectores soporte de la misma clase que se encuentren cercanos.

Reducción de operaciones en la función de decisión. Esta aportación se aplica sobre kernels basados en la distancia euclídea, como son algunos de los utilizados en esta tesis. El método establece en cada momento las cotas de las contribuciones restantes positivas y negativas y detiene el cálculo de la función de decisión cuando no existe posibilidad de cambiar el signo de dicha función.

6.3. Futuras líneas de investigación

Esta tesis ha sido la primera que se realiza en el departamento de Teoría de la Señal y Comunicaciones sobre el procesado y reconocimiento de señales procedentes de sensores de gases y líquidos. Así, las ideas propuestas no solo pretenden dar respuesta a problemas planteados dentro de dicho campo, sino que tiene como objetivo abrir nuevas líneas de investigación. Se presentan a continuación una serie de puntos, algunos de los cuales ya están siendo trabajados aunque de forma preliminar.

Realización de un sistema hardware. Esta vía conecta la línea de investigación estudiada en esta tesis con la tercera línea planteada por las redes de excelencia europeas sobre nariz y lengua electrónica. En el marco de esta tesis se desarrolló el hardware necesario para la adquisición del conjunto de datos de separación de alcoholes. La nueva propuesta incluye el diseño y la implementación de un sistema autónomo donde se ejecuten los métodos propuestos y se controlen las técnicas de la parte sensora. Así, se deberá realizar un control de flujo y de las tensiones que modularan en temperatura a los sensores.

Ampliación del estudio a otras técnicas sensoras. En esta tesis se ha querido recoger una variedad sobre los tipos de sensores utilizados y las técnicas utilizadas para producir señales dinámicas. El objetivo de esta nueva línea es probar con nuevos tipos de datos y sensores, bien mediante conjuntos de datos externos o utilizando el sistema hardware descrito en el apartado anterior. Esta vía incluye también el estudio de sensores experimentales no tan relacionados con la nariz y lengua electrónica pero que proporcionan información analítica de suma utilidad.

Estudio de técnicas de regresión. Esta tesis se ha fijado en los métodos de clasificación y la optimización de los métodos kernel por haber obtenido muy buenos

resultados en la comparación planteada. Para muchas aplicaciones, el objetivo de una técnica analítica multivariante es conocer con exactitud la concentración de una sustancia o el valor exacto de una determinada medida. Aparece un nuevo campo en el que en lugar de utilizar técnicas de clasificación debemos acudir a las denominadas técnicas de regresión, que deben ser comprendidas en profundidad para poder hacer un estudio equivalente al realizado con las técnicas de clasificación en esta tesis. Este campo es de extrema utilidad en aquellas aplicaciones que buscan obtener medidas analíticas sometidas a una legislación, para las que se debe aportar la cantidad y estar seguros que no sobrepasan ciertos límites.

Modelado de fuentes Uno de los aspectos más costosos en este campo de investigación es la necesidad de contar con múltiples experimentos que permitan realizar un entrenamiento de alguno de los métodos expuestos. En este sentido, un gran avance sería contar con modelos matemáticos que permitieran simular ensayos en condiciones ambientales diversas. Para poder abordar el estudio de esta tarea es necesario contar con técnicas de estimación de las funciones densidad de probabilidad que caractericen los sensores y el ruido asociado a los mismos. De esta forma sería posible contar con ensayos virtuales que reducirían el tiempo de experimentación y serían ratificados sobre muestras reales.

Integrar la información de gases y líquidos. La información del gusto y el olfato se mezcla en nuestro cerebro para dar una respuesta conjunta a la identificación de una amplia variedad de sustancias. Así, se debería aprovechar esta analogía para diseñar sistemas y métodos que tuvieran presente la información que proviene tanto de sensores de gases como de líquidos.

Aplicación de técnicas variantes en el tiempo Uno de los grandes problemas que presentan los sistemas de nariz y lengua electrónica es la gran variabilidad de sus respuestas en el tiempo. Para compensar este efecto se utilizan técnicas estadísticas, pero todas ellas parten de la premisa de que la función densidad de probabilidad, aunque desconocida, es constante en el tiempo. Existen otro tipo de métodos de clasificación donde la función densidad de probabilidad puede ser variable en el tiempo, por lo que se debería probar este tipo de métodos en presencia de un comportamiento fuerte de deriva.

Análisis de métodos de clasificación con características variables En esta tesis se ha supuesto que todos los patrones que forman parte del sistema de reconocimiento tienen el mismo número de características. Sin embargo, existen trabajos muy recientes, como los basados en kernels racionales, sobre métodos de clasificación que permiten utilizar patrones de diferente longitud, de forma

que podríamos extraer más coeficientes en aquellas señales que requieran una información más completa.

Modelado de inspiración biológica Los intentos de diseñar clasificadores con una inspiración en los modelos de la naturaleza tienen una gran aceptación por parte de la comunidad científica. En este sentido, se debería tratar de imitar en mayor grado el comportamiento del cerebro humano a la hora de discriminar olores y gustos, incluyendo el aprendizaje no supervisado.

Aplicación en otros campos de investigación Los métodos de clasificación, optimización y de procesado de señal aportados en esta tesis pueden ser aplicados sobre otras líneas de investigación completamente diferentes. En concreto, es de gran interés aplicarlos al reconocimiento de imágenes para su integración con otras líneas desarrolladas por este grupo de investigación.

Bibliografía

[Abe05] S. Abe, Support vector machines for pattern classification, Springer-Verlag, Berlin, **2005**.

