DISEÑO DE UN NUEVO CLASIFICADOR SUPERVISADO PARA MINERÍA DE DATOS

MÁSTER EN INVESTIGACIÓN EN INFORMÁTICA 2008-09

TRABAJO DE INVESTIGACIÓN

PROYECTO FIN DE MÁSTER EN SISTEMAS INTELIGENTES

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense
MADRID

Alumno:
Juan Piorno Campo

Directores:
Gonzalo Pajares Martinsanz
María Guijarro Mata-García

Resumen

Este trabajo se presenta como un estudio comparativo entre una gran variedad de clasificadores utilizados en la minería de datos. Hay diversos métodos aplicados por los distintos clasificadores, no obstante, este trabajo no finaliza puesto que cada clasificador ofrece buenos resultados con un número muy limitado de clases. El objetivo de este trabajo consiste en constatar que actualmente no existe ningún modelo que permita la clasificación de cualquier conjunto de muestras dado, y que simultáneamente obtenga unos resultados satisfactorios. Al mismo tiempo se ha propuesto un nuevo modelo de clasificación que mejora algunos resultados al compararlos con los mejores resultados ofrecidos por los demás clasificadores, y aunque dista de ser un clasificador generalizado, se plantea la combinación de clasificadores como una técnica prometedora dentro de la tendencia actual en clasificación por los mejores resultados que ofrece. Asimismo en la preparación de los datos, con la finalidad de definir una óptima estrategia de prueba, se han utilizado tanto el algoritmo de estratificación, con el fin de heterogeneizar los conjuntos de datos, como las técnicas basadas en validación cruzada "cross-validation" para dividir dichos datos. Con estas técnicas se pretende mejorar los resultados obtenidos por los tres métodos de clasificación clásicos Agrupamiento borroso, Bayes y Vecinos más cercanos (nearest-neighbours).

Palabras clave: clasificación, lógica Fuzzy, clasificador Fuzzy, clasificador Bayesiano, clasificador de nearest-neighbours (vecinos más cercanos), estratificación, ten fold cross-validation (validación cruzada).

Agradecimientos

A María Guijarro y a Gonzalo Pajares, por facilitarme ampliamente la información utilizada, y por su disposición para ayudarme en todo momento con este proyecto.

Abstract

This work appears as a comparative study between a great variety of classifiers used in the data mining. There are diverse methods applied by the different classifiers, nevertheless, this work does not finish since every classifier offers proved with a limited number of classes. The aim of this work consists of stating that nowadays there does not exist any model who allows the classification of any set of samples in view of, and that simultaneously obtains a few satisfactory results. At the same time in this work we have proposed a new model of classification who improves some results on having compared them with the best results offered by other classifiers, and though it is far from being a widespread classifier, the combination of classifiers appears as a promising technology inside the current trend in classification for the best results that it offers. Likewise in the preparation of the information, with the purpose of defining an ideal strategy of test, it have been used the algorithm of stratification, in order to divide the sets of information as the technologies based on crossed validation, for dividing the above mentioned information. With these technologies one tries to improve the results obtained by three classic methods of classification Fuzzy Clustering, Bayes and Nearest Neighbors.

Keywords: classification, Fuzzy logic, Fuzzy classifier, Bayesian classifier, K-nearest-neighbours, stratification, ten fold cross-validation.

INDICE

Capítulo 1

1. Introducción

1.1. Identificación del problema

En la búsqueda por encontrar un clasificador generalizable que pueda utilizarse con cualquier conjunto de datos o muestras, se ha experimentado ampliamente en el tema, según se deduce del estudio bibliográfico realizado; esto ha dado lugar a un elevado número de clasificadores que con sus ventajas e inconvenientes, ninguno a nivel individual, ha sido capaz de convencer a los expertos para utilizarlo como el más prometedor de forma generalizada. En la búsqueda por mejorar el porcentaje de acierto por parte de cada clasificador, el problema encontrado se centra en predecir a qué clase pertenecen las muestras evaluadas, así como en reducir el tiempo de ejecución ante cualquier proceso relacionado con la minería de datos. Este es el objetivo de la clasificación y puede llevarse a cabo de múltiples maneras.

Langley y Simon (1998) opinaban en su artículo que "los objetivos de las máquinas de aprendizaje eran proporcionar un incremento del nivel de autonomía en los procesos de ingeniería del conocimiento con técnicas automáticas que mejorasen los porcentajes de acierto y la eficiencia, por medio del descubrimiento y el uso de reglas sobre los datos. El último test de una máquina de aprendizaje es su habilidad para producir sistemas que sean usados regularmente en la industria, en la educación y en cualquier otro campo". En su estudio describían cómo muchos casos en la vida cotidiana se resolvían mediante reglas o árboles de decisión:

- Incrementar la productividad en procesos químicos de control.
- Diagnóstico de dispositivos mecánicos.
- Análisis de fallos en redes eléctricas.
- Intuir las bases de conocimiento necesarias para diagnosticar fallos en circuitos integrados.
- Análisis de imágenes.

- Analizar materiales y hacer sus ordenaciones cronológicas.
- Análisis de sistemas de seguridad y casos similares.

Considerando que las reglas generalmente no funcionan bien en tareas de aprendizaje complejas, algunos clasificadores utilizados para llevar a cabo dichas tareas son el ID3 o el C4.5, que mediante árboles de decisión terminan aplicando reglas de clasificación basándose en la entropía para obtener la información necesaria.

Existen muchos tipos de clasificadores que pueden llegar a estas conclusiones, otra forma es mediante modelos matemáticos o teóricos, por ejemplo los clasificadores estadísticos que son construidos a partir de la teoría de decisión de Bayes, la cual ofrece un modelo de clasificación que minimiza la probabilidad total de error.

Otra aproximación conocida son las redes neuronales, que aplican capas a un conjunto de muestras en la entrada y éstas dan lugar a otro conjunto de salidas ya entrenadas. Suele ofrecerse un vector de entrada para obtener otro de salida, que será asignado a la clase más significativa. Métodos basados en distancias como el Clasificador basado en máxima probabilidad (Maximum likelihood classifier, MLC), o el de Vecinos más cercanos (K-Nearest Neighbours, K-NN), evalúan las distancias de los vectores de los objetos en la entrada y los clasifican en clases individuales asociados por la mínima distancia.

Asimismo los algoritmos no evolutivos reducen el tiempo de búsqueda respecto a los evolutivos. Además los algoritmos evolutivos, usen estratificación o no, nunca mejoran el rendimiento del algoritmo Leave-one-out (LOOV) cuando se combina con los K-Vecinos más cercanos (K-NN), como se menciona en el artículo de Cano y col. (2004).

También existen clasificadores que combinan múltiples clasificadores de decisión para obtener el mayor porcentaje de acierto buscado (Stefanowski, 2004).

Como puede verse la literatura es rica en métodos de clasificación aplicados a la minería de datos, es decir al tratamiento masivo de los datos. Se deduce en una primera aproximación, que en lugar de tratar de unificar en un método que abarque las ventajas de todos ellos, la tendencia es generar más métodos buscando únicamente mejorar algún aspecto.

Los retos tecnológicos derivados de las aplicaciones que utilizan reglas para su correcto funcionamiento, hacen que en algunos casos la utilización de las herramientas mencionadas anteriormente sea insuficiente, para abordar las propuestas de proyectos demandados por los clientes donde la clasificación de las muestras surge como una tarea fundamental.

Se ha comprobado que la estratificación disminuye considerablemente el tiempo que se necesita para evaluar un conjunto de muestras de datos, además de mejorar la escalabilidad. Los conjuntos de datos que utilizan la estratificación también mejoran ligeramente los resultados respecto a las muestras analizadas sin estratificación.

La estratificación tan sólo es una estrategia de ordenación de los datos de entrada que debe ser aplicada con alguno de los métodos de clasificación.

Además, y lo que es más importante, en muchos casos no existe la posibilidad de llevar a cabo la investigación necesaria para abordar la problemática, particularmente cuando los métodos clásicos no producen los resultados esperados.

Por todo lo expuesto anteriormente, surge una necesidad importante en el ámbito de las aplicaciones reales para afrontar el tema de la clasificación de datos, y un reto para la comunidad científica para tratar de mejorar los procedimientos existentes con la mayor flexibilidad posible.

Esta memoria se organiza en cinco capítulos. En lo que resta del presente capítulo se exponen los objetivos que se plantearon en la investigación aquí recogida, que tratan de dar solución a la problemática existente. También se exponen los métodos estudiados a través de múltiples artículos que tratan de resolver el problema de la clasificación. En el capítulo 2 se realiza una revisión del estado del arte en el tema de clasificación de datos, y se realiza una explicación detallada de los métodos y técnicas elegidos. En el capítulo 3 se propone la estrategia de solución a la problemática planteada. En el capitulo 4 se diseña una estrategia de pruebas con el objetivo de verificar la eficacia del método propuesto cuando se compara con otras estrategias existentes. Finalmente en el capítulo 5 se extraen las conclusiones pertinentes y se exponen las líneas de investigación futuras, que dan pie al inicio de la investigación en los estudios de Doctorado.

1.2. Objetivos de investigación

1.2.1. Objetivos generales

1) Aprender a manejar referencias bibliográficas, así como la forma de abordar las investigaciones.

2) Identificar los métodos de clasificación existentes en la literatura.

3) Determinar los métodos más utilizados.

4) Analizar las ventajas e inconvenientes de cada uno de ellos.

5) Identificar las técnicas de clasificación que mejoran el funcionamiento de los distintos métodos.

6) Analizar las ventajas e inconvenientes de cada una de ellas.

7) Realizar un aporte de carácter investigador con las conclusiones finales.

1.2.2. Objetivos específicos

1) Seleccionar al menos dos métodos de clasificación de entre los más prometedores o apropiados que se hayan estudiado.

2) Seleccionar las bases de datos que constituirán los conjuntos de entrenamiento y de clasificación.

3) Identificar una posible mejora de los métodos existentes o dar una solución novedosa a la problemática de la clasificación en la minería de datos.

4) Implementar la mejora propuesta con el fin de analizar su comportamiento en base a los resultados obtenidos.

5) Identificar líneas de investigación futuras.

1.3. Metodología histórica

A continuación se exponen de forma cronológica los pasos seguidos para llevar a cabo la investigación, junto con la actividad desarrollada en cada uno de ellos.

Se inicia el trabajo de investigación porque la clasificación de datos aparece como un reto que se aborda constantemente, como se comprueba a lo largo de la bibliografía, con aportaciones significativas pero no concluyentes. Este trabajo intentará incorporar

importantes avances en la clasificación y reconocimiento de los datos presentados, para predecir comportamientos y mejorar la identificación de problemas.

El ser humano puede razonar y tomar decisiones a partir de información que raramente es precisa y que muchas veces puede ser modelada por generalizaciones del tipo "modus ponens" clásico. Como consecuencia de ello, se inicia el estudio de los métodos ya implementados con el fin de mejorar los resultados obtenidos por los métodos existentes en la bibliografía.

Actualmente se están estudiando teorías orientadas a la fusión de la información proporcionadas por varios métodos. Mediante agregación borrosa son diversos los contextos en los que se obtiene información útil a partir de datos incompletos, imprecisos o inciertos.

En base a lo anterior, el siguiente paso consiste en la identificación de aquellos métodos simples existentes en la literatura con resultados satisfactorios, para su análisis y el diseño de las estrategias encaminadas hacia la mejora de resultados.

1.4. Clasificadores utilizados en el estudio comparativo

La clasificación se lleva a cabo de forma diferente según el método utilizado, sin embargo todos parten de un mismo conjunto de datos, y todos terminan haciendo una predicción de su porcentaje de acierto. Durante el estudio se han extraído las características principales que marcan el comportamiento de los distintos algoritmos existentes.

1.4.1. Agrupamiento borroso (Fuzzy Clustering)

El método de Agrupamiento borroso es también conocido como Fuzzy Clustering en terminología inglesa. En este trabajo nos referiremos indistintamente a los términos borroso y fuzzy sin ninguna distinción. Su funcionamiento básico consiste en dividir primero el conjunto de muestras en clases, luego aparece una fase de optimización para reducir el número de parámetros utilizados en cada clase, y finalmente combina las clases locales obteniendo una solución final que encuentra una mínima distancia. Esta parte se puede realizar de varias formas. Algunos métodos intentan utilizar una heurística que genera el mínimo conjunto de reglas de decisión para cada clase

(Stefanowski, 2004), si bien, cuando existen atributos que no son significativos, aumenta sustancialmente la complejidad del problema. Otros buscan la mínima distancia mediante un algoritmo teórico-gráfico, obteniendo el camino más corto mediante el método de Dijkstra o alguna de sus variantes (Takagi y col., 2004).

Otro planteamiento mediante la estrategia fuzzy consiste en cubrir con cada regla una región concreta, que irá en el antecedente de un atributo mientras que en el consecuente se maximiza la calidad de esa regla. Esto hace que la asignación en el consecuente determine para cada regla la clase mayoritaria correspondiente (Alatas y Akin, 2005), sin embargo este método ofrece varios problemas, uno de ellos es que requiere que los datos estén adaptados convenientemente para que la función heurística pueda ser utilizada. Otro problema es que necesita utilizar una regla que haga de transición entre otras reglas, pues la probabilidad de asignar una condición a una regla equivocada es elevada, esto y las actualizaciones constantes de las condiciones ralentizan considerablemente el proceso, llegando a producir pérdidas de información.

En contraposición al planteamiento anterior, en Fakhrahmad y col. (2007) sugieren impedir generar todas las reglas posibles con respecto a todas las combinaciones de antecedentes para mejorar la interpretabilidad, de forma que elaboran su clasificador mediante la generación de reglas con distinto peso específico. Es posible que los resultados no sean mucho mejores porque fácilmente producirá la pérdida de datos correspondientes a una clase minoritaria.

1.4.2. Árboles de decisión Fuzzy

Un árbol de decisión Fuzzy (Fuzzy Decisión Tree, FDT) es en sí mismo un árbol de decisión que crea un algoritmo de generación de reglas Fuzzy, compuesto por nodos y arcos que representan sus atributos y sus valores respectivamente. La principal diferencia con respecto a los árboles de decisión clásicos, es que en el árbol de decisión Fuzzy cada arco está asociado a un término lingüístico Fuzzy, donde cada ramificación representa una clase, ésta es asociada a un factor de certeza que indica lo acertado que es continuar o no por esa rama, y en función de su elección elige la rama por la que progresar. Un árbol de decisión común evalúa tan sólo una regla y elige esa rama o no, sin embargo el FDT toma su decisión detrás la evaluación de múltiples reglas. Por eso este método es mucho más poderoso y eficiente que cualquier árbol de decisión clásico.

Un estudio comparativo respecto de este tipo de árboles, frente otros árboles de decisión, se realiza en Kim y Ryu (2005). Uno de los modelos con los que se compara utiliza árboles de decisión divididos en su síntesis de prueba con la técnica “ten-fold cross-validation”. Sin embargo el FDT muestra mejores resultados que cualquier otro árbol de decisión, tanto en porcentaje de acierto como en tamaño del árbol.

Otra variante de los FDT consiste en clasificar los datos en tres fases; una inicial que crea el árbol de decisión, la segunda que lo transforma en un modelo fuzzy y la tercera, que optimiza todos los parámetros (Tsipouras y col., 2008). Esta técnica permite clasificar cualquier conjunto de datos, no obstante los resultados que ofrecen no son mejores que los de otros modelos con los que se ha comparado.

Aunque sus resultados son prometedores mejorando los resultados del C4.5, que ha sido el mejor de los algoritmos de inducción entre los árboles de decisión clásicos, no llegan a determinar automáticamente las funciones para cada atributo, esto hace que las predicciones de acierto sean malas, además de que producen pérdidas de información que afectan considerablemente al porcentaje de acierto.

1.4.3. Meta-conocimiento

Se trata de otra técnica para la clasificación de datos. En el aprendizaje se pasa por distintos procesos, acumulando experiencia mientras explota al máximo sus recursos durante una exhaustiva reestructuración de sus parámetros para mejorar el rendimiento. Una vez adquirido el conocimiento, el aprendizaje adquirido lo utiliza para llevar a cabo el meta-entrenamiento de las muestras basado en reglas Fuzzy. Cuando ha finalizado el proceso podrá asignarse correctamente cualquier dato como se menciona en Castiello y col. (2008). Desgraciadamente seguimos dependiendo de la configuración manual de los parámetros para que el funcionamiento sea correcto.

1.4.4. Redes Neuronales

Para resolver la tarea de clasificación en minería de datos mediante redes neuronales, se utiliza a menudo el algoritmo de Red Neuronal Polinomial (Polinomial neural Network, PNN). Este algoritmo requiere un elevado tiempo de computación porque la red crece considerablemente durante el entrenamiento, por lo que se necesita mucha memoria y velocidad de procesamiento. A su vez aplica descripciones parciales de las muestras

para hacer distinciones entre las capas, que almacenan neuronas incluyendo las características de cada muestra. Como las descripciones parciales en cada capa crecen sucesivamente y los atributos cambian constantemente, la nueva propuesta que se hizo en Misra y col. (2008) fue utilizar en el entrenamiento 2-fold cross-validation, entrenando 2/3 de las muestras y clasificando 1/3. Esto mejoró los resultados respecto a sus predecesores en redes neuronales, pero no lo suficiente para compararse con otros clasificadores como Fuzzy o árboles de decisión.

Otra variante es utilizar técnicas de mejora del rendimiento como en Dam y Abbass (2008), donde se aplica NCL (Negative Correlation Learning) durante el entrenamiento de la red, dando lugar en las salidas a una sub-clase por cada neurona, y la clase principal es designada por el nodo de mayor actividad. Este método es sobretodo rápido, sin embargo su porcentaje de acierto es mejor que el resto de los comparados sólo en algunas ocasiones.

1.4.5. Neuro-Fuzzy

Neuro-Fuzzy es un sistema híbrido que combina la capacidad de aprendizaje de las redes neuronales con la interpretabilidad del sistema fuzzy.

El HNFB invertido (Inverted Hierarchical Neuro-Fuzzy BSP System) (Gonçalves y Vellasco, 2006) se denomina así porque aplica el proceso de aprendizaje HNFB para generar su modelo de estructura. Cada partición del espacio tendrá una regla Fuzzy asociada que generará *k* subreglas, y cada subregla tendrá un nivel que será determinado por el porcentaje de acierto y por la convergencia fuzzy. El rendimiento en tiempo es bueno, sin embargo se ha comparado sólo con cuatro bases de datos, lo que deja el estudio con escaso muestrario de resultados.

1.4.6. Hipermutaciones Fuzzy

Existe también un proceso que utiliza lo que se conoce como hipermutación e inmunización. Los individuos iniciales adquieren aptitudes mediante una función objetivo y los que mejores resultados ofrecen se clonan hasta que estos resultados varían un umbral mínimo y el proceso se detiene. En ese momento se produce una mutación que permite adquirir nuevas aptitudes, y estas generarán resultados distintos en busca de

mejorar los resultados anteriores (Castro y col., 2005). Este proceso ofrece elevados porcentajes de acierto, aunque produce graves problemas de escalabilidad y de tiempo.

1.4.7. Programación Genética (GP)

Esta técnica la introdujo Koza en 1987 (Cordella y col., 2005). Permite establecer relaciones entre datos y se usa para generar árboles de decisión mediante prototipos, que abarcará un conjunto de predicados como condiciones.

Encuentra automáticamente todas las subclases que se presentan en el conjunto de datos. Cada expresión representará una clase o una subclase del problema. Este método trabaja según el paradigma evolutivo. Los algoritmos genéticos codifican un conjunto de reglas de clasificación que luego son ordenadas en cadenas de bits llamadas genes. Algunas operaciones como la reproducción y mutaciones se utilizan para obtener mejores resultados. Para la programación genética se aplican dos tipos de clasificadores de aprendizaje en el entrenamiento de los datos, el primero está basado en las reglas de clasificación con redes neuronales, y el segundo en funciones que incluyen un discriminante que indica si un objeto pertenece o no a una determinada clase. La ventaja de este segundo método es que su eficiencia aumenta porque cada clase sólo encaja con una función, y estas funciones resultan más fáciles de definir.

Una desventaja es que hay un problema de ambigüedad debido a las funciones discriminantes cuando tienen que clasificar la pertenencia de una muestra a dos clases muy próximas, o cuando no encaja con ninguna. El otro problema grave es que necesita más tiempo del deseado para su entrenamiento, aunque el resultado final sea de un elevado acierto.

Otras pruebas realizadas para mejorar la eficiencia tanto en porcentaje de acierto como en tiempo, es la combinación de un algoritmo evolutivo y un buscador local (García y col., 2008), que incluyen una o varias fases de búsqueda sin recaer en ciclos. Estos algoritmos también podrían ser hibridados con otros clasificadores de baja complejidad como 1-NN para obtener un mejor comportamiento. El fallo principal de los algoritmos evolutivos, es que a cambio de poder analizar muestras muy grandes cuando estas incurren en problemas de escalabilidad, tienden a desocuparse del porcentaje de acierto y se centran en la reducción del tiempo de análisis.

1.4.8. Vecinos más cercanos

Es uno de los algoritmos más simples y atractivos en la clasificación de patrones. Realiza una estimación a posteriori de la probabilidad sobre la cercanía de sus vecinos. Para la clasificación de una muestra toma como referencia los valores de los *k* vecinos más próximos mediante distancias, generalmente Euclídeas, aunque en el método propuesto por Wang y col. (2005), utiliza la influencia que ha podido producir cada muestra en el entrenamiento mediante notaciones estadísticas. Se establecen las *k* distancias de menor valor a los *k* vecinos, finalmente se elige la clase a la que pertenezca el mayor número de los *k* vecinos involucrados.

1.4.9. Redes neuronales y Bayes

Utiliza una estructura Bayesiana para inferir los pesos y encontrar el modelo neuronal con mejores capacidades y menos complejidad. Los resultados obtenidos con el estándar MLP (clasificador basado en perceptrón multicapa) (Tsipouras y col, 2008; Saastamoinen y Ketola, 2006; Dam y Abbass, 2008; Castro y Von Zuben, 2006) son similares a los de las redes neuronales Bayesianas, con el inconveniente de que la asignación de neuronas es mayor. En Castro y Von Zuben (2006) se muestra un diseño basado en redes MLP, con la alternativa de una función de activación automática para cada neurona. Algunos aspectos se pueden mejorar tales como la extensión del algoritmo para incluir la arquitectura de otra red neuronal, aunque su principal desventaja es que el ajuste de los parámetros es determinante en el rendimiento del clasificador, significando que en otras bases de datos diferentes a las utilizadas en sus experimentos su eficiencia puede disminuir drásticamente.

1.5. Tendencias actuales en combinación de clasificadores

Recientemente se han publicado los trabajos que a continuación se relacionan en el área de la combinación de clasificadores, todos ellos reflejados en Guijarro (2009):

1. En Guijarro y Pajares (2009) se propone un esquema de clasificación no supervisada a partir del clasificador conocido como Agrupamiento Borroso de naturaleza supervisada. En su conjunto, se trata de un método de índole local, de

suerte que cada píxel se clasifica atendiendo a los criterios proporcionados por seis clasificadores individuales.

2. En Pajares, Guijarro y col. (2009) se propone un esquema de clasificación no supervisada de naturaleza global mediante el esquema de optimización basado en un proceso de *enfriamiento simulado*. Cada píxel se clasifica teniendo en cuenta su propia naturaleza, así como la influencia ejercida por los píxeles vecinos.

3. En Guijarro y col. (2008) se propone igualmente una estrategia de naturaleza no supervisada con el mismo esquema de la anterior. En este caso se combinan dos clasificadores individuales bajo una perspectiva global utilizando la información de la vecindad de un píxel en la imagen. El proceso global consiste en un procedimiento de relajación iterativo hasta conseguir el máximo grado de estabilización.

4. En Guijarro y col. (2007*a*) el procedimiento propuesto es de naturaleza local y consiste en la combinación del método de agrupamiento borroso con el método no paramétrico de la ventana de Parzen. La mencionada combinación se lleva a cabo bajo el paradigma de la probabilidad condicionada de Bayes. Mediante el clasificador de agrupamiento borroso se obtiene la probabilidad a priori, mientras que el estimador de la ventana de Parzen proporciona la probabilidad a posteriori. El método es de naturaleza no supervisada basado en un esquema similar a los mencionados en los puntos anteriores.

5. En Guijarro y col. (2007*b*) se presenta una estrategia de naturaleza local, que utiliza para la combinación el método de agrupamiento borroso y el método de cuantización vectorial. El planteamiento vuelve a ser de naturaleza no supervisada, de forma que el primero establece una partición inicial con los datos disponibles, la cual es transferida al segundo y mejorada hasta conseguir su validación.

6. En Guijarro y col. (2007*c*) se establece un planteamiento local similar al propuesto en el punto cuatro con la diferencia de que en lugar del método de agrupamiento borroso se utiliza el estimador de máxima verosimilitud. El

esquema no supervisado sigue la misma estrategia que en el caso del trabajo del punto cuatro.

1.6. Aportaciones a la investigación

De entre todos los métodos revisados en la literatura, los que mejores resultados ofrecen son Vecinos más cercanos, Bayes y Fuzzy. Se ha optado por seleccionarlos para este trabajo de investigación cuyo objetivo principal consiste en mejorar su rendimiento desde el punto de vista de eficacia en el porcentaje de acierto. Como se ha mencionado previamente, los términos *borroso y Fuzzy* se utilizarán indistintamente a lo largo de esta memoria, al igual que *Vecinos más cercanos Fuzzy* y *Nearest Neighbours Fuzzy*.

Un aspecto importante a destacar en el tema de la clasificación es que ningún algoritmo que se haya aplicado hasta el momento, es capaz de clasificar *cualquier conjunto* de muestras teniendo éxito en el porcentaje de aciertos, lo que significa que esta labor aún está pendiente. Este es el punto de inflexión entre lo que actualmente conocemos por modelos de clasificación, y un clasificador que pueda utilizarse de forma general para todos los conjuntos de datos a evaluar.

Se ha observado el funcionamiento de algunos métodos que optimizan la división de muestras del conjunto inicial facilitando su posterior aprendizaje, la *estratificación* y el *n-fold cross-validation* son dos de los candidatos que abordan esta problemática.

Se han utilizado las técnicas de clasificación combinada *máximo*, *mínimo*, *mediana* y *votación mayoritaria* (“*majority voting*”), que dan más consistencia a los resultados obtenidos y eligen el mejor clasificador en cada momento.

En el caso que nos ocupa se aprecia, aunque no siempre, una mejora sustancial de los resultados con la combinación entre los métodos de estudio y las técnicas utilizadas, respecto de los obtenidos con cada clasificador de forma individual. Sin embargo estos resultados son el producto de un ajuste de parámetros, que mejora el acierto en algunas bases de datos y ofrece resultados pobres en otras, por lo que surge la necesidad de buscar nuevas soluciones que mejoren los resultados de cualquier base de datos sin que éstos dependan de una parametrización concreta.

Queda todavía por resolver un aspecto relacionado con el procedimiento de clasificación automático que se pretende desarrollar. Los métodos de clasificación seleccionados tienen el inconveniente de que son de naturaleza supervisada, en el sentido de que es preciso realizar una inicialización de forma dirigida sobre el número de clases de cada conjunto de muestras, junto con las muestras asociadas a cada clase.

Los resultados obtenidos se han comparado con los que proporcionan los métodos individuales, y también con los derivados de la combinación de dichos métodos con alguna estrategia de clasificación, verificando que el método propuesto consigue más estabilidad que los extraídos de la bibliografía, siendo este el principal objetivo del trabajo.

Como resumen de todo lo anterior las aportaciones de investigación realizadas en el presente trabajo son las siguientes:

1) Diseño de un clasificador supervisado basado en tres métodos de clasificación clásicos.

2) Combinación de los métodos de clasificación con técnicas para mejorar el rendimiento, la eficiencia y la velocidad del proceso.

3) Aplicación de recursos que facilitan la división de las muestras iniciales en subconjuntos, para mejorar su entrenamiento y su posterior clasificación.

4) Estudio comparativo del método propuesto frente a otros métodos de clasificación, tanto combinados como aislados.

Capítulo 2

2. Revisión de métodos utilizados

2.1. Introducción

De los múltiples métodos estudiados en la literatura, que ofrecen resultados favorables en su combinación entre tiempo de ejecución y porcentaje de acierto, se ha creído conveniente investigar los siguientes:

- Clasificador Bayesiano.
- Agrupamiento borroso (Pajares y Cruz, 2002; Pajares y col. 2002).
- Vecinos más cercanos fuzzy (Nearest Neighbours Fuzzy).

Todas las bases de datos con las que se ha trabajado han sido descargadas del repositorio de Internet UCI Machine Leraning Repository (UCI MLR) Asunción y Newman (2009), que contiene cientos de bases de datos para realizar pruebas con algoritmos de clasificación.

Dentro del campo de la minería de datos, un avance reciente es el hecho de que la ***combinación*** de clasificadores obtiene mejores resultados que los clasificadores utilizados de forma individual (Valdovinos y col. 2005; Kuncheva, 2004; Kumar y col. 2002; Kittler y col. 1998). Existen diferentes estudios relativos al tema de la combinación donde se destacan las ventajas de este planteamiento, en el sentido de que la combinación permite resaltar las bondades de los clasificadores al mismo tiempo que se atenúan o desaparecen las desventajas (Partridge y Griffith, 2002; Deng y Zhang 2006).

Para llevar a cabo la selección, cada clasificador individual proporciona una decisión y se elige un único clasificador como el mejor de acuerdo a alguna estrategia preestablecida. Sin embargo, en la fusión los clasificadores se combinan utilizando diferentes estrategias (Kuncheva, 2004; Grim y col. 2002; Kittler y col. 1998; Duda y Hart, 2000). Dentro de estas estrategias se incluyen las denominadas: votación mayoritaria, reglas de Máximo, Mínimo o de la Mediana.

Se entiende por característica alguna propiedad que describa el contenido de la muestra (Valdovinos y col. 2005; Puig y García, 2006; Hanmandlu y col. 2004). Podemos decir que cada dato está compuesto por un vector de características, donde la principal propiedad es la clase a la que pertenece la muestra, puesto que ésta identifica las muestras que llegan a la entrada en un grupo u otro.

2.2. Descripción de los clasificadores seleccionados

Para llevar a cabo el estudio y las pruebas que se han realizado al objeto de verificar el comportamiento de nuestra estrategia, a continuación se describen de forma exhaustiva, los tres clasificadores clásicos mencionados en el primer apartado, que constituyen la base de dicha estrategia y han sido ampliamente probados y verificados en la literatura.

2.2.1. Agrupamiento borroso

El objetivo de la técnica de clasificación conocida como *Agrupamiento borroso o Fuzzy Clústering* consiste en dividir *n* objetos $x \in X$ caracterizados por *p* propiedades en *c* clústeres o grupos. Supongamos el conjunto de datos $X = \{x_1, x_2, ..., x_N\} \in \Re^p$ un subconjunto del espacio real *p*-dimensional $\Re^p$. Cada $x_k = \{x_{k_1}, x_{k_2}, ..., x_{k_p}\} \in \Re^p$ se denomina vector de características, x_{k_j} es la *j*-ésima característica de la observación x_k.

Este clasificador puede encontrarse perfectamente especificado en las siguientes referencias clásicas: Bezdek (1981), Duda y Hart, (2000) o Zimmermann (1991).

Puesto que los elementos de un clúster deben ser tan similares entre sí como sea posible y a la vez deben ser tan diferentes a los elementos de otros clústeres como también sea posible, el proceso se controla por el uso de medidas de similitud basadas en distancias. Así la similitud o la diferencia entre dos puntos x_k y x_l puede interpretarse como la distancia entre esos puntos.

Una distancia entre dos objetos de un universo X es una función $d : XxX \rightarrow R^+$ que toma valores reales que denotamos $d(x_k, x_l) = d_{kl} \geq 0$ y que cumple tres propiedades:

1. $d_{kl} = 0 \Leftrightarrow x_k = x_l$ para todo 0≤ k, l ≤ N
2. Simetría: $d_{kl} = d_{lk}$ para todo 0≤ k, l ≤ N

3. Desigualdad triangular $d_{kl} \leq d_{kj} + d_{jl}$ para todo 0≤ k, j , l≤ N

Cada partición del conjunto $X = \{x_1, x_2, ..., x_n\}$ puede enfocarse desde dos perspectivas; fuzzy y no fuzzy. Una partición no fuzzy se conoce en terminología inglesa como crisp. Si se desea realizar una partición del conjunto *X* en *c* clústeres tendremos $S_i\{i = 1, ..., c\}$ subconjuntos.

Para cada partición, S_i define un conjunto borroso μ_i sobre el universo X, $\mu_i : X \rightarrow [0,1]$ que asigna lo que se conoce como grado de pertenencia μ_{ik} de cada objeto x_k al subconjunto S_i. Así pues denotamos dicho grado de pertenencia del elemento x_k al clúster S_i como $\mu_i(x_k) = \mu_{ik}$. En el caso de conjuntos crisp un objeto x_k se dice que pertenece a un S_i dado y no pertenece al resto. Esto se expresa con los valores discretos {0,1} de la siguiente forma $\mu_{ik} = 1$ para indicar que pertenece y $\mu_{ik} = 0$ para expresar que no pertenece. Por el contrario, en el caso de conjuntos fuzzy se dice que un objeto puede pertenecer a diferentes subconjuntos y así se habla por ejemplo de que x_k pertenece a un conjunto S_i con grado de pertenencia μ_{ik} y a S_j con grado de pertenencia μ_{jk}. Como ejemplo, supongamos que se tienen tres conjuntos S_i, S_j y S_h, en este caso podríamos decir que el objeto x_k pertenece a los conjuntos con los siguientes grados de pertenencia $\mu_{ik} = 0.4$, $\mu_{jk} = 0.5$ y $\mu_{hk} = 0.1$. Los valores tomados pertenecen al intervalo continuo [0,1].