[Acevedo05] J. Acevedo, S. Maldonado, S. Al-Khalifa, P. Gil y J. W. Gardner, "Design of a programmable, portable and low-cost electronic nose", en Proc. of ISOEN 05, editado por S. Marco y I. Montoliu, Universidad de Barcelona, **2005**, 304–307.

[Acevedo06] J. Acevedo, S. Maldonado-Bascón, S. Lafuente-Arroyo, H. Gómez-Moreno y P. Gil-Jiménez, "Model Selection for Support Vector Machines Using Ant Colony Optimization in an Electronic Nose Application", en Ant Colony Optimization and Swarm Intelligence, 5th International Workshop, ANTS 2006, Brussels, Belgium, September 4-7, 2006, Proceedings, editado por M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli y T. Stützle, **2006**, 468–475.

[Acevedo07a] F. Acevedo, S. Maldonado, E. Domínguez, A. Narváez y F. López, "Probabilistic support vector machines for multi-class alcohol identification", *Sensors and Actuators B: Chemical*, tomo 122, 1, 227–235, **2007**.

[Acevedo07b] J. Acevedo, S. Maldonado-Bascón, P. Siegmann, S. Lafuente-Arroyo y P. Gil, "Tuning L1-SVM Hyperparameters with Modified Radius Margin Bounds and Simulated Annealing", en Computational and Ambient Intelligence, 9th International Work-Conference on Artificial Neural Networks, IWANN 2007, San Sebastián, Spain, June 20-22, 2007, Proceedings, editado por F. S. Hernández, A. Prieto, J. Cabestany y M. Graña, **2007**, 284–291.

[Acevedo07c] J. Acevedo, S. Maldonado-Bascón, P. Siegmann, S. Lafuente-Arroyo y P. Gil-Jiménez, "Multi-class Support Vector Machines

Based on Arranged Decision Graphs and Particle Swarm Optimization for Model Selection", en Adaptive and Natural Computing Algorithms, 8th International Conference, ICANNGA 2007, Warsaw, Poland, April 11-14, 2007, Proceedings, Part II, editado por B. Beliczynski, A. Dzielinski, M. Iwanowski y B. Ribeiro, Springer, **2007**, tomo 4432 de *Lecture Notes in Computer Science*, 238–245.

[Acevedo09] J. Acevedo, S. Maldonado, S. Lafuente, P. Siegmann y F. López, "Computational load reduction in decision functions using support vector machines", *Signal Processing*, Aceptado. DOI 10.1016/j.sigpro.2009.63.032., **2009**.

[Adams03] J. D. Adams, G. Parrot, C. Bauer, T. Sant, L. Manning y M. Jones, "Nanowatt chemical vapor detection with self-sensing, piezoelectric microcantilever array", *Applied Physics Letters*, tomo 83, 3428–3430, **2003**.

[Nakamura94] M. Nakamura, I. Sugimoto, H. Kuwano y R. Lemos, "Chemical sensing by analysing dynamics of plasma polymer film-coated sensors", *Sensors and Actuators B: Chemical*, tomo 20, 231–237, **1994**.

[Al-Khalifa03] S. Al-Khalifa, S. Maldonado-Bascón y J. Gardner, "Identification of CO and NO2 using a thermally resistive microsensor and support vector machine", *IEE Prodeedings Scientific Technology Journal*, tomo 150, **2003**.

[Apetrei07] C. Apetrei, I. M. Apetrei, I. Nevares, M. Alamo, V. Parra, M. L. Rodriguez-Mendez y J. A. Saja, "Using an e-tongue based on voltammetric electrodes to discriminate among red wines aged in oak barrels or aged using alternative methods: Correlation between electrochemical signals and analytical parameters", *Electrochimica Acta*, tomo 52, 7, 2588–2594, **2007**.

[Baldini06] F. Baldini, Optical chemical sensors, Springer, Nueva York, **2006**.

[Barker03] M. Barker y W. Rayens, "Partial least squares for discrimination", *Journal of Chemometrics*, tomo 17, 166–173, **2003**.

[Barlett92] P. N. Barlett y J. W. Gardner, Sensors and sensory systems for an electronic nose, NATO Science Series, Londres, **1992**.

[Bartlett98] P. Bartlett y J. Shawe-Taylor, "Generalization performance of support vector machines and other pattern classifiers", en Advances in kernel methods. support vector learning, editado por C. B. B. Scholkopf y A. Smola, **1998**.

[Baxter97] J. M. Baxter, M. E. Crews, J. Dennis, I. Goodall y D. Anderson, "The determination of the authenticity of wine from its trace elements composition", *Food Chemistry*, tomo 60, 443–450, **1997**.

[Bishop06] C. M. Bishop, Pattern recognition and machine learning, Springer, Nueva York, **2006**.

[Bredensteiner99] E. Bredensteiner y K. Bennet, "Multicategory classification by support vector machines", *Computational Optimizations and Applications*, tomo 12, 53-79, **1999**.

[Breijo02] E. G. Breijo, L. G. Sanchez, J. I. Civera, A. T. Ferrando y G. P. Boluda, "Thick-film multisensor for determining water quality parameters", *IECON 02 [Annual Conference of the Industrial Electronics Society, 2002 Proceedings of the IEEE/IE International*, tomo 4, 2791–2796, **2002**.

[Breiman01] L. Breiman, "Random Forests", *Machine Learning*, tomo 45, 1, 5–32, **2001**.

[Brezmes01] J. Brezmes, Diseño de una nariz electrónica para la determinación no destructiva del grado de maduración de la fruta, Tesis Doctoral, Universidad Politécnica de Cataluña, **2001**.