Dado $X = \{x_1, x_2, ..., x_n\}$ y el conjunto V_{cn} de todas las matrices reales de dimensión *c x n*, con $2 \leq c < n$. Se puede obtener una matriz representando la partición de la siguiente manera $U = \{\mu_{ik}\} \in V_{cn}$. Tanto en el supuesto crisp como en el fuzzy se deben cumplir las siguientes condiciones:

1) $\mu_{ik} \in \{0,1\}$ *crisp* o $\mu_{ik} \in [0,1]$ *fuzzy* $\quad 1 \leq i \leq c;\ 1 \leq k \leq n$

2) $\sum_{i=1}^{c} \mu_{ik} = 1 \qquad 1 \leq k \leq n$

3) $0 < \sum_{k=1}^{n} \mu_{ik} < n \qquad 1 \leq i \leq c$

Para ilustrar los conceptos anteriores, sea $X = \{x_1, x_2, x_3\}$. Con él podríamos construir las siguientes particiones teniendo en cuenta que para el ejemplo utilizaremos un valor de $c = 2$.

"crisp"	*"fuzzy"*
$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$	$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0.3 & 0.5 & 0 \\ 0.7 & 0.5 & 1 \end{bmatrix}$
$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0.9 & 0.4 & 0.2 \\ 0.1 & 0.6 & 0.8 \end{bmatrix}$

La localización de un clúster S_i se representa por su centro $v_j = \{v_{j_1}, v_{j_2}, ..., v_{j_p}\} \in \Re^p$ con $j = 1,\ldots,c$, alrededor del cual se concentran los objetos.

La definición básica de llevar a cabo el problema de la partición fuzzy para $m > 1$ y siendo esta el peso exponencial, consiste en minimizar la función objetivo según la ecuación (2.1):

$$\min z_m(U; v) = \sum_{k=1}^{n} \sum_{j=1}^{c} \mu_{jk}^m \left\| x_k - v_j \right\|_G^2 \qquad (2.1)$$

donde G es una matriz de dimensión $p \times p$ que es simétrica y definida positiva. Así se puede definir una norma general del tipo,

$$\left\| x_k - v_j \right\|_G^2 = \left(x_k - v_j\right)^t G \left(x_k - v_j\right) \qquad (2.2)$$

Diferenciando la función objetivo para v_i (suponiendo constante U) y μ_{jk} (suponiendo constante v) y aplicando la condición de que $\sum_{j=1} \mu_{jk} = 1$, se obtiene,

$$v_j = \frac{1}{\sum_{k=1}^{n} \left(\mu_{jk}\right)^m} \sum_{k=1}^{n} \left(\mu_{jk}\right)^m x_k \qquad j = 1,\ldots,c \qquad (2.3)$$

$$\mu_{jk} = \frac{\left(\frac{1}{\left\| x_k - v_j \right\|_G^2} \right)^{2/m-1}}{\sum_{h=1}^{c} \left(\frac{1}{\left\| x_k - v_h \right\|_G^2} \right)^{2/m-1}} \qquad j = 1,...,c; k = 1,...,n \tag{2.4}$$

donde *m* se conoce como peso exponencial, que permite disminur la influencia del ruido al obtener los centros de los clústeres, reduciendo la relevancia de los valores pequeños de μ_{jk} (puntos lejanos a v_j) frente a valores altos de μ_{jk} (puntos cercanos a v_j). Cuanto mayor sea el valor de *m*, mayor será dicha influencia.

2.2.2. Clasificador Bayesiano

La clasificación mediante el algoritmo Bayesiano ofrece la solución óptima al problema, incluyendo la característica fuzzy de la probabilidad de pertenencia de cada muestra a todas las clases. Para la evaluación de esta regla se requiere un conocimiento a priori de la probabilidad y de la densidad de las clases.

En el caso general y más típico de una distribución de probabilidad Gaussiana o Normal multivariable, ni la media **m** ni la matriz de covarianza *C* son conocidas. Por tanto, esos parámetros desconocidos constituyen las componentes del vector de parámetros $\boldsymbol{w} = \{\boldsymbol{m}, C\}$. Consideremos el supuesto univariable con $\boldsymbol{m} = m$ y $C = \sigma^2$, en cuyo caso

$$\ln p(x_i/\boldsymbol{w}) = -\frac{1}{2}\ln 2\pi C - \frac{1}{2C}(x_i - m)^2 \tag{2.6}$$

$$\nabla_{\boldsymbol{w}} \ln p(x_i/\boldsymbol{w}) = \begin{bmatrix} \frac{1}{C}(x_i - m) \\ -\frac{1}{2C} + \frac{(x_i - m)^2}{2C^2} \end{bmatrix} \tag{2.7}$$

La minimización sobre los datos de entrenamiento conduce ahora a las condiciones,

$$\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\hat{C}}(x_i - \hat{m}) = 0 \qquad -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{2\hat{C}} + \frac{1}{n}\sum_{i=1}^{n}\frac{(x_i - \hat{m})^2}{2\hat{C}^2} = 0 \tag{2.8}$$

donde $\hat{m}$ y $\hat{C}$ son las estimas de máxima verosimilitud para *m* y *C*, respectivamente. Sustituyendo $\hat{m}$ y $\hat{\sigma}^2 = \hat{C}$ obtenemos las estimas de máxima verosimilitud para *m* y σ^2.

$$\hat{m} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - m)^2 \tag{2.9}$$

Aunque el análisis del caso multivariable es básicamente muy similar, se requiere mucha más manipulación. El resultado muy bien conocido en estadística, es que las estimas de máxima verosimilitud para **m** y *C* están dadas por

$$\mathbf{m} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \qquad C = \frac{1}{n-1}\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^{\mathrm{t}} \tag{2.10}$$

La expresión (2.10) nos dice que la estima de máxima verosimilitud para el vector media es la media simple. La estima de máxima verosimilitud para la matriz de covarianza es la media aritmética de las *n* matrices $(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^{\mathrm{t}}$. Puesto que la verdadera matriz de covarianza es el valor esperado de la matriz $(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^{\mathrm{t}}$, se obtiene un resultado muy satisfactorio.

Una vez estimados los parámetros **m** y C, la función de densidad de probabilidad queda perfectamente especificada por la ecuación dada en 2.11 suponiendo que dicha función sigue una distribución Gaussiana.

$$p(\mathbf{x} \mid \mathbf{m}, \mathrm{C}) = \frac{\mathbf{1}}{(2\pi)^{d/2}|C|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^t C^{-1}(\mathbf{x}-\mathbf{m})\right\} \tag{2.11}$$

Según la teoría general de la probabilidad de Bayes, dado ***x*** el objetivo que se plantea es asignar ***x*** a alguna de las clases existentes. Supongamos que las clases son $S_1, S_2, \ldots, S_c$ para cada una de ellas se puede estimar la función de densidad de probabilidad dada en (2.11) obteniendo ($\boldsymbol{m}_1, C_1$) para la clase S_1, ($\boldsymbol{m}_2, C_2$) para la clase S_2 y así sucesivamente hasta llegar a la clase S_c con ($\boldsymbol{m}_c, C_c$).

Por tanto, dada la observación ***x*** se trata de determinar la probabilidad a posteriori de que dicha muestra pertenezca a la clase s_j. El proceso para llevar a cabo esta operación se puede realizar por medio de la teoría de la probabilidad de Bayes calculando la probabilidad a posteriori como sigue,

$$P(s_j \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \boldsymbol{m}_j, \mathrm{C}_j) P(s_j)}{\sum_{j=1}^{c} p(\mathbf{x} \mid \boldsymbol{m}_j, \mathrm{C}_j)} \tag{2.12}$$

El numerador del segundo miembro de la ecuación (2.12) incluye dos términos, a saber:

1) La función de densidad de probabilidad $p(\mathbf{x} \mid \boldsymbol{m}_j, \mathrm{C}_j)$ que está dada en (2.11).

2) La probabilidad a priori $P(s_j)$ de que dicha muestra pertenezca a la clase s_j.

Uno de los problemas que se plantean respecto del cálculo de la probabilidad consiste en cómo determinar esta probabilidad antes de observar la muestra.

En cualquier caso, la decisión final sobre la asignación de $\boldsymbol{x}$ a una clase dada se basa en el siguiente criterio (Duda y col., 2000):

$$\boldsymbol{x} \in s_j \quad \text{si} \quad P(s_j \mid \boldsymbol{x}) > P(s_k \mid \boldsymbol{x}) \quad \forall k \neq j \tag{2.13}$$

Teniendo en cuenta la ecuación (2.12) este criterio se puede expresar como sigue ya que el denominador en (2.12) representa la densidad de probabilidad mixta, que no tiene carácter discriminante puesto que es la misma en todos los casos.

$$\boldsymbol{x} \in s_j \quad \text{si} \quad p(\boldsymbol{x} \mid s_j) P(s_j) > p(\boldsymbol{x} \mid s_k) P(s_k) \quad \forall k \neq j \tag{2.14}$$

Como se ha mencionado anteriormente, la única cuestión pendiente estriba en el cómputo de la probabilidad a priori. Cuando ésta no se conoce, lo que se hace es fijarla para todas las clases a un valor constante, por ejemplo a 0.5, para que no intervenga realmente en la decisión.

2.2.3. Vecinos más cercanos

De todos los métodos estadísticos de reconocimiento de patrones, el de Vecinos más cercanos, también llamado *Nearest Neighbours* en terminología inglesa (K-NN) ha mostrado por lo general un elevado rendimiento. La estructura es similar a la utilizada en las redes de Bravais, aunque se desarrollaron por Fix y Hodges (1951).

Para estudiar el comportamiento del algoritmo *K-NN* se ha partido de un conjunto de datos previamente etiquetados en la fase de entrenamiento, de esta forma se ha podido observar cómo dicho clasificador asigna, en la fase de clasificación a una nueva

muestra, una de las clases disponibles en la fase de entrenamiento (Dasarathy y Sánchez, 2000).

Se utiliza clasificación supervisada estimando la distancia de cierto número de muestras (*K* vecinos) a la muestra que se pretende clasificar, determinando su pertenencia a la clase de la que encuentre más vecinos etiquetados, basándonos en un criterio de mínima distancia.

Es una técnica válida solo para datos numéricos, no para clasificadores de textos. En función del tipo de datos que queramos evaluar, a veces conviene utilizar más vecinos o menos, generalmente más vecinos reducen el ruido de la clasificación, sin embargo hace que haya clases demasiado parecidas, y esto es un inconveniente a la hora de decidir la clase a la que pertenece la muestra evaluada.

Ante este problema se aplican reglas que determinan qué hacer en caso de empate entre dos clases respecto a la pertenencia de una muestra. De todos modos la mejor forma de elegir el número de vecinos es mediante el entrenamiento de las muestras.

El principal problema que se encuentra con este algoritmo es cuando hay ruido o los datos contienen características irrelevantes, por ejemplo dos atributos relevantes perderán peso frente a veinte irrelevantes. Una forma de corregirlo es asignando un peso a las distancias de cada atributo, siendo más importantes unos atributos que otros. Se está trabajando para generar algoritmos que evolucionan para mejorar la escalabilidad.

Dado un conjunto de muestras $X = \{x_1, x_2, ..., x_N\} \in \Re^p$ con función de distancia *d*, se permite un preprocesamiento para responder a dos cuestiones:

1) *Vecino más cercano*: localizar la muestra x_l en *X* más cercana a x_k, con $1 \leq k \leq N$ y $k \neq l$.

2) *Rango r*: dado un umbral *r* y un punto x_k, devolver todos los puntos x_l que satisfagan $0 \leq d(x_k, x_l) = d_{kl} \leq r$.

El algoritmo más sencillo para la búsqueda del vecino más cercano es el conocido como fuerza bruta, que calcula todas las distancias de las muestras del entrenamiento respecto a una muestra concreta, y asigna como conjunto de vecinos más cercanos aquél cuya distancia sea mínima.

El índice de similitud entre individuos más utilizado es la distancia euclídea:

$$d(k,l) = \| \sqrt{(x_{k1} - x_{l1})^2 + (x_{k2} - x_{l2})^2 + ... + (x_{kp} - x_{lp})^2} \| \quad (2.15)$$

Y los requisitos que deben cumplir las funciones de distancias son:

1) $d(k,l) \geq 0$

2) $d(k,k) = 0$

3) $d(k,l) = d(l,k)$

4) $d(k,l) \leq d(k,j) + d(j,l)$

Supongamos que el conjunto de datos $X = \{x_1, x_2, ..., x_N\} \in \Re^p$ es un subconjunto del espacio real *p*-dimensional $\Re^p$. Cada $x_k = \{x_{k_1}, x_{k_2}, ..., x_{k_p}\} \in \Re^p$ se denomina vector de características, siendo x_{k_j} la *j*-ésima característica de la muestra x_k.

En la fase de entrenamiento se almacenan las clases y las características que se deben tener en cuenta. Posteriormente se pasa a la fase de clasificación, que evalúa las muestras de las que se desconoce la clase a la que pertenece según el patrón de los datos evaluados inicialmente, calculando las distancias y seleccionando los *k* vecinos más cercanos, que permitirá definir la clase a la que pertenecerá el nuevo elemento.

La búsqueda de los *K* vecinos más cercanos de forma exhaustiva es costosa, pero no mucho más que la búsqueda de un único vecino más cercano, siempre que *K* sea menor que el valor del conjunto de entrenamiento.

Se pueden encontrar los *K* vecinos de la forma siguiente:

1) Cuando se calcula una distancia, ésta se almacena en un vector o lista que contiene los *K* vecinos más cercanos hasta el momento.

2) Se utiliza la distancia de la muestra evaluada al vecino más alejado de los *K* vecinos más cercanos encontrado hasta el momento para hacer una poda.

3) La elección final se realiza asignando a la nueva muestra la clase a la que pertenecen la mayoría de los *k* vecinos.

La figura 2.16 muestra un sencillo ejemplo, tomado de Wikipedia (http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm) que intenta clasificar el

punto verde. Si con el clasificador *k-NN* se utilizan los 3 vecinos más cercanos, la muestra pertenecerá a la clase formada por triángulos porque 2 de los elementos que están más cerca de la muestra evaluada son triángulos y uno cuadrado. Sin embargo, si se utiliza el clasificador con los 5 vecinos más cercanos (5-*NN*), el elemento evaluado pertenecerá a la clase formada por cuadrados ya que hay 3 cuadrados y dos triángulos entre las cinco muestras más próximos al dato.

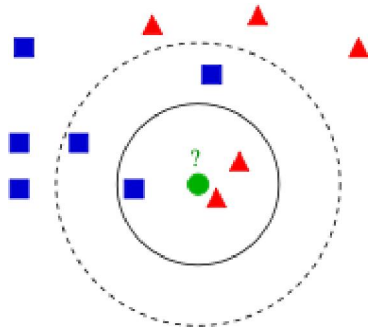


Figura 2.1: Clasificación por análogos

En el entrenamiento cada muestra se añade a la estructura que aprende inicialmente. El valor devuelto por el algoritmo en el caso de elegir k=1, será asociar a la muestra evaluada la clase de un único vecino, el más cercano.

La figura 2.1 refleja un ejemplo de la dificultad que tiene la clasificación de muestras muy cercanas o pertenecientes a clases solapadas, puesto que sin un ajuste de parámetros el acierto producido puede reducirse considerablemente.

2.2.3.1. Vecinos más cercanos Fuzzy (K-NN Fuzzy)

Proviene de la terminología inglesa *K-Nearest Neighbours Fuzzy* (K-NN Fuzzy). Es una variante de Vecinos más cercanos que tampoco necesita conocimiento a priori de las muestras. Se implanta como una estrategia mejorada que utiliza la lógica Fuzzy, de manera que una condición de bondad evalúa las muestras para que éstas puedan pertenecer a más de una clase. El conjunto de muestras desconocidas se asocia a cada clase en función de la distancia a los vecinos conocidos más cercanos a la misma, asignando un grado de pertenencia que indicará cuánto pertenece la muestra a cada uno de los clusters.

Kerwin (2005), compara los *Vecinos más cercanos Fuzzy* utilizando la base de datos Iris con otros algoritmos como Vecinos más cercanos crisp, K-Medias o Bayes, y demuestra cómo los *Vecinos más cercanos Fuzzy* obtienen, en la mayoría de los casos, mejores

resultados que sus competidores, siendo una minoría las situaciones en las que sus rivales reflejan resultados competitivos.

El método *K-NN Fuzzy* tiene la habilidad de producir modelos de clasificación precisos al compararlo con otros paradigmas utilizados en los problemas de clasificación, y el rendimiento obtenido muestra resultados muy buenos (Rosa y Ebecken, 2003).

K-NN Fuzzy se propuso inicialmente para solucionar los problemas de empate que se producían utilizando la teoría de conjuntos Fuzzy. Este método asigna las muestras de las clases a los patrones de entrada, en contraposición a lo que hace K-*NN* que asigna la nueva muestra a una clase directamente.

La ventaja de utilizar la teoría de conjuntos Fuzzy es que se evita la asignación arbitraria, porque los valores de los elementos entrenados en el aprendizaje suministran tanta información que posteriormente, los resultados obtenidos podrán ser utilizados en la clasificación con un elevado nivel de certeza.

La idea básica es asignar una función de distancias desde los *k* vecinos más cercanos. Estos rodearán la muestra y podrán pertenecer a distintas clases, manteniendo el principio de K-NN, la decisión se tomará con la información extraída de los vecinos más próximos.

Una variante para obtener más información es hacer más influyentes no solo los que se encuentren más cerca del objetivo basándose en las distancias, sino los que tengan mayor grado de pertenencia (Kim y Ryu, 2005).

La clase elegida para la muestra entrante se asignará observando los vecinos con mayor grado de pertenencia, esto hará la decisión final más precisa que cuando se comparaba solo con distancias, sobre todo en el caso de clases solapadas.

La base del algoritmo *KNN-Fuzzy* está en asociar la relevancia de un centro a sus *K* vecinos más cercanos y que estos formen parte de la clase. Los resultados serán mejores dependiendo de la precisión de las características extraídas, mostrando la localización del espacio de características para el aprendizaje.

El conjunto de datos $X = \{x_1, x_2, ..., x_N\} \in \Re^p$ debe particionarse en c clústeres $S_i\{i = 1, ..., c\}$, donde $\mu_i(x)$ es el grado de pertenencia de la muestra x_j a la clase S_i, siendo $1 \leq j \leq N$.

El grado de pertenencia de una muestra a cada una de las clases está influenciado por la inversa de las distancias de los vecinos. Tomando en consideración la proximidad de la muestra a ser clasificada respecto a una determinada clase, la inversa de las distancias sirve como peso.

$$\mu_i(x) = \frac{1/\| x - S_i \|^{2/(m-1)}}{\sum_{i=1}^{c} (1/\| x - S_i \|^{2/(m-1)})} \qquad \text{donde } i = 1, \ldots, c \tag{2.19}$$

m determina el peso exponencial, esto es la importancia que le damos al centro por estar más cerca de la muestra evaluada. Si el valor de *m* es igual a 2, el peso será la distancia al cuadrado para cada punto. Para establecer la distancia a las clases existen diferentes propuestas, una de ellas es considerar el centro de la clase y calcular la distancia de una muestra dada a dicho centro. Este es el criterio adoptado en este trabajo, como se explica posteriormente en el capítulo tres.

Si este valor disminuye hacemos que el peso de los vecinos más cercanos sea menos relevante. Cuando *m* es mayor que uno significa más contribución del peso.

Los pasos del método general de búsqueda en el espacio de características son:

1) Seleccionar un candidato a vecino más cercano por aproximación de entre los individuos del conjunto de entrenamiento.
2) Calcular su distancia *d* al individuo en cuestión.
3) Si la distancia es menor que la distancia del vecino más cercano hasta el momento, d_{kl}, se actualiza el vecino más cercano y se eliminan del conjunto de entrenamiento los individuos que no pueden estar más cerca que el estudiado.
4) Repetir los pasos anteriores hasta que no queden muestras por seleccionar en el conjunto de entrenamiento, ya sea por haber sido seleccionadas o eliminadas.

El vecino más cercano x_k respecto al centro S_i, ejerce mayor influencia cuando refleja el grado de pertenencia, que está representado en la ecuación 2.18. El vecino x_k más alejado mostrará menos influencia.

Al ir más allá de la información dada por la distancia, poniendo énfasis en el grado y haciendo de éste un factor a tener en cuenta, se puede elegir un vecino que contiene

características más parecidas a la muestra que se evalúa cuando las distancias no están bien consideradas, y especialmente en situaciones como en clases solapadas.

Este sistema tiene principalmente dos ventajas respecto al método de Vecinos más cercanos clásico:

- Determinar el grado de pertenencia supone un factor de certeza a la hora de asignar una muestra a una clase.
- Utilizando la teoría Fuzzy, el problema de distancias iguales a distintos centros se soluciona mediante la pertenencia, además de que los resultados reflejan más éxito en la distribución de clases.

2.3. Métodos clásicos de combinación de clasificadores

El tema relativo a la combinación de clasificadores, también denominada hibridación, ha sido ampliamente estudiado en la literatura, por ejemplo, Partridge y Griffith (2002) establecen un marco de trabajo fijando los criterios fundamentales para diseñar un método híbrido, que utiliza diferentes clasificadores.

Existen algunas alternativas sobre la combinación de clasificadores ampliamente difundidas en la literatura especializada sobre el tema. Siguiendo las referencias de Kuncheva (2004), Kittler y col. (1998) o Duda y col. (2000) el problema se plantea en los términos que se describen a continuación.

La muestra x_k que deseamos asignar se suministra a los distintos clasificadores para su procesamiento. Supongamos que disponemos de m clasificadores y varias clases S_i donde i identifica cada clase. En la literatura se han descrito cuatro métodos de combinación con una aceptación generalizada.

En el método denominado *votador mayoritario* cada clasificador determina la clase a la que pertenece una muestra dada, y dependiendo del número de votos obtenido a favor de cada clase se asigna a la que obtenga mayoría. En caso de empate se aplican reglas en segundo plano como el tanto por ciento, para decantar el voto hacia una u otra clase.

En Kittler y col. (1998) se establece un marco de referencia para la combinación de clasificadores desde el punto de vista probabilista, es decir que las salidas proporcionadas por cada clasificador son de naturaleza probabilista. Se trata de un

planteamiento muy próximo al que se hace en este trabajo de investigación. Por tanto los métodos de combinación allí expresados constituyen buenas referencias para ser utilizadas como medidas de comparación.

En concreto son tres los métodos expresados en forma de reglas, a saber:

- *Reglas de máximo*
- *Reglas de mínimo*
- *Regla de media*

La tabla 2.1 muestra las decisiones tomadas por cada clasificador para cada una de las *i* clases. En la tabla 2.2 se ofrecerán las reglas utilizadas por dichos clasificadores combinando las decisiones obtenidas de la tabla 2.1.

	Clase 1	**Clase 2**		**Clase *i***
Clasificador 1	${}^{1}S_1$	${}^{1}S_2$	…	${}^{1}S_i$
Clasificador 2	${}^{2}S_1$	${}^{2}S_2$	…	${}^{2}S_i$
Clasificador *m*	${}^{m}S_1$	${}^{m}S_2$	…	${}^{m}S_i$

Tabla 2.1 Decisiones dadas por cada clasificador

Método	**Clase 1**	**Clase 2**		**Clase i**	**Regla aplicada**
Máximo	${}^{h}S_1$	${}^{h}S_2$	…	${}^{h}S_i$	${}^{H}S_J = \max_{j=1}^{i} \max_{h=1}^{m} \{{}^{h}S_j\}$
Mínimo	${}^{h}S_1$	${}^{h}S_2$	…	${}^{h}S_i$	${}^{H}S_J = \min_{j=1}^{i} \max_{h=1}^{m} \{{}^{h}S_j\}$
Media	${}^{M}S_1 = \frac{1}{m}\sum_{l=1}^{m} {}^{i}S_1$	${}^{M}S_2 = \frac{1}{m}\sum_{l=1}^{m} {}^{i}S_2$	…	${}^{M}S_i = \frac{1}{m}\sum_{l=1}^{m} {}^{i}S_i$	${}^{H}S_J = \max_{j=1}^{i} \{{}^{M}S_j\}$

Tabla 2.2 Métodos sobre combinación de clasificadores

2.4. Técnicas de partición de conjuntos de muestras para el entrenamiento

Existen diferentes posibilidades que pueden aplicarse para mejorar el rendimiento de los clasificadores en la minería de datos. Son propuestas que mediante la reordenación de los datos y antes de acceder a la entrada, hacen que la clasificación sea más eficiente a la hora de evaluar las muestras por los distintos clasificadores, por haber tenido un mejor entrenamiento.

Una aplicación de esta índole muy conocida es la conocida como *validación cruzada* o *Cross-validation* en terminología inglesa a la que nos referiremos con esta terminología a lo largo de la memoria. Dicha técnica, tiene múltiples variantes como el *10-Fold Cross-Validation* o el *Leave-One-Out*. La *Estratificación* es otro método más innovador pero no por ello menos importante.

Para mejorar los resultados de los clasificadores se han utilizado todas ellas en este estudio comparativo. A continuación proporcionamos una breve explicación de cada una de ellas.

2.4.1. Estratificación

Uno de los principales problemas que a menudo surgen en los clasificadores de datos es que el tamaño de los conjuntos de muestra que se analizan suele ser muy grande. Cuando esto sucede, generalmente 2/3 de esos datos se utilizan para el entrenamiento de las muestras y el 1/3 restante se usa para su clasificación, de tal modo que si por ejemplo hubiese 900 muestras para evaluar ordenadas por la clase a la que pertenecen, y suponiendo que hubiera 10 clases, se entrenarán las 600 primeras y al estar ordenadas por clase es posible que alguna clase no se entrenase en este proceso. Este hecho dará lugar a que en la fase de clasificación se encuentren algunas de las clases que no se han entrenado previamente y viceversa, aumentando significativamente el porcentaje de error en la decisión final.

Para evitar este problema, es conveniente que haya representación de todas las clases tanto en el entrenamiento como en la clasificación. El proceso de heterogeneizar el conjunto de muestras entre las dos fases se le llama *estratificación*.

La *estratificación* hace una división inicial del conjunto de datos en subconjuntos disjuntos con una distribución heterogénea de clases. Cada uno de los subconjuntos es independiente de los demás y la cantidad de muestras que tenga determinará su tamaño. Utilizando el número adecuado de subconjuntos se pueden reducir de forma significativa el tiempo de ejecución. Esta reducción permitirá la ejecución de más recursos por parte del algoritmo que vaya a utilizarse (Cano y col. 2004), así se mejorará su rendimiento y con ello aumentará el porcentaje de acierto.

Siguiendo la estrategia inicial de la *estratificación,* el conjunto de datos inicial Z es dividido en N subconjuntos disjuntos de igual tamaño $Z = \{X_1, X_2, ..., X_N\}$. La

distribución de las clases en el proceso de partición se mantiene, y el conjunto de datos que se entrena *Y* será complementario a *X*, $Y = Z \setminus X$.

Dicho en otras palabras, cuando un conjunto de datos está dividido en grupos homogéneos, mediante la *estratificación* se produce un intercambio de elementos entre estos grupos con el fin de hacer una distribución heterogénea. El uso de la *estratificación* es vital cuando hay un número elevado de clases para un grupo de muestras pequeño. El papel de la *estratificación* es un agrupamiento inteligente de los elementos que componen la población de determinado subconjunto.

Zseby (2003) llega a la conclusión de que utilizar subconjuntos de igual tamaño da resultados similares a utilizarlos de diferentes tamaños. Sin embargo, sería posible mejorar la precisión de este proceso si buscamos los límites entre los subconjuntos de forma automática utilizando un análisis del cluster, eligiendo límites óptimos entre éstos en lugar de hacerlos de tamaño fijo o fijarlos manualmente (Fernandes y col., 2008). Además permitiría reducir la cantidad de información a almacenar en memoria reduciendo el tráfico durante el proceso.

En el ejemplo de la figura 2.2 puede apreciarse cómo se pasa de un conjunto con 900 muestras ordenadas por clase, habiendo 10 clases diferenciadas por colores, y el mismo conjunto de muestras después de que se le haya aplicado la técnica de la *estratificación*:

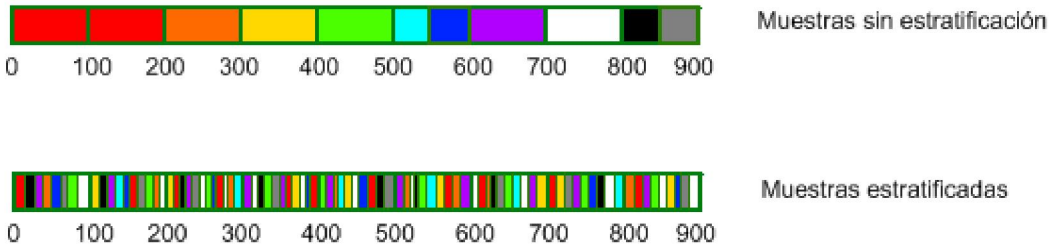


Figura 2.2: Heterogeneización de un conjunto de muestras

En la primera imagen se aprecia el conjunto de 900 muestras bien diferenciadas por clases, siendo las 200 primeras pertenecientes a la clase 1, de color rojo, las siguientes 100 muestras pertenecientes a la segunda clase de color naranja, y así sucesivamente con todas las muestras.

Si no se utiliza la *estratificación*, al entrar en la fase de entrenamiento se tomarán las 600 primeras muestras (2/3 del total) y por consiguiente no habrá entrenado cuatro de las diez clases existentes. Sucederá igual al hacer la clasificación, puesto que las 300 muestras que intentará clasificar será el 1/3 restante, y el número de aciertos será nulo.

La muestra estratificada hace una división heterogénea de los datos, para que tanto al entrenar como al clasificar, todas las clases del conjunto de muestras se puedan evaluar a la vez.

Los algoritmos no evolutivos que aplican esta técnica pueden ejecutarse de una forma más eficiente ya que reducen los recursos necesarios. Es una buena solución al problema de la escalabilidad ya que permite a los algoritmos aplicados mantener su eficiencia incluso ejecutando conjuntos de datos de gran tamaño.

2.4.2. Cross-Validation

La estrategia de *Cross-Validation* consiste en la división de un conjunto de muestras que se pueden analizar en dos conjuntos disjuntos de datos. Uno de estos conjuntos entrenará las muestras que contiene, y los resultados obtenidos se aplicarán al otro conjunto que será utilizado para la clasificación de muestras. Hay que tener en cuenta que la división de las muestras en dos conjuntos fijos es una simplificación de la implementación de una división en *k* subconjuntos. El resultado se obtiene tras una optimización en cada iteración, acotando la probabilidad de error estimado como promedio de los errores en cada iteración (Hurtado, 2007).

El *Cross-Validation* tiene una clara y grave desventaja, puesto que la división aleatoria de un pequeño conjunto de datos para el análisis implica la casi segura pérdida de información que no podrá ser recuperada. La pregunta clave sería determinar el número de conjuntos en los que se debe dividir para obtener el rendimiento óptimo. Aunque en cada iteración se hace un promedio del error producido, existe el problema de que no hay representatividad de las muestras. Este grave problema lo solucionamos utilizando la técnica de la estratificación.

La forma más común de aplicar la técnica de *Cross-Validation* es dejar el 10% de las muestras para realizar la evaluación y entrenar el 90% restante según la figura 2.3:

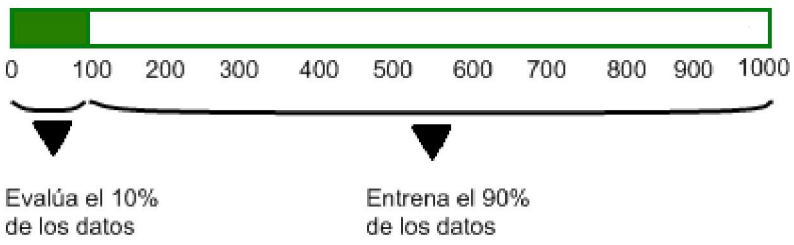


Figura 2.3. Técnica de *Cross-Validation* sobre 1000 muestras

Una síntesis del procedimiento realizado en *Cross-Validation* es el siguiente:

- Entrena el 90% de las muestras para extraer un modelo.
- Evalúa con el modelo extraído con el 10% de datos restantes, obteniendo la tasa de error (porcentaje de acierto) del algoritmo.
- Repite el proceso 9 veces de tal forma que se evalúe cada vez el 10% de los datos, entrenando el 90% restante
- Calcula la media del resultado final obtenido.

2.4.2.1. Ten-fold Cross-Validation

Cuando el conjunto de muestras se separa en bloques X y el número de estos es 10, se determina el *10-fold Cross-Validation*. Cross-Validation no es una técnica desconocida porque diversos estudios la han considerado una medida fiable que ofrece buena complejidad en tiempo real (Bouckaert, 2008). El modelo repite diez veces el proceso de entrenamiento y el porcentaje de acierto resultante es utilizado como una medida de bondad del algoritmo evaluado (Lakshmanan, 2007).

Si el conjunto de datos es Z, cada partición aleatoria de muestras X tiene el mismo número de datos que las demás, y los subconjuntos resultantes son $X_1, X_2, ..., X_N$.

El algoritmo se especifica en la ecuación (2.20):

For $k=1$ to N

X_k se evalúa y entrenan Z-X_k

n_k = resultado para X_k (2.20)

Devuelve $(n_1+n_2+\ldots+n_N)/|N|$

Se explica de la siguiente manera; evaluando todos los subconjuntos X_k desde el primero hasta el último, se clasifica el subconjunto X_k, entrenando el resto para incrementar el aprendizaje (Z-X_k). Este proceso se realizará tantas veces como sea necesario para analizar todas las muestras, extrayéndose la media al calcular los resultados de la clasificación de cada uno de los subconjuntos.

Este es el algoritmo más genérico posible que se lleva a cabo en *N-fold Cross-Validation*, para concretar en *10-fold Cross-Validation* hay que darle valor a $N=10$.

En la figura 2.4 podemos ver cómo actúa Cross-Validation aplicado al caso concreto de 10 folds. Se trata de un ejemplo con 1000 muestras, de forma que para aplicar el

algoritmo antes descrito realiza la división en 10 subconjuntos (*N=10*) y ejecuta los siguientes conjuntos de instrucciones;

- Para cada iteración:
    - Entrenan todos los bloques salvo el bloque en color verde, que será el utilizado para la clasificación.
    - Devuelve el resultado obtenido.
- Al finalizar el proceso:
    - Realiza una media de los resultados extraídos en cada una de las iteraciones para calcular el porcentaje de acierto del algoritmo.