[Brudzewski06] K. Brudzewski, S. Osowski, T. Markiewicz y J. Ulaczyk, "Classification of gasoline with supplement of bio-products by means of an electronic nose and SVM neural network", *Sensors and Actuators B: Chemical*, tomo 113, 135–141, **2006**.

[Burges98] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, tomo 2, 121–167, **1998**.

[Calvo07] D. Calvo, A. Duran y M. del Valle, "Use of sequential injection analysis to construct an electronic-tongue: Application to multidetermination employing the transient response of a potentiometric sensor array", *Analytica Chimica Acta*, tomo 600, 97–104, **2007**.

[Carpenter91] G. A. Carpenter, S. Grossberg y J. H. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network", *Neural Networks*, tomo 4, 5, 565, **1991**.

[Carpenter92] G. A. Carpenter, S. Grossberg, M. Markuzon, J. H. Reynolds y D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps", *IEEE Transactions on Neural Networks*, tomo 3, 5, 698–713, **1992**.

[Casalinuovo06] I. A. Casalinuovo, D. D. Pierro, M. Coletta y P. D. Francesco, "Application of electronic noses for disease diagnosis and food spoilage detection", *Sensors*, tomo 6, 1428–1439, **2006**.

[Cavicchi95] R. E. Cavicchi, J. S. Suehle, K. G. Kreider, M. Gaitan y P. Chaparala, "Fast temperature programmed sensing for microhotplate gas sensors", *IEEE Electron Device Letters*, tomo 16, 286–288, **1995**.

[Cawley07] G. Cawley y N. L. Talbot, "Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters", *Journal of machine learning research*, tomo 8, 841–861, **2007**.

[Chang01] C. Chang y J. Chih, LIBSVM: a library for support vector machines, **2001**, URL `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Chapelle99] O. Chapelle y V. Vapnik, "Model selection for support vector machines", *Advances in Neural Information Processing Systems*, tomo 1, 230–236, **1999**.

[Chapelle02a] O. Chapelle, Support vector machines: induction principle, adaptive tuning and prior knowledge, Tesis Doctoral, Université Pierre et Marie Curie. Paris., **2002**.

[Chapelle02b] O. Chapelle, V. Vapnik, O. Bousquet y S. Mukherjee, "Choosing Multiple Parameters for Support Vector Machines", *Machine Learning*, tomo 46, 1, 131–159, **2002**.

[Chen91] S. Chen, C. F. Cowan y P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks", *IEEE Transactions on Neural Networks*, tomo 2, 2, 302–309, **1991**.

[Chen05] X. Chen, M. Cao, Y. Li, W. Hu, P. Wang, K. Ying y H. Pan, "A study of an electronic nose for detection of lung cancer based on a virtual SAW gas sensors array and imaging recognition method", *Measurement Science and Technology*, tomo 16, 8, 1535–1546, **2005**.

[Chung03] K. Chung, W. Kao, C. Sun, L. Wang y C. Lin, "Radius margin bounds for support vector machines with the RBF kernel", *Neural Computation*, tomo 15, 11, 2643–2681, **2003**.

[Clerc02] M. Clerc y J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space.", *IEEE Transactions on Evolutionary Computation*, tomo 6, 1, 58–73, **2002**.

[Clifford83] P. K. Clifford y D. T. Tuma, "Characteristics of semiconductor gas sensor II, Transient response to temperature change", *Sensors and Actuators B: Chemical*, tomo 3, 233–254, **1983**.

[Cohen95] A. Cohen y R. D. Ryan, Wavelets and multiscale signal processing, Chapman and Hall, **1995**.

[Coifman92] R. R. Coifman y M. V. . Wickerhauser, "Entropy-based algorithms for best basis selection", *Information Theory, IEEE Transactions on*, tomo 38, 2, 713–718, **1992**.

[Correig05] X. Correig, "Monitoring the process of roasting hazelnuts with a multisensor system", en Proc. of ISOEN 05, editado por S. Marco y I. Montoliu, Universidad de Barcelona, **2005**, 252–255.

[Crammer01] K. Crammer y Y. Singer, "On the Algorithmic Implementation of Multi-class SVMs", *Journal of machine learning research*, tomo 2, 265–292, **2001**.

[Cybenko89] G. Cybenko, "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals and Systems*, tomo 2, 303-314, **1989**.

[Dai07] H. Dai y G. Shi, "Gas sensors based on conducting polymers", *Sensors*, tomo 7, 267–307, **2007**.

[Daubechies92] I. Daubechies, Ten lectures on wavelets, SIAM, Nueva York, **1992**.

[Díaz-Delgado02] R. Díaz-Delgado, Tin oxide gas sensors: an electrochemical approach, Tesis Doctoral, Universidad de Barcelona, **2002**.

[DeCoste00] D. DeCoste y K. Wagstaff, "Alpha seeding for support vector machines", en KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, **2000**, 345–349.

[Decoste01] D. Decoste, M. Burl, A. Hopkins y N. Lewis, "Support vector machines and kernel fisher discriminants: a case study using electronic nose data", en Proceedings of the 4th Workshop on Mining Scientific Datasets, **2001**.

[Demsar06] J. Demsar, "Statistical comparison of classifiers over multilple datasets", *Journal of Machine Learning Research*, tomo 7, 1, 1–30, **2006**.

[Demuth05] H. Demuth y M. Beale, Neural network for use with matlab, Mathworks Inc, **2005**.

[Devijver82] P. A. Devijver y J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, London, **1982**.

[Distante02] C. Distante, M. Leo, P. Siciliano y K. C. Persaud, "On the study of feature extraction methods for an electronic nose", *Sensors and Actuators B: Chemical*, tomo 87, 274–288, **2002**.