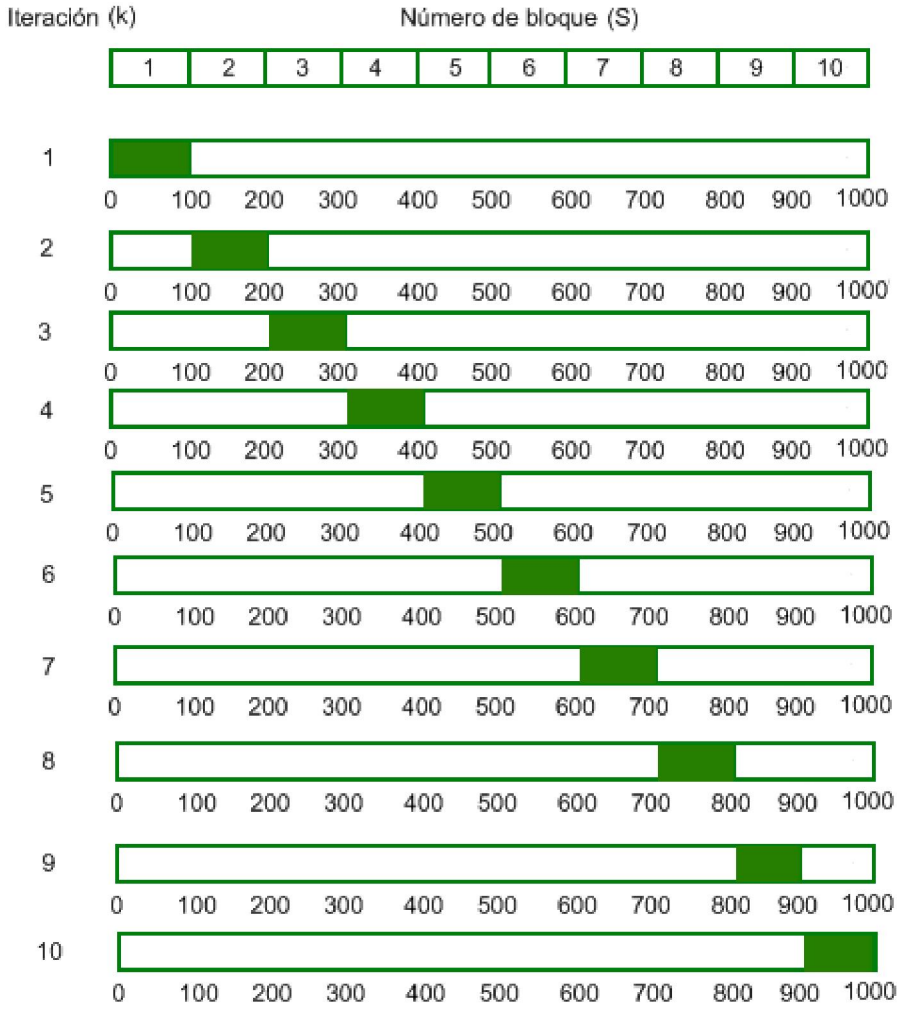


Figura 2.4: Diez particiones que utilizan *10-fold Cross-Validation*

Cross-Validation es una técnica que se ha utilizado con métodos basados en redes neuronales (Zhang y col, 2006; Liu y col., 2008), Nearest Neighbors, Fuzzy, etc. ya que está demostrado que mejora la complejidad computacional de los mismos al separar el

conjunto de muestras inicialmente dadas en varios subconjuntos (Merwe y Hoffman, 2001).

2.4.2.2. Leave-One-Out

Este es el caso límite de Cross-Validation donde el conjunto de muestras X no se subdivide en N subconjuntos sino que se realiza, $X = |Z|$, lo que proporciona una mejor estimación al utilizar todos los datos menos uno en el entrenamiento (Merwe y Hoffman, 2001).

Las muestras no se dividen en subconjuntos, simplemente omiten en cada iteración un dato que será aplicado para la clasificación, y entrena con los datos restantes. El resultado con el dato que se clasifica se almacena, se pasa el siguiente dato a la fase de clasificación y se entrena con todos los demás incluyendo el dato que se clasificó la primera vez. Una vez clasificado el último dato de la muestra, calcula con todos los datos el porcentaje de acierto (Liu y col., 2008; Baumann, 2003).

Las ventajas de este algoritmo son que utiliza un mayor número de datos para entrenar, y que tiene un resultado determinista, pues el experimento siempre ofrecerá los mismos resultados. Sin embargo, poniendo un ejemplo supongamos dos clases en un conjunto de muestras, con la mitad de muestras pertenecientes a cada clase. Cuando una muestra es extraída del conjunto de datos para la clasificación, su clase queda en minoría en el entrenamiento, la mayoría induce a predecir siempre la clase equivocada puesto que el *Leave-One-Out* estima un porcentaje de acierto para la mayoría de muestras, por lo que el resultado será un 0% de acierto (Kohavi, 1995).

Un inconveniente que tiene es que necesita ser ejecutado sobre conjuntos de muestras relativamente pequeños, ya que si la muestra a evaluar es muy grande, el coste computacional de recursos puede ser excesivo gastando demasiado tiempo en la ejecución del proceso.

Otra característica es que el uso del *Leave-One-Out* no puede ser combinado con la estratificación porque como hemos dicho, esta forma de evaluación extrae las muestras de una en una, lo que significa que aunque el orden sea distinto no habrá variación de los resultados.

Suponiendo 10 muestras (no conjuntos de muestras), el proceso *Leave-One-Out* que aparece en la figura 2.5, representa en cada iteración una muestra del conjunto total para

la clasificación, entrenando con todas las demás. Al finalizar el entrenamiento hace una media de los resultados extraídos en cada iteración obteniendo el porcentaje de acierto del algoritmo para ese conjunto de muestras.

Figura 2.5. *Leave-One-Out* sobre diez muestras

Capítulo 3

3. Clasificación, problemática y diseño

3.1. Introducción

Antes de comenzar el proceso de diseño de un clasificador conviene destacar las fases o mecanismos involucrados en todo proceso de aprendizaje-clasificación. La figura 3.1 muestra un esquema general donde este proceso queda detallado.

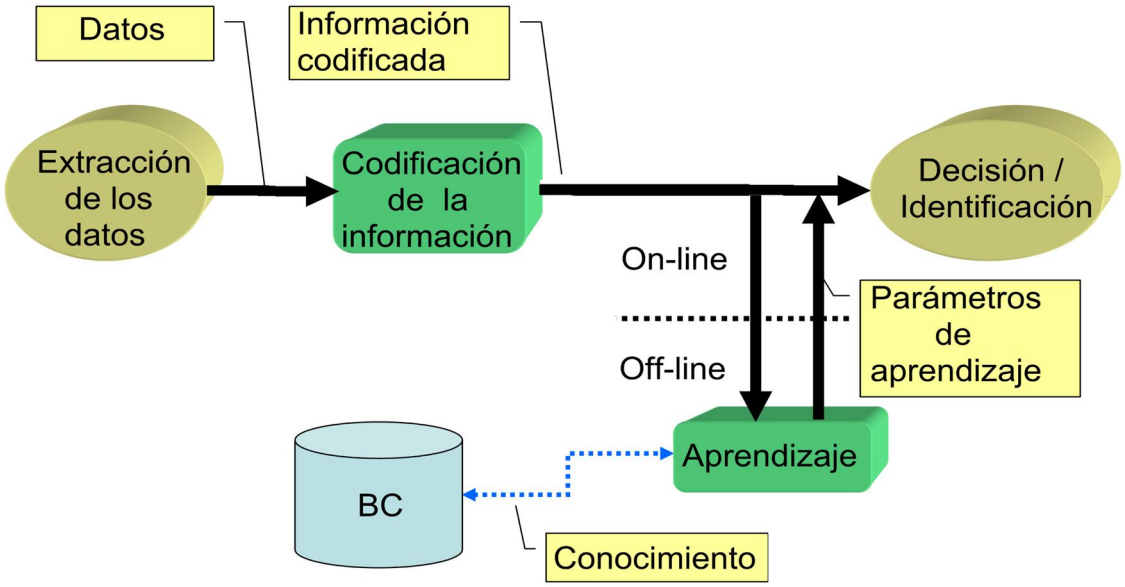


Figura 3.1 Características generales del clasificador

Existen dos procesos perfectamente identificados: *on-line* y *off-line*. Durante el proceso *off-line* se lleva a cabo el aprendizaje propiamente dicho, mientras que en el *on-line* se realiza la clasificación o reconocimiento.

Los pasos a seguir durante el proceso son los siguientes:

Extracción de datos: en nuestro caso, se extraen los datos a partir de un fichero de texto que contiene valores numéricos separados por comas, que serán los atributos, donde el último valor de cada muestra corresponde a la clase del dato. Se realiza la *estratificación* para separar las clases y heterogeneizar el conjunto de muestras y se aplica *ten-fold cross-validation* o *leave-one-out*.

Codificación de la información: una vez que la información se dispone de forma que pueda ser procesada adecuadamente, será codificada en forma de vector o matriz y normalizada en unos rangos de variabilidad apropiados. Estos datos se utilizarán tanto en el aprendizaje como en la clasificación.

Aprendizaje: con esta información se aplican los métodos de aprendizaje que se deseen, tras los cuales se *aprende*, es decir, se obtienen los parámetros correspondientes según el método aplicado. Estos parámetros se almacenan en la Base de Conocimiento (BC) con el fin de poderlos recuperar durante la fase de clasificación *on-line*.

Para cada algoritmo se aprenden los parámetros necesarios para su funcionamiento:

- **Fuzzy Clústering**: se obtienen los centros de los clústeres y grados de pertenencia a los mismos.
- **Clasificador Bayesiano**: se obtienen los centros de los clústeres y las matrices de covarianza de los mismos.
- **Clasificador de Nearest-Neighbours Fuzzy**: se obtienen los centros de los clusters y los grados de pertenencia de los mismos.

Decisión: Una vez que los parámetros de aprendizaje están almacenados en la BC, llegan nuevas muestras a esta fase y se asigna cada una a la clase que pertenezca, según la decisión tomada en función de los resultados extraídos por cada clasificador.

Las muestras clasificadas durante este proceso se incorporan nuevamente a la BC, de forma que los parámetros de aprendizaje puedan actualizarse con el máximo número de muestras de entrenamiento, en futuros procesos de aprendizaje.

3.2. Antecedentes

La motivación para el diseño del clasificador que se propone en el presente trabajo tiene sus antecedentes en un primer intento por aplicar el clasificador de *Bayes* en toda su extensión, es decir, teniendo en cuenta que la decisión de clasificación mediante *Bayes* tiene en cuenta tanto la función de densidad de probabilidad estimada según la ecuación 2.11 como la probabilidad a priori, ecuaciones 2.12 a 2.14.

La clasificación óptima actualmente es una tarea pendiente, y elegir un clasificador adecuado para realizar todo el estudio es muy complicado. Sin embargo hay estudios

que afirman que el mejor clasificador es el *Nearest Neighbours Fuzzy* cuando utiliza la técnica *Leave-One-Out*, (Alippi y col., 2008).

Los clasificadores *Bayes* y *Fuzzy* fueron elegidos porque a través de la bibliografía se ha demostrado el éxito que han logrado a lo largo de tantos años de uso.

La técnica de *Cross-Validation* llamada *Leave-One-Out* no requiere información a priori sobre la distribución estadística de las clases y, aunque es costoso a nivel computacional, la facilidad de su combinación con otras aplicaciones que mejoran el rendimiento fue una de las razones que motivó su utilización.

La *Estratificación* es imprescindible en el proceso de clasificación a la hora de generar conjuntos de datos heterogéneos, por lo que su utilización es vital para el correcto funcionamiento del *Ten-Fold Cross-Validation*. Así mismo el *Ten-Fold Cross-Validation* es una técnica que ha demostrado un comportamiento novedoso en el proceso de clasificación, mejorando sustancialmente los porcentajes de acierto.

3.3. Diseño del método

Para diseñar el método se ha divido el proceso en varias partes bien diferenciadas. Llevar un orden es importante, ya que cada fase utilizará información de la fase anterior.

El planteamiento general que se hace en el proceso de clasificación propuesto es el siguiente:

Antes del entrenamiento

1. Se aplican la *estratificación* que se encarga de colocar los datos de forma desordenada en el caso de que las clases aparezcan inicialmente agrupadas, heterogeneizando el conjunto de muestras.
2. Según el proceso que vaya a utilizarse, se aplica el *cross-validation* con la técnica *ten-fold cross-validation* o con *leave-one-out*, para subdividir el conjunto inicial de muestras antes del proceso de aprendizaje

Fase de entrenamiento

3. Las muestras iniciales que constituyen las entradas al sistema, se utilizan para realizar una estima de la función de densidad de probabilidad de Bayes, y para obtener los centros de los clústeres mediante el clasificador Fuzzy.

4. Tanto los parámetros estimados en Bayes ($\boldsymbol{m}_j$, C_j) como los centros Fuzzy ($\boldsymbol{v}_j$) se almacenan en la base de conocimiento *BC*.

Fase de clasificación

5. Ante la llegada de cada nueva muestra $\boldsymbol{x}_s$, se trata de decidir a qué clase pertenece. Para ello se recupera de la *BC* el conocimiento almacenado previamente durante la fase de entrenamiento: $\boldsymbol{m}_j$, C_j, $\boldsymbol{v}_j$.
6. Para cada $\boldsymbol{v}_j$ se obtiene el grado de pertenencia μ_{sj} de dicha muestra a cada una de las clases según el clasificador fuzzy. Estos valores constituyen un conocimiento previo sobre la pertenencia de la muestra a las clases, dicho grado de pertenencia puede actuar como la probabilidad a priori para el clasificador Bayesiano. Por tanto, la probabilidad a priori de que la muestra $\boldsymbol{x}_s$ pertenezca a la clase w_j se obtiene asignándole dicho grado de pertenencia $P(w_j) = \mu_{sj}$.
7. Conocida la probabilidad a priori y calculando la verosimilitud a través de la función de densidad de probabilidad, la probabilidad a posteriori se obtiene fácilmente a través de la ecuación 2.12.
8. La decisión final se realizará mediante la combinación de las estrategias de clasificación, que determinarán para cada caso el algoritmo óptimo entre los estudiados.

En la figura 3.2 se muestra el esquema general del procedimiento propuesto como alternativa en la clasificación de minería de datos. En él se distinguen las dos fases ya mencionadas y perfectamente definidas: entrenamiento y clasificación.

En la Base de Conocimiento *BC* se almacenan los siguientes datos:

- Los centros $\boldsymbol{v}_j$ procedentes del clasificador *Fuzzy*.
- Los centros $\boldsymbol{m}_j$ y las matrices de covarianza $\boldsymbol{C_j}$ procedentes de *Bayes*.
- Los centros $\boldsymbol{k}_j$ procedentes del clasificador *K-NN Fuzzy*.

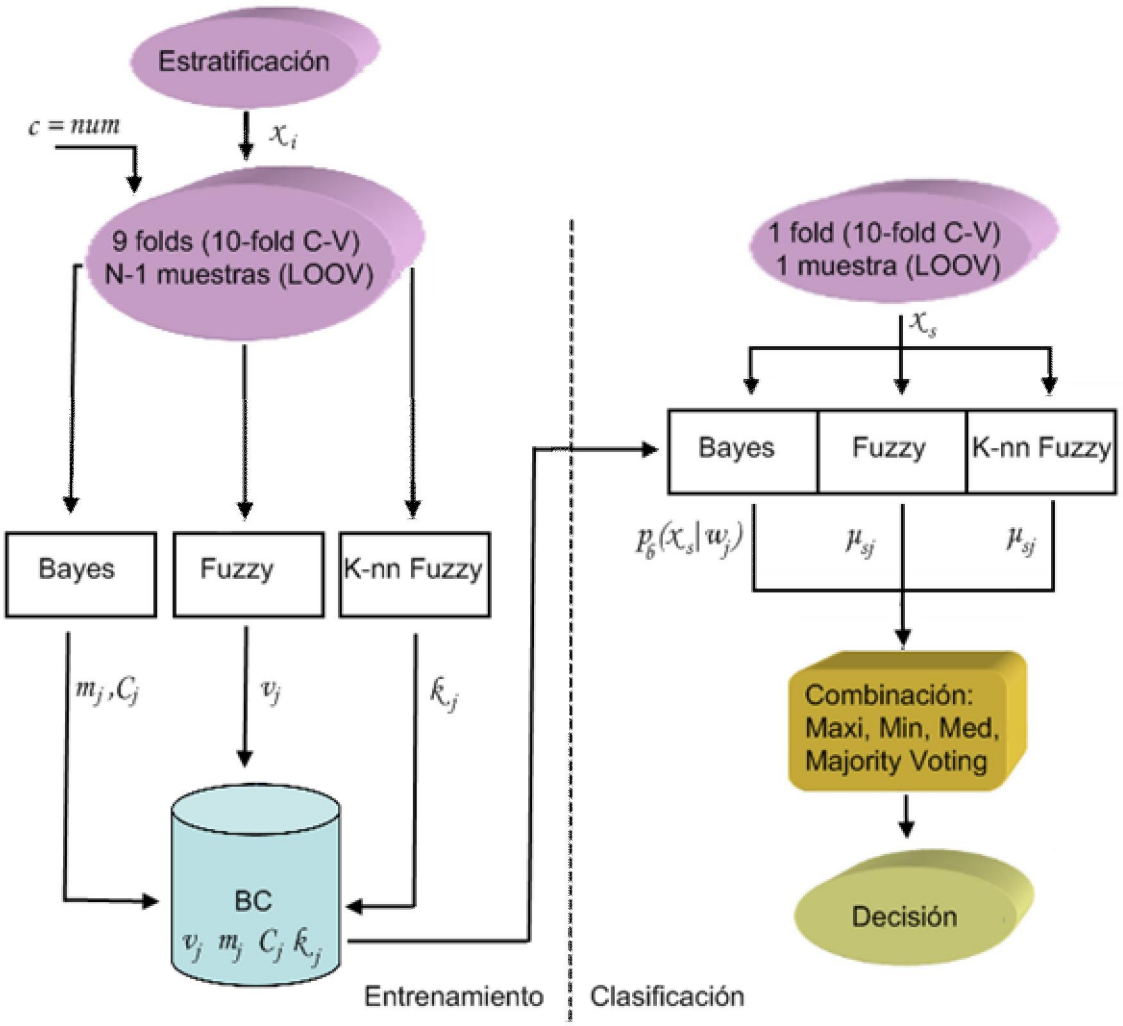


Figura 3.2 Diseño del clasificador

En los demás apartados de cada una de las fases obtenemos una serie de datos que vienen resumidos en la tabla 3.1, que sirven para calcular los resultados mediante los distintos algoritmos.

Clasificador	**Fase de Aprendizaje**			**Fase de Decisión**	
	Naturaleza	**Parámetros aprendidos**	**Parámetros requeridos**	**Regla de decisión**	**Salidas fase de decisión**
Agrupamiento borroso (Fuzzy)	Supervisado	centros de las clases $\boldsymbol{v}_j$	m (peso exponencial)	máximo grado de pertenencia	Grados de pertenencia de $\boldsymbol{x}_s$ a las clases w_j: μ_{sj}
Estima de máxima verosimilitud (Bayes)	Supervisado	centros de las clases y matrices de covarianza $\boldsymbol{m}_j$, C_j	no aplicable	máxima probabilidad	Probabilidades de pertenencia de $\boldsymbol{x}_s$ a las clases w_j: $p_b\left(\boldsymbol{w}_j \mid \mathbf{x}_s\right)$
Vecinos Más Cercanos Fuzzy	Supervisado	Centros de las clases $\boldsymbol{k}_j$	m (peso exponencial)	máximo grado de pertenencia	Grados de pertenencia de $\boldsymbol{x}_s$ a las clases w_j: μ_{sj}

Tabla 3.1. Información extraída en el proceso

Para completar el proceso de aprendizaje y comprobar la eficiencia del método utilizado, la fase de clasificación debe determinar a qué clase pertenece una muestra dada $\boldsymbol{x}_s$. El procedimiento consiste en que cada una de los métodos empleados recupere exactamente lo que necesita de la *BC*. Se explican a continuación:

a) Clasificación mediante Bayes

Siguiendo el esquema de la figura 3.2, el clasificador de *Bayes* extrae de la *BC* tanto los centros de cada una de las clases como las matrices de covarianza, es decir: $\boldsymbol{m}_j$ y C_j.

Con esta información recuperada y mediante la ecuación 2.11, que reproducimos aquí por simplicidad, ecuación 3.1, se obtiene la probabilidad de pertenencia a cada una de las clases. Concretamente, mediante la ecuación 3.1 indicamos que la probabilidad de *Bayes* (p_b) de que la muestra $\boldsymbol{x}_s$ pertenezca a la clase w_j representada por sus parámetros $w_j \equiv (\boldsymbol{m}_j, C_j)$ es la que se proporciona a continuación,

$$p_b(\boldsymbol{x}_s \mid \mathbf{m}_j, \mathrm{C}_j) = \frac{\mathbf{1}}{(2\pi)^{d/2}\left|C_j\right|^{1/2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{x}_s - \mathbf{m}_\mathbf{j}\right)^t C_j^{-1}\left(\boldsymbol{x}_s - \mathbf{m}_j\right)\right\} \qquad (3.1)$$

De esta forma calcularemos tantas p_b como clases tengamos, es decir j = 1, 2, …,c. Dado que los valores p_b pueden exceder el límite de +1 y por tanto sobrepasar el valor límite superior de los valores de probabilidad, es práctica común derivada de la propia teoría de *Bayes* proceder a realizar una normalización, que en realidad se trata de la aplicación del cálculo de la función de densidad de probabilidad mixta. Por consiguiente, aplicando dicha normalización obtenemos finalmente,

$$p_b(\boldsymbol{x}_s \mid \mathbf{m}_j, \mathrm{C}_j) = \frac{p_b(\boldsymbol{x}_s \mid \mathbf{m}_j, \mathrm{C}_j)}{\sum_{i=1}^{c} p_b(\boldsymbol{x}_s \mid \mathbf{m}_i, \mathrm{C}_i)} \qquad (3.2)$$

De esta forma se consigue restringir los valores de probabilidad al rango [0,+1], que será un requisito imprescindible para llevar a cabo posteriormente la operación de agregación mediante los operadores del mismo nombre de naturaleza borrosa.

En resumen, si se dispone de las clases $w_1, w_2, \ldots, w_c$ como posibles para clasificar la muestra dada, se dispondrá igualmente del mismo número de probabilidades de pertenencia a cada una de las clases según la ecuación (3.2), obteniendo finalmente la serie de valores dados en la ecuación (3.3).

$$\begin{gathered} p_b(\boldsymbol{x}_s \mid w_1) \equiv p_b(\boldsymbol{x}_s \mid \boldsymbol{m}_1, \mathrm{C}_1);\ p_b(\boldsymbol{x}_s \mid w_2) \equiv p_b(\boldsymbol{x}_s \mid \boldsymbol{m}_2, \mathrm{C}_2), \\ .., p_b(\boldsymbol{x}_s \mid w_c) \equiv p_b(\boldsymbol{x}_s \mid \boldsymbol{m}_c, \mathrm{C}_c) \end{gathered} \qquad (3.3)$$

b) Clasificación mediante el método Fuzzy

El clasificador *Fuzzy* recibe los centros de los clústeres $\boldsymbol{v}_j$ almacenados en la *BC*. Dada la muestra de entrada $\boldsymbol{x}_s$ y utilizando la ecuación 2.4 determinamos el grado de pertenencia μ_{sj}, de dicha muestra a cada una de las clases j. Esta operación se lleva a cabo mediante la ecuación citada que reproducimos a continuación por simplicidad,

$$\mu_{sj} = \frac{\left(\frac{1}{\left\|\boldsymbol{x}_s - \boldsymbol{v}_j\right\|_G^2}\right)^{2/m-1}}{\sum_{h=1}^{c}\left(\frac{1}{\left\|\boldsymbol{x}_s - \boldsymbol{v}_h\right\|_G^2}\right)^{2/m-1}} \qquad j = 1,\ldots,c;\ k = 1,\ldots,n \qquad (3.4)$$

Realizado este proceso, se determinan los grados de pertenencia de la citada muestra a cada una de las clases. Resultando de la propia definición de los grados de pertenencia,

que los mismos están restringidos al rango de valores [0,+1] como en el caso de la probabilidad de Bayes.

De igual modo que con el clasificador de Bayes, si se dispone de las clases $w_1, w_2, ..., w_c$ como posibles para clasificar la muestra dada, se dispondrá igualmente del mismo número de grados de pertenencia a cada una de las clases según la ecuación 3.4, obteniendo finalmente la secuencia de valores dados en 3.5.

$$\mu_{s1} \equiv \mu(\boldsymbol{x}_s, w_1),\ \mu_{s2} \equiv \mu(\boldsymbol{x}_s, w_2)\ , \ldots,\ \mu_{sc} \equiv \mu(\boldsymbol{x}_s, w_c) \tag{3.5}$$

Cuando se utiliza el clasificador *Fuzzy* de forma individual, es decir sin ser combinado con ningún otro, el criterio para determinar a qué clase pertenece $\boldsymbol{x}_s$ consiste en seleccionar el máximo valor de entre todos los grados de pertenencia obtenidos y asignar $\boldsymbol{x}_s$ a la clase que corresponda. Formalmente esto se expresa como sigue,

$$\boldsymbol{x}_s \in w_j \mid \quad \mu_{sj} > \mu_{sk} \quad \forall k \neq j \quad j = 1,2,...c \tag{3.6}$$

c) Clasificación mediante Vecinos más cercanos Fuzzy

Vecinos más cercanos Fuzzy recibe los centros de los clústeres $\boldsymbol{k}_j$ almacenados en la *BC*. Dada la muestra de entrada $\boldsymbol{x}_s$ y utilizando la ecuación 2.19 determinamos el grado de pertenencia μ_j, de dicha muestra a cada una de las clases *j*. Esta operación se lleva a cabo mediante la ecuación citada que reproducimos a continuación por simplicidad,

$$\mu_i(x) = \frac{1/\| x - \boldsymbol{k}_i \|^{2/(m-1)}}{\sum_{j=1}^{c} (1/\| x - \boldsymbol{k}_j \|^{2/(m-1)})} \tag{3.7}$$

Realizado este proceso, se determinan los grados de pertenencia de la muestra a cada una de las clases. Resultando de la propia definición de los grados de pertenencia, que los mismos están restringidos al rango de valores [0,+1], y esto permitirá hacer comparaciones tanto con Bayes como con Fuzzy.

De igual modo que con los clasificadores Bayes y Fuzzy, si se dispone de las clases $w_1, w_2, ..., w_c$ como posibles para clasificar la muestra dada, se dispondrá igualmente del mismo número de grados de pertenencia a cada una de las clases según la ecuación 3.7, obteniendo finalmente la secuencia de valores dados en 3.8.

$$\mu_{s1} \equiv \mu(\boldsymbol{x}_s, w_1),\ \mu_{s2} \equiv \mu(\boldsymbol{x}_s, w_2)\ , \ldots,\ \mu_{sc} \equiv \mu(\boldsymbol{x}_s, w_c) \tag{3.8}$$

Cuando se utiliza el clasificador *Vecinos más cercanos Fuzzy* de forma individual $\boldsymbol{x}_s$ se selecciona para una clase en función del máximo valor de los grados de pertenencia de entre todos los obtenidos. Formalmente esto se expresa en la ecuación (3.9).

$$\boldsymbol{x}_s \in w_j \mid \quad \mu_{sj} > \mu_{sk} \quad \forall k \neq j \quad j = 1,2,...c \tag{3.9}$$

3.4. Elección final o hibridación

Como colofón a la fase de clasificación y tras procesar la muestra $\boldsymbol{x}_s$ por medio de los tres clasificadores mencionados, se obtienen una serie de probabilidades de pertenencia a las diferentes clases para el caso de *Bayes*, y se obtienen también los grados de pertenencia para *Fuzzy* y *Vecinos más cercanos Fuzzy*. Es decir, las salidas proporcionadas por los diferentes clasificadores se pueden sintetizar según el esquema proporcionado por la tabla 3.2 dada a continuación.

Clasificador	**Clase 1** (w_1)	**Clase 2** (w_2)		**Clase *c*** (w_c)
Bayes ecuación (3.7)	$p_b(\boldsymbol{x}_s \mid w_1)$	$p_b(\boldsymbol{x}_s \mid w_2)$	…	$p_b(\boldsymbol{x}_s \mid w_c)$
Fuzzy ecuación (3.8)	$\mu(\boldsymbol{x}_s, w_1)$	$\mu(\boldsymbol{x}_s, w_2)$	…	$\mu(\boldsymbol{x}_s, w_c)$
K-NN Fuzzy ecuación (3.9)	$\mu(\boldsymbol{x}_s, w_1)$	$\mu(\boldsymbol{x}_s, w_2)$	…	$\mu(\boldsymbol{x}_s, w_c)$

Tabla 3.2 Salidas al procesar la muestra $\boldsymbol{x}_s$ por los clasificadores

El problema que se plantea en este momento es determinar finalmente cuál será la decisión final tomada respecto en la asignación de $\boldsymbol{x}_s$ a una de las clases, esto se lleva a cabo mediante la combinación. La combinación utiliza la lógica del producto porque funciona adecuadamente en la mayoría de los casos, eligiendo el mejor de los resultados obtenidos de los clasificadores.

3.5. Criterios de clasificación

El teorema de "Nada es gratuito" (No free lunch theorem) dice que no hay razones independientes de contexto y de la aplicación que justifiquen la superioridad de un tipo de clasificador sobre otro. Si un algoritmo parece superior a otro en determinadas circunstancias es consecuencia de su ajuste particular al problema de reconocimiento de patrones, no a su superioridad general como algoritmo. Los aspectos más importantes a la hora de elegir un determinado clasificador son:

- La información a priori.
- La cantidad de patrones para el entrenamiento.

- Las funciones de coste.
- La recompensa.

Para determinar la efectividad durante el proceso de clasificación la medida de bondad utilizada es la **tasa de error** *r***,** o **precisión** definida en la ecuación (3.10) que se mide evaluando el número de casos mal clasificados *m*, respecto a los casos totales evaluados *t*.

$$r = \frac{m}{t} \tag{3.10}$$

Utilizamos esta medida porque poniendo un ejemplo, supongamos que dentro de un conjunto de muestras existen 990 datos de la clase 1 y 10 de la clase 2, el clasificador puede tener fácilmente un sesgo hacia la clase 1. Si el clasificador clasifica todas las muestras como de la clase 1 su precisión será el 99%, pero esto no significa que sea un buen clasificador, pues tuvo el 100% de error en la clasificación de las muestras de la clase 2.

De la misma forma que se calcula el error que se produce durante la clasificación también se calcula su **acierto** *a***,** 3.11 que evalúa la situación contraria aplicando la fórmula al número de casos correctamente evaluados *c*, respecto a los casos totales *t*.

$$a = 1 - r = \frac{c}{t} \tag{3.11}$$

3.6. Métodos de clasificación combinados

Hasta ahora se han estudiado tres clasificadores clásicos catalogados como individuales, pues son capaces de realizan los procesos de aprendizaje y clasificación por sí mismos. En los últimos tiempos se ha abierto una línea de investigación basada en la combinación de clasificadores, son los denominados sistemas combinados o multi-clasificador, cuya finalidad estriba en mejorar los resultados de los clasificadores individuales.

Ho (2002) aborda el tema de la clasificación tratando de buscar las mejores combinaciones posibles, aún a costa del aumento de la complejidad computacional.

Dietterich (2000) sugirió tres tipos de razones por las cuales un clasificador combinado puede resultar más eficaz que un clasificador individual: *estadísticas*, *computacionales* y de *representación*.

a) *Estadísticas*: supongamos que se dispone de varios clasificadores para llevar a cabo la tarea de clasificar un conjunto de datos *X*. Cada uno de ellos producirá ciertos resultados, cabe la posibilidad de que elegido uno al azar, éste no sea exactamente el óptimo para el conjunto de datos dado, con lo que la elección no habrá sido la adecuada. Por el contrario, si seleccionamos varios de ellos, la probabilidad de que la elección sea errónea disminuye, soslayando así el riesgo de elegir un clasificador no apropiado.

b) *Computacionales*: algunos algoritmos de clasificación basados en búsquedas heurísticas pueden llegar a bloquearse por falta de solución. También en los clasificadores basados en métodos de optimización se puede caer en mínimos locales difíciles de superar. La combinación de diferentes clasificadores puede evitar alguna de las situaciones anteriores.

c) *Representación*: se refiere al hecho de que determinados tipos de datos pueden adaptarse mejor a ciertos clasificadores, con el fin de proceder a su clasificación. Por ejemplo, supóngase que los datos *X* se clasifican muy bien con clasificadores de tipo lineal, pues bien, si utilizáramos otro tipo de clasificadores, tales como redes neuronales, que carecen de la mencionada linealidad, es posible que los resultados no sean los deseados. Combinando clasificadores se podrían evitar tales situaciones.

3.6.1. Terminología y taxonomías

Siguiendo la filosofía establecida (Kuncheva, 2004), la combinación de clasificadores puede verse desde distintas perspectivas, tal y como se muestra en la figura 3.3.