[Distante03] C. Distante, N. Ancona y P. Siciliano, "Support vector machines for olfactory signals recognition", *Sensors and Actuators B: Chemical*, tomo 88, 30–39, **2003**.

[Duan03] K. Duan, S. Keerthi y A. Poo, "Evaluation of simple performance measures for tuning SVM hyperparameters", *Neurocomputing*, tomo 51, 41–59, **2003**.

[Duda00] R. O. Duda, P. E. Hart y D. G. Stork, Pattern Classification, Wiley-Interscience Publication, **2000**.

[Durán05a] A. Durán, M. Cortina, L. Velasco, J. A. Rodríguez, S. Alegret y M. Valle, "Virtual instrument as an atomated potentiometric e-tongue based on SIA", en Proc. of ISOEN 05, editado por S. Marco y I. Montoliu, Universidad de Barcelona, **2005**, 116–120.

[Durán05b] M. Durán, G. Sberveglieri y J. W. Gardner, "Benchmarking feature selecition for e-noses", en Proc. of ISOEN 05, editado por S. Marco y I. Montoliu, Universidad de Barcelona, **2005**, 33–35.

[Efron83] B. Efron y G. Gong, "A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation", *The American Statistician*, tomo 37, 1, 36–48, **1983**.

[Efron86] B. Efron y R. Tibshirani, "Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy", tomo 1, 1, 54–75, **1986**.

[Efron97] B. Efron y R. Tibshirani, "Improvements on Cross-Validation: The .632+ Bootstrap Method", *Journal of the American Statistical Association*, tomo 92, 438, 548–560, **1997**.

[Eklov97] T. Eklov, P. Martesson y I. Lundstrom, "Enhanced selectivity of MOSFET gas sensors by systematical analysis of transient parameters", *Analytica Chimica Acta*, tomo 353, 291–300, **1997**.

[Fan05] R. Fan, P.Chen y C. Lin, "Working Set Selection Using Second Order Information for Training Support Vector Machines", *Journal of Machine Learning Research*, tomo 6, 1889–1918, **2005**.

[Franc04] V. Franc, Statistical pattern recognition toolbox for matlab, Tesis Doctoral, Czech Technical University in Prague, **2004**.

[Gallardo04] J. Gallardo, S. Alegret y M. Valle, "A flow-injection electronic tongue based on potentiometric sensors for the determination of nitrate in the presence of chloride", *Sensors and Actuators B: Chemical*, tomo 101, 72–80, **2004**.

[Gallardo05] J. Gallardo, S. Alegret y M. Valle, "Application of a potentiometric electronic tongue as a classification tool in food analysis", *Talanta*, tomo 66, 5, 1303–1309, **2005**.

[Gan05] H. L. Gan, C. P. Tan, Y. B. Man, I. Noraini y S. A. Nazimah, "Monitoring the storage stability of RBD palm olein using the electronic nose", *Food Chemistry*, tomo 89, 2, 271–282, **2005**.

[Garcia04] D. Garcia, N. Barie, M. Rapp y R. Aparicio, "Analysis of virgin olive oil volatiles by a novel electronic nose based on a miniaturized SAW sensor array coupled with SPME enhanced heads-

pace enrichment", *Journal of agricultural and food chemistry*, tomo 25, 7475–7479, **2004**.

[Garcia06] M. Garcia, M. Aleixandre, J. Gutierrez y M. C. Horrillo, "Electronic nose for wine discrimination", *Sensors and Actuators B: Chemical*, tomo 113, 911-916, **2006**.

[Gardner92] J. W. Gardner, H. V. Shurmer y T. T. Tan, "Application of an electronic nose to the discrimination of coffees", *Sensors and Actuators B: Chemical*, tomo 6, 71–75, **1992**.

[Gardner94] J. W. Gardner y P. N. Bartlett, "A brief history of electronic noses", *Sensors and Actuators B: Chemical*, tomo 18, 210–211, **1994**.

[Gardner98] J. W. Gardner, M. Craven, C. Dow y E. L. Hines, "The prediction of bacteria type and culture growth phase by an electronic nose with a multi-layer perceptron network", *Measurement Science and Technology*, tomo 9, 1, 120–127, **1998**.

[Gardner99] J. W. Gardner y P. N. Barlett, Electronic nose, principles and applications, Oxford Science Publications, Londres, **1999**.

[Gardner01] J. W. Gardner y V. K. Varadan, Microsensors, Mems and Smart Devices, John Wiley & Sons, Inc., New York, NY, USA, **2001**.

[Gardner04] J. W. Gardner y J. Yinon, "Electronic Noses and Sensors for the Detection of Explosives", **2004**.

[Geladi86] P. Geladi y B. R. Kowalski, "PLS Tutorial", *Analitica Chimica Acta*, tomo 185, 1–17, **1986**.

[Gestel04] T. Gestel, J. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. Moor y J. Vandewalle, "Benchmarking least squares support vector machine classifiers", *Machine Learning*, tomo 54, 1, 5-32, **2004**.

[Gil07] R. Gil, M. Rosa, R. Vicen y F. López, "A new algorithm for the fast search of the k nearest patterns", en Proc. of the Fifteenth European Signal Processing Conference, editado por M. Dománski, European Association for Signal Processing, **2007**.

[Goldberg89] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Professional, Indianapolis, **1989**.

[Gomez04] M. Gomez y F. Heredia, "Effect of the maceration technique on the relationships between anthocyanin composition and objective color Syrah wines", *Journal of Agricultural Food and Chemistry*, , 52, 5117–5123, **2004**.

[Gorton95] L. Gorton, "Carbon paste electrodes modified with enzymes, tissues, and cells", *Electroanalysis*, tomo 7, 23–45, **1995**.