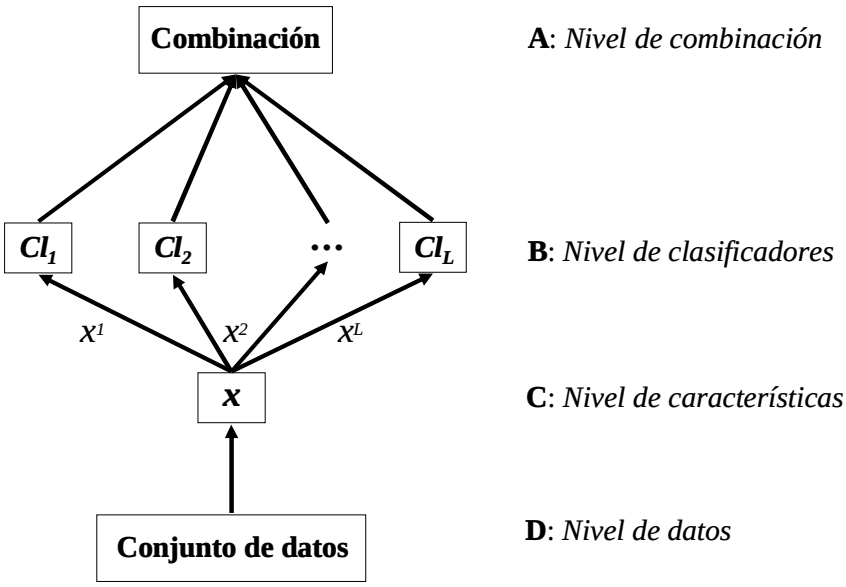


Figura 3.3 Niveles en la combinación de clasificadores

El nivel de datos, D, se refiere a la utilización de diferentes subconjuntos de datos para la clasificación. El nivel C se refiere a la combinación de propiedades o atributos de las muestras a clasificar; ciñéndonos de nuevo al caso de los conjuntos de datos, los atributos serían las propiedades de las muestras, que se combinarían en este nivel. El nivel B hace referencia al uso de distintos clasificadores individuales (Cl_1, Cl_2,…, Cl_L) utilizados para la combinación. Finalmente, el nivel A está relacionado con la estrategia utilizada para llevar a cabo la combinación.

En el presente trabajo, la combinación se aborda desde los niveles A y B, principalmente desde el A. Es en el nivel A donde se han diseñado las estrategias de combinación, que se describen en el capítulo cuatro a partir de los clasificadores individuales descritos en la sección 3.3 de este capítulo. En el capítulo cuatro independientemente de la estrategia de combinación elegida, se utilizarán siempre los tres clasificadores cuyos resultados, de forma independiente, permitirán hacer una optimización a nivel global y sacar conclusiones imposibles de obtener sin la combinación. Desde de este momento, nos centramos exclusivamente en el nivel A.

3.6.2. Métodos para combinar las salidas de los clasificadores

Como se verá posteriormente, la combinación de clasificadores se puede realizar tanto en la fase de entrenamiento como en la de decisión. En la fase de decisión el patrón $\boldsymbol{x}_s$ debe clasificarse como perteneciente a una clase w_j de acuerdo con dicha combinación.

En adelante, por simplicidad y sin pérdida de generalidad se suprime el subíndice del vector representativo de la muestra $\boldsymbol{x}_s$ identificándola como $\boldsymbol{x}$, de este modo se utiliza este vector genérico tanto para las fases de entrenamiento como de decisión.

Refiriéndonos a la figura 3.3 se dispone de un conjunto de L clasificadores en el nivel B, pertenecientes a la categoría de individuales, cuyas salidas deben combinarse en el nivel A. Por tanto, se entiende que desde esta perspectiva, la combinación se realiza en la fase de decisión.

No obstante, antes de abordar la problemática de la combinación distinguimos entre tres tipos de salidas, que pueden ser proporcionadas de forma general por los clasificadores (Xu y col., 1992; Kuncheva 2004):

Tipo 1 (nivel abstracto): cada clasificador Cl_i genera una etiqueta $s_i \in \Omega$ para la clase w_i, con $i = 1,2,\ldots,L$, de forma que para cada muestra a clasificar con vector de entrada $\boldsymbol{x}$, los L clasificadores definen un vector $\boldsymbol{s} = [s_1, s_2, ..., s_L] \in \Omega^L$. En este nivel de abstracción no existe información sobre las etiquetas seleccionadas, ni existen etiquetas alternativas que puedan sugerirse. Por definición, cualquier clasificador es capaz de generar una etiqueta para $\boldsymbol{x}$. Las salidas de este tipo son las más generales de todas.

Tipo 2 (nivel de rango): la salida de cada clasificador Cl_i es un subconjunto de Ω, con las opciones de clase ordenadas según el grado de posibilidad de ser la etiqueta correcta (Ho y col., 1994; Tubbs y Alltop, 1991). Este tipo es deseable para problemas que requieran un número elevado de clases.

Tipo 3 (nivel de medida): cada clasificador Cl_i genera como salida un vector c-dimensional $\left[d_{i,1},...,d_{i,j},...,d_{i,c}\right]^T$, cuyas componentes son en realidad funciones del vector de entrada $\boldsymbol{x}$ representativo de la muestra a clasificar. Para simplificar la notación se emplea $d_{i,j}$ en lugar de $d_{i,j}(\boldsymbol{x})$, que sería el correcto. El valor $d_{i,j}$ representa el grado de apoyo dado por el clasificador Cl_i para la muestra $\boldsymbol{x}$ respecto de la clase w_j. A este tipo pertenecen los clasificadores individuales estudiados en la sección 3.3. Como se ha mencionado previamente, en este trabajo de investigación la combinación se realiza en el nivel A, utilizando las salidas proporcionadas por los clasificadores que participan en la combinación; en la tabla 3.2 se sintetizan dichas salidas para la fase de decisión.

El presente trabajo se centra en los clasificadores de tipo tres, para los que a continuación se exponen diversos métodos clásicos, que sirven para validar y comparar las estrategias de combinación utilizadas en las salidas de los clasificadores.

3.6.2.1. Votación Mayoritaria

Se trata del famoso método de consenso, de ahí su nombre original en terminología inglesa de "*majority voting*". La combinación se puede llevar a cabo mediante tres enfoques diferentes: unanimidad, mayoría simple y pluralidad. En el caso de la unanimidad, todos los clasificadores coinciden en su decisión, la mayoría simple se refiere a la coincidencia del 50% más uno y finalmente la pluralidad consiste en seleccionar la clase sobre la que coinciden como la correcta el mayor número de clasificadores.

Supóngase que las etiquetas de salida de los clasificadores vienen dadas como vectores c-dimensionales binarios $[d_{i,1}, d_{i,2}, ..., d_{i,c}]^T \in \{0,1\}^c$, $i = 1,...,L,$ donde $d_{i,j}$ es igual a 1 si el clasificador Cl_i etiqueta la muestra $\boldsymbol{x}$ como perteneciente a w_j y cero en caso contrario. El voto plural llega a ser una decisión de conjunto para la clase w_k si,

$$\sum_{i=1}^{L} d_{i,k} = \max_{j=1}^{c} \sum_{i=1}^{L} d_{i,j} \tag{3.13}$$

El voto plural se conoce en sentido amplio como votación mayoritaria, su planteamiento proviene del contexto electoral habiendo sido utilizado en diferentes trabajos de investigación, entre los que destacan Lin y col. (2003), Kittler y col. (1998), Lam y Suen (1997) o Battiti y Colla (1994).

Una de las variantes que se suelen introducir en este tipo de combinación es la del voto ponderado, en el sentido de que cada clasificador posee un peso específico que se tiene en cuenta a la hora de tomar la decisión (Kuncheva, 2004).

3.6.2.2. Combinación mediante funciones, máximo, mínimo y media

Siguiendo con los clasificadores de tipo tres, en esta sección se introducen una serie de funciones para combinar las salidas de los clasificadores.

Sin pérdida de generalidad, se supone que las salidas generadas por cada clasificador toman valores continuos en el intervalo [0,1]. Recordemos que $d_{i,j}(\boldsymbol{x})$ expresa el grado de apoyo dado por el clasificador Cl_i a la muestra $\boldsymbol{x}$ según su pertenencia a la clase w_j.

Cuanto mayor sea su valor para una determinada clase w_j, tanto más probable resulta el hecho de que dicha muestra pertenezca a esa clase. Volviendo sobre la tabla 3.2, donde se proporcionan las salidas de los tres clasificadores expresados, el mayor grado de apoyo se corresponde con los valores máximos del grado de pertenencia y de las probabilidades.

Las salidas de los L clasificadores para una muestra dada se pueden organizar en forma de matriz, conocida como *perfil de decisión*, $PD(\boldsymbol{x})$,

$$PD(\boldsymbol{x}) = \begin{bmatrix} d_{1,1}(\boldsymbol{x}) & \text{L} & d_{1,j}(\boldsymbol{x}) & \text{L} & d_{1,c}(\boldsymbol{x}) \\ \text{M} & & \text{M} & & \\ d_{i,1}(\boldsymbol{x}) & \text{L} & d_{i,j}(\boldsymbol{x}) & \text{L} & d_{i,c}(\boldsymbol{x}) \\ \text{M} & & \text{M} & & \\ d_{L,1}(\boldsymbol{x}) & \text{L} & d_{L,j}(\boldsymbol{x}) & \text{L} & d_{L,c}(\boldsymbol{x}) \end{bmatrix} \tag{3.14}$$

Las filas se refieren a las salidas proporcionadas por cada clasificador Cl_i con respecto a las diferentes clases w_j, con $j = 1,\ldots, c$; cada columna representa el apoyo proporcionado por los L clasificadores $Cl_1,\ldots, Cl_L$ para que $\boldsymbol{x}$ pertenezca a la clase w_j. Los métodos descritos a continuación utilizan la matriz $PD(\boldsymbol{x})$ para encontrar la salida resultante de la combinación, que representa el apoyo total dado a $\boldsymbol{x}$ por la combinación de clasificadores respecto a su pertenencia a una clase determinada.

Existen dos enfoques para realizar esta tarea, en primer lugar se puede utilizar el hecho de que los valores en cada columna j de la matriz PD representan los apoyos individuales que cada clasificador otorga a $\boldsymbol{x}$ para la clase w_j, la combinación de estos valores será el apoyo total proporcionado por los clasificadores combinados. La combinación de métodos que utilizan una columna de $PD(\boldsymbol{x})$ se denominan "class conscious" en Kuncheva y col. (2001), empleada para realizar el análisis comparativo con los métodos combinados propuestos en este trabajo de investigación.

Alternativamente, se puede suprimir el concepto de clase y tratar los valores $d_{i,j}(\boldsymbol{x})$ como características en un nuevo espacio de características, denominado *espacio de características intermedio*. La decisión final se realiza mediante otro clasificador que

toma este espacio de características intermedio como entrada y genera una etiqueta de clase a la salida.

En Kuncheva y col. (2001) este tipo de métodos se denominan “class indifferent”; de esta forma se pueden construir diversas capas de clasificadores. El principal problema de este segundo tipo de clasificadores combinados estriba en cómo entrenar dichas arquitecturas para estar seguros de que el aumento de la complejidad mejore la eficacia de la combinación, lo que da lugar a realizar la combinación tanto en la fase de entrenamiento como de decisión. Estos motivos han servido para descartar este tipo de estrategias entre las sugeridas en el capítulo cuatro, pues en vistas a mejorar la eficiencia del proceso, las estrategias de combinación propuestas en este trabajo se realiza únicamente en la fase de decisión.

A continuación se describen una serie de clasificadores combinados que Kuncheva (2004) llama *no entrenables*, sirven para expresar que la combinación no tiene parámetros extra que deban ser obtenidos mediante entrenamiento durante la fase de decisión.

El apoyo total, mencionado previamente, proporcionado por los L clasificadores, $Cl_1,\ldots,$ Cl_L, para que $\boldsymbol{x}$ pertenezca a la clase w_j es el resultado de la combinación de los valores en la j-ésima columna de $PD(\boldsymbol{x})$ según la expresión (3.15).

$$\gamma_j(\boldsymbol{x}) = \Im\left[d_{1,j}(\boldsymbol{x}),\ldots,d_{i,j}(\boldsymbol{x}),\ldots,d_{L,j}(\boldsymbol{x})\right] \tag{3.15}$$

donde $\Im$ define la función de combinación.

La decisión sobre la clase a la que pertenece $\boldsymbol{x}$ se determina como el índice j correspondiente al máximo de los $\gamma_j(\boldsymbol{x})$ valores, según la siguiente regla:

$$\boldsymbol{x} \in w_j \text{ si } \gamma_j(\boldsymbol{x}) > \gamma_k(\boldsymbol{x}) \ \forall k \neq j, \quad \text{donde } j,\ k = 1,\ldots,c$$

La función $\Im$ puede elegirse de diferentes maneras, las elecciones más comúnmente utilizadas vienen dadas por las ecuaciones (3.16) y (3.17) (Kittler y col., 1998).

Media aritmética

$$\gamma_j(\boldsymbol{x}) = \frac{1}{L}\sum_{i=1}^{L} d_{i,j}(\boldsymbol{x}) \tag{3.16}$$

Mínimo/Máximo

$$\gamma_j(\boldsymbol{x}) = \min_i \{d_{i,j}(\boldsymbol{x})\} \quad \gamma_j(\boldsymbol{x}) = \max_i \{d_{i,j}(\boldsymbol{x})\} \tag{3.17}$$

En resumen, para la comparación de los resultados obtenidos con los métodos de combinación propuestos en el capítulo cuatro, de los métodos anteriores se seleccionan los siguientes: votación mayoritaria, media aritmética, máximo, mínimo.

Capítulo 4

4. Análisis de resultados

4.1. Objetivos del análisis

En este punto se trata de verificar y validar la estrategia propuesta. Al tratarse de un clasificador compuesto que realiza un proceso de aprendizaje previo a la clasificación, se plantean los siguientes tres objetivos de verificación:

- Comportamiento de las distintas estrategias aplicadas por el método.
- Comportamiento del método frente a clasificadores simples y combinados.
- Comportamiento del método a medida que se incrementa el aprendizaje.

Para el primer objetivo se eligen los clasificadores propuestos, Bayes, Fuzzy y Nearest Neighbours Fuzzy, y se buscan como estrategias de combinación las técnicas del Máximo, Mínimo, Media Aritmética y Votación mayoritaria. Además en la fase inicial, se aplican distintos métodos para la reordenación de las muestras y el incremento del rendimiento de los algoritmos, que con la Estratificación y el ten-fold Cross-Validation.

Para el segundo objetivo se seleccionan los resultados de los clasificadores combinados con nuestros métodos aplicando las reglas de la sección 3.6, y se comparan con los mejores resultados de un amplio número de tablas extraídas de la bibliografía.

Finalmente para la consecución del tercer objetivo se organizan dos etapas diferenciadas, que aclaran el proceso llevado a cabo para el diseño del método.

4.2. Descripción de las bases de datos utilizadas

Se dispone de un conjunto de 22 bases de datos adquiridas durante el mes de Noviembre de 2008 del repositorio *UCI Learning Machine Repository* (Asunción y Newman, 2009) Se trata de bases de datos numéricas cuya dimensión varía en función del número de atributos que presente cada conjunto de muestras. Las bases de datos se adquirieron en diferentes días a medida que iban aumentando los objetivos, de tal modo que se amplió progresivamente el conjunto de bases de datos debido a que cada una ofrecía diferentes características que permitían realizar pruebas de distinta índole.

Las 22 bases de datos sufrieron el mismo proceso de evaluación para observar su comportamiento y así poder comparar los resultados obtenidos con los existentes en la literatura para esas mismas bases. El método aplicado es supervisado porque cada conjunto de muestras no sólo tiene un número de atributos diferente a las demás, sino que el número de clases debe ser especificado antes de llevar a cabo el aprendizaje.

En la tabla 4.1 podemos encontrar la información resumida de los datos utilizados, determinando el número de muestras que evalúan ***n***, el número de clases ***c*** y el número de atributos ***a*** que tiene cada base de datos. Además se ha añadido una pequeña descripción con información detallada de cada una de ellas en la columna correspondiente.

#	*Categorías de datos*	*a*	*c*	*n*	*Descripción*
1	**Breast Cancer Wisconsin**	**12**	**2**	**569**	Las características son computerizadas de imágenes digitales con el FNA (fine needle aspirate) de la masa mamaria. Describen las características del núcleo de la célula presentado en la imagen.
2	**Bupa liver disorders**	**6**	**2**	**345**	BUPA Medical Research Ltd.: resultados de análisis de sangre, cuya presencia parece que sean los causantes de desórdenes en el funcionamiento del hígado.
3	**Australian**	**15**	**2**	**6650**	Sirve para aplicaciones de tarjetas de crédito. Es interesante porque mezcla atributos con valores pequeños y grandes. Todos los valores reales se han cambiado por símbolos por seguridad.
4	**Ozone**	**73**	**2**	**2536**	Es una colección de muestras ordenadas cronológicamente para establecer mediante sus atributos si un día es normal o es propenso a elevados índices de ozono.
5	**Pima Indians**	**8**	**2**	**768**	National Institute of Diabetes and Digestive and Kidney Diseases (V. Sigillito): parámetros de herencia en la tribu India Pima.
6	**Ionosphere**	**34**	**2**	**351**	Sistema cuyo objetivo es recoger la estructura en la ionosfera mediante antenas de alta frecuencia. La señal atraviesa esta capa y el radar devuelve la calidad de la estructura.
7	**Heart SpectF**	**44**	**2**	**267**	Diagnóstico cardíaco por imágenes. Cada paciente es clasificado en las categorías normal o anormal.
8	**Heart**	**13**	**2**	**270**	Mediante los atributos de cada muestra (cada pacientes) se determina si son propensos a tener una enfermedad cardíaca.
9	**Sonar**	**60**	**2**	**208**	Cada atributo representa la energía en una banda de frecuencia tomada en distintos ángulos en un periodo de tiempo, y clasifica objetos como rocas y metálicos.
10	**Survival**	**3**	**2**	**306**	El conjunto de datos es de un estudio de pacientes que entre 1958 y 1970 sobrevivieron a operaciónes de cáncer de pecho.
11	**Hepatitis**	**19**	**2**	**155**	Utiliza atributos como características de los pacientes, y en función de los valores determinan si tienen hepatitis.