[Grandvalet02] Y. Grandvalet y S. Canu, "Adaptive scaling for feature selection in SVMs", en Advances in Neural Information Processing Systems, editado por S. Becker, S. Thrun y K. Obermayer, MIT Press, **2002**, 553–560.

[Gu02] C. Gu, Smoothing spline ANOVA models, Springer-Verlag, Nueva York, **2002**.

[Gualdron06] O. Gualdron, E. Llobet, J. Brezmes, X. Vilanova y X. Correig, "Coupling fast variable selection methods to neural network-based classifiers: Application to multisensor systems", *Sensors and Actuators B: Chemical*, tomo 114, 522–529, **2006**.

[Gutierrez-Osuna99a] R. Gutierrez-Osuna, "Transient response analysis of an electronic nose using multi-exponential models", *Sensors and Actuators B: Chemical*, tomo 61, 170–182, **1999**.

[Gutierrez-Osuna99b] R. Gutierrez-Osuna y H. T. Nagle, "A Method for Evaluating Data-Preprocessing Techniques for Odor Classification with an Array of Gas Sensors", *IEEE Transactions on Systems, Man, And Cybernetics*, tomo 29, 626–632, **1999**.

[Gutierrez-Osuna02] R. Gutierrez-Osuna, "Pattern analysis for machine olfaction: a review", *IEEE sensors journal*, tomo 2, 3, **2002**.

[Gutierrez-Osuna03] R. Gutierrez-Osuna, "Signal processing methods for drift compensation", en 2nd Workshop of NOSE II, **2003**.

[Guyon98] I. Guyon, J. Makhoul, R. Schwartz y V. Vapnik, "What Size Test Set Gives Good Error Rate Estimates?", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, tomo 20, 1, 52–64, **1998**.

[Hauptmann00] P. Hauptmann, R. Borngraeber, J. Schroeder y J. Auge, "Artificial electronic tongue in comparison to the electronic nose. State of the art and trends", *Frequency Control Symposium and Exhibition, 2000 Proceedings of the 2000 IEEE/EIA International*, 22–29, **2000**.

[Heilig97] A. Heilig, N. Bârsan, U. Weimar, M. Schweizer-Berberich, J. W. Gardner y W. Göpel, "Gas identification by modulating temperatures of SnO2-based thick-film sensors", *Sensors Actuators B: Chemical*, tomo 43, 45–51, **1997**.

[Hierlemann05] A. Hierlemann, Technology and development observation report, *Inf. téc.*, NOSE II, **2005**.

[Hierlemann08] A. Hierlemann y R. Gutierrez-Osuna, "High order chemical sensing", *Chemical Reviews*, tomo 108, 563–613, **2008**.

[Hines99] E. L. Hines, E. Llobet y J. W. Gardner, "Electronic noses: a review on signal processing techniques", *IEE Proceedings on circuit devices and systems*, tomo 146, 6, **1999**.

[Holger04] H. Holger y T. Stützle, Stochastic Local Search: Foundations and Applications, Morgan Kaufmann, **2004**.

[Horrillo98] M. C. Horrillo, J. Getino, L. Arés, J. I. Robla, I. Sayago y F. J. Gutiérrez, "Measurements of VOCs with a semiconductor electronic nose", *Journal of Electrochemistry Society*, tomo 145, 7, 2486Ű–2489, **1998**.

[Hush93] D. R. Hush y B. G. Home, "Progress in supervised neural network", *IEEE Signal Processing Magazine*, tomo 10, 8–39, **1993**.

[Huyberechts97] G. Huyberechts, P. Szecowka, J. Roggen y B. W. Licznerski, "Simultaneous quantification of carbon monoxide and methane in humid air using a sensor array and an artificial neural network", *Sensors and Actuators B: Chemical*, tomo 45, 123–130, **1997**.

[Ingber93] L. Ingber y B. Rosen, Genetic algorithms and very fast simulated reannealing: A comparison, *Inf. téc.*, Lester Ingber Research, **1993**.

[Ionescu02] R. Ionescu y E. Llobet, "Wavelet transform-based fast feature extraction from temperature modulated semiconductor gas sensors", *Sensors and Actuators B: Chemical*, tomo 81, 289–295, **2002**.

[Jenssen07] R. Jenssen, T. Eltoft, M. Girolami y D. Erdogmus, "Kernel maximum entropy data transformation and an enhanced spectral clustering algorithm", en Advances in neural information processing systems, 19, editado por B. Schölkopf, J. Platt y T. Hoffman, MIT Press, Cambridge, MA, 633–640, **2007**.

[Joachims00] T. Joachims, "Estimating the generalization performance of a SVM efficiently", en Proc. of ICML-00, editado por P. Langley, Morgan Kaufmann , San Francisco, US, **2000**, 431–438.

[Kanellopoulos97] I. Kanellopoulos y G. Wilkinson, "Strategies and best practice for neural network image classification", *Remote Sensing*, tomo 4), 18, 711–725, **1997**.

[Kenneth04] K. S. Kenneth, N. A. Rakow y A.Sen, "Colorimetric sensor arrays for molecular recognition", *Tetrahedron*, tomo 60, 49, 11133–11138, **2004**.

[Kermani99] B. G. Kermani, S. Schiffman y H. T. Nagle, "Using neural networks and genetic algorithms to enhance performance in an electronic nose", *IEEE Transactions on Biomedical Engineering*, tomo 46, 4, 429–439, **1999**.

[Kounaves97] S. P. Kounaves, "Voltammetry techniques", en Handbook of instrumental techniques for analytical chemistry, editado por I. Settle, Prentice-Hall, **1997**.

[Kreßel98] U. Kreßel, "Pairwise Classification and Support Vector Machines", en Advances in Kernel Methods – Support Vector Learning, editado por B. Schölkopf, C. Burges y A. Smola, MIT Press, 225–268, **1998**.

[Kuswandi07] B. Kuswandi, J. Huskens y W. Verboom, "Optical sensing systems for microfluidic devices: A review", *Analytica Chimica Acta*, tomo 601, 2, 141–155, **2007**.

[Laborante07] A. F. Laborante, A. Morales-Rubio, M. Guardia y B. F. Reis, "A multicommuted stop-flow system employing LEDs-based photometer for the sequential determination of anionic and cationic surfactants in water", *Analytical Chimica Acta*, tomo 600, 1, 58–65, **2007**.

[Lee04] M. Lee, S. Keerthi, J. Chong y D. DeCoste, "An efficient method for computing leave-one-out error in support vector machines with Gaussian kernels", *IEEE Transactions on Neural Networks*, tomo 15, 3, 750–757, **2004**.

[Lee06] S. Lee, Encyclopedia of chemical processing, CRC Press, Londres, **2006**.

[Legin01] A. Legin, A. Rudnitskaya, D. Clapham, B. Seleznev, K. Lord y Y. Vlasov, "Electronic tongue for pharmaceutical analytics: quantification of tastes and masking effects", *Analytical and Bioanalytical Chemistry*, tomo 380, 1, 36–45, **2001**.

[Leone05] A. Leone, C. Distante, N. Ancona, K. C. Persaud, E. Stella y P. Siciliano, "A powerful method for feature extraction and compression of electronic nose responses", *Sensors and Actuators B, Chemical*, tomo 105, 378–392, **2005**.

[Lessmann06] S. Lessmann, R. Stahlbock y S.Crone, "Genetic algorithms for support vector machine model selection", *Neural Networks*, tomo 1, 3063–3069, **2006**.

[Li05] G. Li, C. Martinez y S. Semancik, "Controlled electrophoretic patterning of polyaniline from a colloidal suspension", *Journal of the American Chemical Society*, tomo 127, 13, 4903–4909, **2005**.

[Lin07] H. Lin, C. J. Lin y R. C. Weng, "A note on Platt's probabilistic outputs for support vector machines", *Machine Learning*, tomo 68, 3, 267–276, **2007**.

[Llobet97] E. Llobet, Selectivity enhancement of metal oxide semiconductor chemical sensors through the study of their transient response to a step-change in gas concentration, Tesis Doctoral, Universidad Politécnica de Cataluña, **1997**.

[Llobet02] E. Llobet, J. Brezmes, R. Ionescu, X. Vilanova, S. Al-Khalifa, J. W. Gardner, N. Barsan y X. Correig, "Wavelet transform and fuzzy ARTMAP-based pattern recognition for fast gas identification using a micro-hotplate gas sensor", *Sensors and Actuators B: Chemical*, tomo 83, 238–244, **2002**.

[Llobet06] E. Llobet, "Dynamic pattern recognition methods and system identification", en Handbook of machine olfaction, editado por T. C. Pearce, S. S. Schiffman, H. T. Nagle y J. W. Gardner, Wiley VCH, Weinheim, **2006**.

[Llobet07] E. Llobet, O. Gualdron, M. Vinaixa, N. El-Barbri, J. Brezmes, X. Vilanova, B. Bouchikhi, R. Gomez, J. Carrasco y X. Correig, "Efficient feature selection for mass spectrometry based electronic nose applications", *Chemometrics and Intelligent Laboratory Systems*, tomo 85, 2, 253–261, **2007**.

[Lozano05] J. Lozano, J. P. Santos y M. C. Horrillo, "Classification of white wine aromas with an electronic nose", *Talanta*, tomo 67, 610–616, **2005**.

[Luntz69] A. Luntz y V. Brailovsky, "On the estimation of characters obtained in statistical procedure of recognition", *Technicheskaya Kibernetica*, tomo 3, **1969**.

[Maestre05] E. Maestre, I. Katakis, A. Narváez y E. Domínguez, "A multianalyte flow electrochemical cell: application to the simultaneous determination of carbohydrates based on bioelectrocatalytic detection", *Biosensors and Bioelectronic*, tomo 21, 774, **2005**.

[Mallat99] S. Mallat, A wavelet tour of signal processing. 2nd Edition, Academic Press, New York, **1999**.

[Martina07] V. Martina, K. Ionescu, L. Pigani, F. Terzi, A. Ulrici, C. Zanardi y R. Seeber, "Development of an electronic tongue based on a PEDOT-modified voltammetric sensor", *Analytical and Bioanalytical Chemistry*, tomo 387, 6, 210–2110, **2007**.

[Martinelli04] E. Martinelli, G. Pennazza, C. D. Natale y A. DAmico, "Chemical sensors clustering with the dynamic moments approach", *Sensors and Actuators B: Chemical*, tomo 101, 346–352, **2004**.

[Miller04] D. Miller, R. Arguello y G. Greenwood, "Evolving artificial neural network structures: experimental results for biologically-inspired adaptive mutations", en Evolutionary Computation. CEC2004., **2004**.

[Mitrovics98] J. Mitrovics, H. Ulmer, U. Weimar y W. Göpel, "Modular sensor systems for gas sensing and odor monitoring: The MOSES Concept", *ACS Symposium Series: Chemical Sensors and Interfacial Design*, tomo 31, 307–315, **1998**.

[Montag01] S. Montag, M. Frank, H. Ulmer, D. Wernet, W. Gopel y H. G. Rammensee, "Electronic nose detects major histocompatibility complex-dependent prerenal and postrenal odor components", *Proceedings of the National Academy of Sciences*, tomo 17, 1–6, **2001**.

[Moreno02] F. Moreno, L. Mico y J. Oncina, "Extending LAESA fast nearest neighbour algorithm to find the k nearest neighbours.", *Lecture Notes in Computer Science*, , 2396, 718–Ű724, **2002**.

[Moreno06] L. Moreno, C. J. A. Merlos, N. Abramova y A. Bratov, "Multisensor array used as an electronic tongue for mineral water analysis", *Sensors and Actuators B: Chemical*, tomo 116, 130-134, **2006**.

[Moseley91] P. T. Moseley, A. M. Stoneham y D. Williams, Techniques and mechanisms in gas sensing, Adam Hilger, Bristol, **1991**.

[Natale03] C. D. Natale, A. Macagnano, E. Martinelli, R. Paolesse, G. D'Arcangelo, C. Roscioni, A. Finazzi-Agro y A. D'Amico, "Lung cancer identification by the analysis of breath by means of an array of non-selective gas sensors", *Biosensors and Bioelectronics*, tomo 18, 10, 1209–1218, **2003**.

[Olafdottir05] G. Olafdottir, "Rapid control of smoked atalantic salmon quality by electronic nose: correlation with classical evaluation methods", en Proc. of ISOEN 05, editado por S. Marco y I. Montoliu, Universidad de Barcelona, **2005**, 315–317.

[Panigrahi05] S. Panigrahi, D. Banerjee, S. Balasubrmanian, H. Gu, C. M. Logue y M. Marchello, "Wavelet-based classification models for discriminating meat contamination using a comercial electronic

nose", en Proc. of ISOEN 05, editado por S. Marco y I. Montoliu, Universidad de Barcelona, **2005**, 96–102.

[Panigrahi08] S. Panigrahi, C. Young, L. Khot, J. Glower y C. Logue, "Integrated electronic nose system for detection of Salmonella contamination in meat", *IEEE Sensors Applications Symposium*, tomo 1, 85–88, **2008**.

[Pardo05] M. Pardo y G. Sberveglieri, "Classification of electronic nose data with support vector machines", *Sensors and Actuators B: Chemical*, tomo 107, 730–737, **2005**.

[Pardo07] M. Pardo y G. Sberveglieri, "Random forests and nearest shrunken centroids for the classification of sensor array data", *Sensors and Actuators B: Chemical*, tomo disponible on-line, **2007**.

[Parra06] V. Parra, A. Arrieta, J. A. Fernandez, M. L. Rodriguez y J. A. D. Saja, "Electronic tongue based on chemically modified electrodes and voltammetry for the detection of adulterations in wines", *Sensors and Actuators B: Chemical*, tomo 118, 448–453, **2006**.

[Pearce06] T. C. Pearce, S. S. Schiffman, H. T. Nagle y J. W. Gardner, Handbook of machine olfaction, Wiley VCH, Weinheim, **2006**.

[Peng05] H. Peng, F. Long y C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, tomo 27, 8, 1226–1238, **2005**.

[Perera02] A. Perera, T. Sundic, A. Pardo, R. Gutierrez-Osuna y S. Marco, "A portable electronic nose based on embedded PC technology and GNU/Linux: hardware, software and applications", *IEEE Sensors Journal*, tomo 2, 3, 235–246, **2002**.

[Persaud82] K. C. Persaud y G. Dodd, "Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose", *Nature*, tomo 299, 352–355, **1982**.

[Phaisangittisagul07] E. Phaisangittisagul, Signal Processing using Wavelets for Enhancing Electronic Nose Performance, Tesis Doctoral, North Carolina State University, **2007**.

[Platt98] J. Platt, "Fast training of SVMs using sequential minimal optimization", en Advances in Kernel Methods – Support Vector Learning, editado por B. Schölkopf, C. Burges y A. Smola, MIT Press, 185–208, **1998**.

[Platt99] J. Platt, "Probabilistic Outputs for Support Vector Machines and Comparison to Regularized Likelihood Methods", en Advances in Large Margin Classifiers, editado por A. Smola, B. Scholkopf y D. Schuurmans, MIT Press, 61–74, **1999**.

[Platt00] J. Platt, "Large Margin DAGs for Multiclass Classification", en Advances in Neural Information Processing Systems, editado por S. Solla, T. Keen y K. Müller, MIT Press, 547–553, **2000**.

[Proakis96] J. G. Proakis y D. G. Manolakis, Digital signal processing: principles, algorithms, and applications, Prentice Hall, **1996**.

[Pudil94] P. Pudil, J. Novicovska y J. Kitller, "Floating search methods in feature selection", *Pattern recognition letters*, tomo 15, 11, 1119–1125, **1994**.

[Röck08] F. Röck, N. Barsan y U. Weimar, "Electronic nose: Current status and future trends", *Chemical Reviews*, tomo 1, 73–85, **2008**.

[Reddy00] A. K. Reddy, Modern electrochemistry, Springer, Nueva York, **2000**.

[Ripley93] B. Ripley, "Statistical aspects of neural networks", en Networks and Chaos - Statistical and Probabilistic Aspects, editado por W. Kendall, CRC Press, Inc., Boca Raton, FL, USA, **1993**.

[Riul03] A. Riul, A. M. Gallardo, S. V. Mello, S. Bone, D. M. Taylor y L. H. Mattoso, "An electronic tongue using polypyrrole and polyaniline", *Synthetic Metals*, tomo 132, 2, 109–116, **2003**.

[Saevels04] S. Saevels, A. Z. Berna, J. Lammertyn, C. D. Natale y B. M. Nicolai, "Characterisation of QMB sensors by means of the BET adsorption isotherm", *Sensors and Actuators B: Chemical*, tomo 101, 242–251, **2004**.

[Salomon02] P. Salomon, P. Sibani y R. Frost, Facts, conjectures, and improvements for simulated annealing, SIAM, Nueva York, **2002**.

[Scholkopf01] B. Scholkopf y A. J. Smola, Learning with kernels: support vector machines, regularization, optimization, and beyond, MIT Press, Cambridge, MA, USA, **2001**.

[Shawe-Taylor04] J. Shawe-Taylor y N. Cristianini, Kernel methods for pattern analysis, Cambridge University Press, **2004**.

[Shilbayeh04] N. F. Shilbayeh y Z. M. Iskandarani, "Quality control of coffe using an electronic nose", *American Journal of Applied Sciences*, tomo 12, 129–135, **2004**.

[Somol00] P. Somol y P. Pudil, "Oscillating search algorithms for feature selection", en Proceedings of the 15th IAPR International Conference on Pattern Recognition, **2000**, 406–409.

[Stearns76] S. Stearns, "On selecting features for pattern classifiers", en Proceedings of the 3rd International Conference on Pattern Recognition, **1976**, 71-75.

[Strang96] G. Strang y N.Ñguyen, Wavelets and filter banks, SIAM, Nueva York, **1996**.

[Suykens99] J. Suykens y J. Vandewalle, "Least squares support vector machine classifiers", *Neural Processing Letters*, tomo 9, 3, 293–300, **1999**.

[Suykens02] J. A. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor y J. Vandewalle, Least squares support vector achines, World Scientific Pub., Singapur, **2002**.

[Takagi01] S. Takagi, K. Toko, K. Wada y T. Ohki, "Quantification of suppression of bitterness using an electronic tongue", *Journal of Pharmaceutical Sciences*, tomo 90, 12, 2042–2048, **2001**.

[Tipping01] M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine", *Journal of Machine Learning Research*, tomo 1, 211–244, **2001**.

[Tipping03] M. Tipping y A. C. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models", en Proc. of the Ninth International Workshop on Artificial Intelligence and Statistics, editado por C. M. Bishop y B. J. Frey, Society for Artifical Intelligence and Statistics, **2003**.

[Trelea03] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection", *Information Processing Letters*, tomo 85, 6, 317–325, **2003**.

[Trojanowicz00] M. Trojanowicz, Flow inyection analysis, instrumentation and applications, World Scientific, **2000**.

[Vapnik98] V. Vapnik, The nature of statistical learning theory, Springer-Verlag, Berlin, **1998**.

[Vapnik00] N. V. Vapnik, The nature of statistical learning theory, Springer-Verlag, Berlin, **2000**.

[Vergara05] A. Vergara, E. Llobet, J. Brezmes, P. Ivanov, X. Vilanova, I. Gracia, C. Cane y X. Correig, "Optimised temperature modulation of metal oxide micro-hotplate gas sensors through multilevel pseudo random sequences", *Sensors and Actuators B: Chemical*, tomo 111, 271–280, **2005**.

[Vergara07] A. Vergara, E. Llobet, E. Martinelli, C. D. Natale, A. DAmico y X. Correig, "Feature extraction of metal oxide gas sensors using dynamic moments", *Sensors and Actuators B: Chemical*, tomo 122, 219–226, **2007**.

[Vlasov97] Y. Vlasov, A. Legin y A. Rudnitskaya, "Cross-sensitivity evaluation of chemical sensors for electronic tongue: determination of heavy metal ions", *Sensors and Actuators B: Chemical*, tomo 44, 532–537, **1997**.

[Weston99] J. Weston y C. Watkins, "Support vector machines for multiclass pattern recognition", en Proceedings of the seventh european symposium on artificial neural networks, **1999**.

[Weston00] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio y V. Vapnik, "Feature Selection for SVMs", en Proceedings of the NIPS 2000 conference, **2000**, 668–674.

[Wickerhauser94] M. V. Wickerhauser, Adapted wavelet analysis from theory to software, A. K. Peters Ltd., Natick, USA, **1994**.

[Wilcoxon45] F. Wilcoxon, "Individual comparisons by ranking methods", *Biometrics*, tomo 1, 80–83, **1945**.

[Wilkens64] W. Wilkens y A. D. Hatman, "An Electronic analog for the olfactory processes", *Annals of the New York Academy of Sciences*, tomo 116, 608–612, **1964**.

[Williams01] C. K. Williams y M. Seeger, "Using the Nyström method to speed up kernel machines", en Adavances in neural information processing, 13, editado por T. K. Leen, T. G. Diettrich y V. Tresp, MIT press, Cambridge, MA, **2001**.

[Winquist00] F. Winquist, S. Holmin, C. Krantz-Pulcker, P. Wide y I. Lundstrom, "A hybrid electronic tongue", *Analytica Chimica Acta*, tomo 406, 147–157, **2000**.

[Wise06] B. M. Wise, PLS Toolbox Version 4 for use with MATLAB, Eigenvector, **2006**.

[Wold77] S. Wold y M. Sjostrom, "SIMCA: A method for analyzing chemical data in terms of similarity and analogy", en Chemometrics Theory and Application, editado por B. R. Kowalski, American Chemistry Society, Whasington, 243–282, **1977**.

[Zhang08] Y. Zhang, "Evolutionary computation based automatic SVM model selection", en Fourth International Conference on Natural Computation. ICNC '08., editado por J. Ma y Y. Yin, **2008**, tomo 2, 66–70.