#	Categorías de datos	a	c	n	Descripción
12	**Iris**	**4**	**3**	**150**	R.A. Fisher & M. Marshall: diferentes clases de lirios (plantas).
13	**Wine**	**13**	**3**	**178**	Forina, M. et al, PARVUS & S. Aeberhard: análisis químicos de distintas clases de vinos.
14	**Ann Thyroid**	**21**	**3**	**3428**	El problema es determinar si un paciente es hipotiroideo.
15	**New Thyroid**	**5**	**3**	**215**	Los test realizados sirven para intentar predecir si los pacientes tienen tiroides, distinguiendo entre tres tipos.
16	**Waveform Noise**	**20**	**3**	**825**	Son valores numéricos que evalúan el tipo de ondas.
17	**Balance Scale**	**4**	**3**	**625**	Utilizada para el modelado de resultados psicológicos, pudiendo indicar el balanceo de las muestras o su equilibrio.
18	**Vehicle**	**18**	**4**	**946**	Turing Institute, Glasgow, Scotland. Clasificación de vehículos por medio de su silueta, para lo que se ofrecen distintos ángulos para cada muestra.
19	**Lymphography**	**18**	**4**	**148**	University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Gracias a M. Zwitter y M. Soklic por suministrar esta base de datos. Usado para diagnóstico de tratamientos.
20	**Glass**	**10**	**6**	**214**	B. German, V. Spiehler from Central Research Establishment & Diagnostic Products Corporation: diferentes propiedades de cristales.
21	**Satimage**	**36**	**6**	**432**	Department of Statistics and Data Modelin, University of Strathclyde, Glasgow. Utiliza imágenes digitales de una misma escena con distintos espectros.
22	**Shuttle**	**9**	**7**	**1934**	La NASA ha permitido el uso de esta base de datos numérica, que está ordenada según fue extraída aunque una parte del total de los datos ha sido eliminada para la validación.

Tabla 4.1: Descripción detallada de las bases de datos utilizadas

4.3. Método utilizado para la aplicación

La implementación se ha realizado en Matlab, que es un producto de la compañía The Mathworks http://www.mathworks.com/products/matlab/. Se trata de un lenguaje científico interpretado. Debido a este hecho el coste computacional de los procesos implementados en Matlab es superior a otros lenguajes de distinta naturaleza. No obstante, el objetivo de este trabajo no consiste en estudiar los tiempos de cómputo sino el comportamiento del método propuesto.

4.4. Análisis de resultados

4.4.1. Determinación del número de clases

El primer objetivo que se plantea consiste en determinar el número de clases que van a poseer los datos de entrenamiento.

Debido a que las muestras utilizadas son muy heterogéneas fue posible realizar distintas pruebas, comprobando el correcto funcionamiento de los algoritmos en conjuntos de datos que contenían de dos a siete clases.

Cuando la partición tiene un número de clases bajo, significa que existen muchas muestras que son asignadas a clases de una forma muy forzada, lo que se traduce en que las mismas se sitúan lejos de los centros de los clústeres y por tanto generan valores de los grados de pertenencia bajos, según el método de agrupamiento borroso.

En el caso de un número de clases alto, existen muestras que se asignan a un clúster pero que en realidad podrían estar asignadas a cualquier otro clúster próximo. Esto se traduce en el hecho de que existirán muestras con elevados grados de pertenencia para clases diferentes.

Si hablásemos de 1-NN existe un problema de ruido cuando una muestra debe pertenecer a una clase, y tiene como más cercano un vecino de una clase diferente, como sucede en el caso de las clases solapadas. Esto hace que automáticamente se asigne la muestra a la clase equivocada.

En el otro extremo está el caso de que de forma empírica, no puede utilizarse el algoritmo de Vecinos más cercanos Fuzzy con más vecinos que el número total de muestras, aunque esto significa que utilizar tantos vecinos como muestras en la evaluación dará como porcentaje de aciertos el porcentaje de muestras de la clase mayoritaria. Significa que cualquier muestra será asociada a la clase mayoritaria del conjunto, y en consecuencia esto conlleva a un resultado nada deseable.

Por esta razón se ha buscado un número adecuado de vecinos mediante el proceso de prueba y error. Las pruebas se realizaron para estimar el comportamiento de la técnica de clasificación y así predecir nuevas situaciones; para ello se utilizó la fórmula de la precisión de la ecuación (3.11).

Se tomaron ciertas precauciones en la extracción de un número adecuado de vecinos ya que el principal objetivo era identificar un clasificador con la validez más general

posible, que pudiera utilizarse de forma estándar en todas las situaciones y con cualquier base de datos, asegurando un acierto considerable en las muestras evaluadas, aunque el óptimo final requiriera de ciertos ajustes en la configuración de parámetros de entrada. Mediante las pruebas se llegó a la conclusión de que siete era el número óptimo de vecinos para la mayoría de muestras de este trabajo. Esto dio lugar a que la evaluación de las muestras con *K-NN* Fuzzy utilizase una $K = 7$ en todas las bases de datos analizadas.

4.4.2. Resultados obtenidos

El proceso que se llevó a cabo con las 22 bases de datos de la tabla 4.1, permitió la elaboración de la tabla 4.2 donde se muestran los resultados obtenidos con nuestro diseño de clasificador, éstos fueron comparados con el mejor resultado para cada base de datos obtenidos de 21 artículos estudiados en la bibliografía.

BBDD	Num clases	FUZZY			FUZZY-KNN			BAYES			Mejor Paper
		EST	SIN EST	LOO -CV	EST	SIN EST	LOO -CV	EST	SIN EST	LOO -CV	
CancerWisconsin	2	82.73	85.77	77.05	92.62	91.92	92.79	93.68	90.15	87.60	96 (1)
Bupa	2	64.76	59.24	62.46	68.95	61.43	65.51	75.15	70.89	71.37	92 (2)
Australian	2	76.09	73.04	65.51	94.74	65.36	69.85	89.98	89.57	90.15	NA
Ozono	2	85.6	84.13	85.25	90.87	91.13	91.19	92.87	87.34	89.08	NA
Pima	2	61.58	61.57	62.84	72.40	73.44	72.13	73.14	71.02	73.27	77 (6)
Ionosphere	2	69.44	67.59	63.82	86.57	84.63	83.76	85.47	84.38	80.15	93 (3)
HeartSpectF	2	68.81	65.23	79.02	75.65	71.90	75.28	80.05	78.09	77.47	NA
HeartSpect	2	63.01	61.94	62.92	81.43	71.43	76.40	72.20	70.08	69.14	NA
Heart	2	80	78.52	62.92	64.44	65.90	63.70	75.48	76.17	76.89	80 (3)
Sonar	2	68.70	65.26	62.40	72.23	72.08	73.08	71.14	70.43	72.08	80 (4)
Survival	2	71.27	69.09	73.53	71.28	73.18	70.59	74.18	72.78	70.15	NA
Hepatitis	2	78.56	70	90.62	84.28	90	92.19	90.89	90.67	91.75	NA
Iris	3	88.10	88	86.66	97.33	96.67	96.67	95.14	92.80	95.04	98(2)
BreastCancerW	3	72.28	72.04	70.48	72.33	71.44	69.96	72.89	70.37	71.05	96 (1)
Wine	3	73	71.11	72.58	74.17	77.50	76.97	75.14	73.38	74.07	100 (2)
Ann_thyroid	3	80.15	79.42	84.37	91.60	91.66	93.21	91.61	91.03	92.33	97 (2)
New_thyroid	3	91.95	87.73	89.77	94.73	94.09	93.95	94.63	92.31	93.14	97
WaveformNois	3	73.51	72.52	72.55	79.55	79.39	83.76	80.55	78.41	81.57	NA
BalanceScale	3	62.61	61.30	60.22	75.86	77.83	87.36	78.86	76.91	79.80	NA
Vehicles	4	65.12	64.22	65.18	64.99	64.55	64.77	64.09	63.85	64.51	80 (3)
Lymphography	4	71.11	71.20	71.02	74.46	76.82	79.05	76.46	71.23	75.55	NA

Glass	6	73.39	71.15	74.14	73.86	74.77	78.22	80.08	78.15	74.37	90 (5)
Satimage	6	82	80.13	75.23	82.53	82.40	84.26	83.53	80.89	81.12	NA
Segment	7	92.45	92	95.11	96.49	96.32	96.58	95.79	92.30	93.22	NA
Shuttle	7	96.32	94.82	97.69	99.56	99.45	99.56	98.48	98.12	99.04	NA

Leyenda indicando el artículo al que se hace referencia en la síntesis anterior	
(1)	Gu y Wu, 2008
(2)	Takagi y col., 2004
(3)	Tsipouras y col., 2008
(4)	Fakhrahmad y col., 2007
(5)	Tsai y col., 2008
(6)	Saastamoinen y Ketola, 2006
NA	Ninguno de los artículos estudiados han sacado resultados de esta base de datos

Tabla 4.2: Resultados obtenidos y comparación con otros métodos de la bibliografía

El proceso de verificación y validación se lleva a cabo de la siguiente manera:

- Dados los resultados extraídos mediante los distintos algoritmos, se aplica la combinación de los métodos propuestos contemplando la posibilidad de realizar diferentes hibridaciones.
- Para determinar cuál es la propuesta más efectiva debemos verificar su comportamiento frente a los cuatro métodos clásicos utilizados en la literatura, descritos en la sección 2.3 y la tabla 2.1, que son la Regla de máximo (MA), Regla de mínimo (MI), Regla de media (RM) y Votación mayoritaria (VM).
- Finalmente se presentan los resultados obtenidos de la combinación de los métodos Bayes, Fuzzy y Vecinos más cercanos Fuzzy, y serán comparados con los resultados extraídos de la literatura.

Esta comparación permitió la elaboración de las siguientes gráficas:

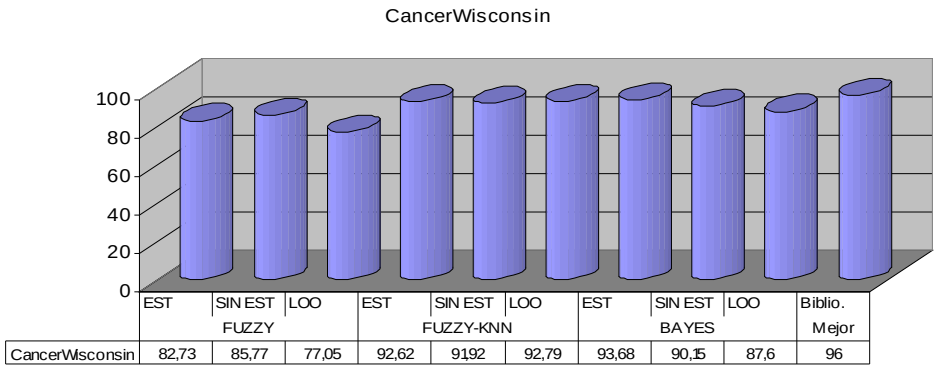


Figura 4.1: Resultados para CancerWisconsin

CancerWisconsin presenta elevados resultados en prácticamente la totalidad de los algoritmos utilizados, superando el 90% siempre que se utilizan los Vecinos más cercanos con Fuzzy.

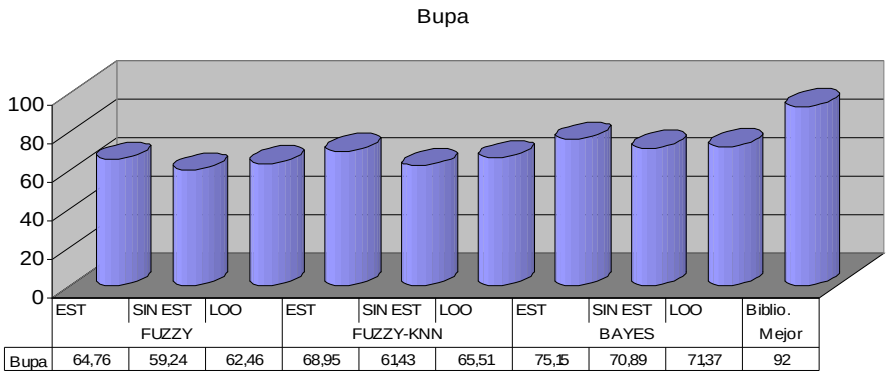


Figura 4.2: Resultados para Bupa

Los resultados obtenidos para ***Bupa*** con el clasificador Bayes presentan un porcentaje superior al 70% en todo momento. Ciertamente no podría compararse con el obtenido en las bases de datos de la bibliografía, sin embargo, hay que destacar que el estudio que proporcionó ese 92% se realizó con tan sólo cuatro bases de datos, lo que generalmente va asociado a un ajuste de parámetros para beneficiar las muestras estudiadas. También indicar que el resto de los artículos ofrecen aproximadamente un 70% de acierto en esta base de datos.

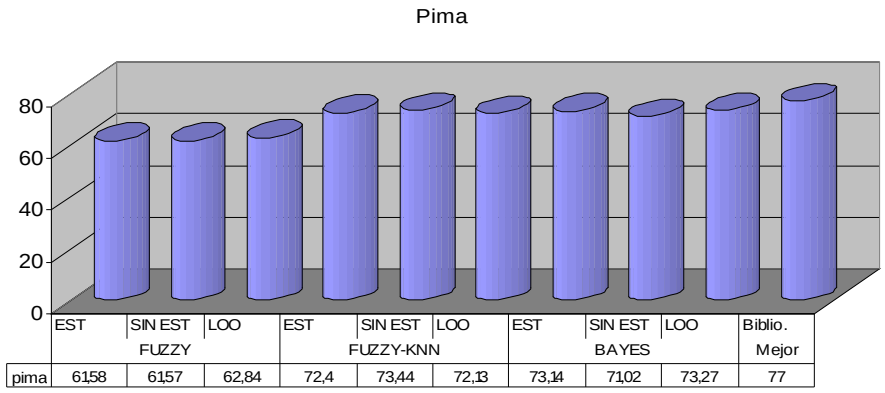


Figura 4.3: Resultados para Pima

Entre los clasificadores utilizados con *Pima* vuelven a destacar el Vecinos más cercanos Fuzzy y el algoritmo Bayes, superando en algunos casos el 73% de acierto, casi comparable al mejor resultado de toda la bibliografía estudiada con un 77%.

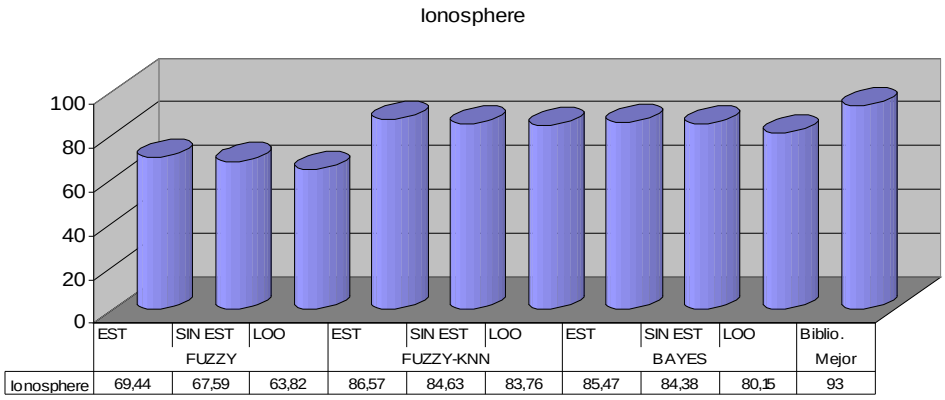


Figura 4.4: Resultados para Ionosphere

En el caso de *Inosphere*, sobretodo destacan los malos resultados obtenidos con la técnica Fuzzy. Sin embargo, con el Vecinos más cercanos Fuzzy nuevamente los resultados son muy buenos superando el 86% cuando se utiliza la estratificación. En la bibliografía destaca un 93%.

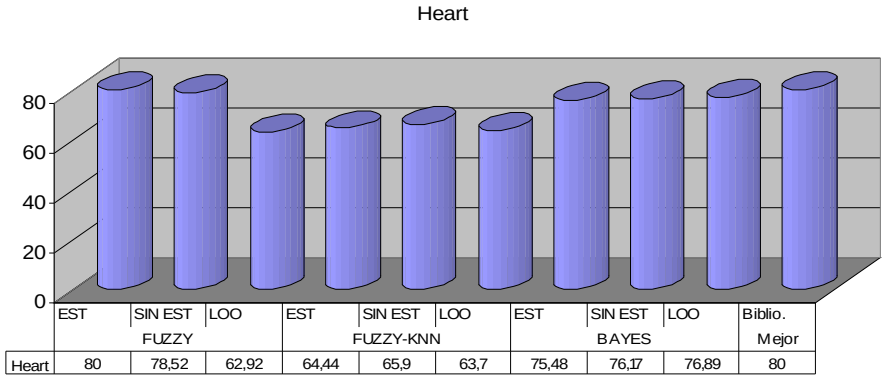
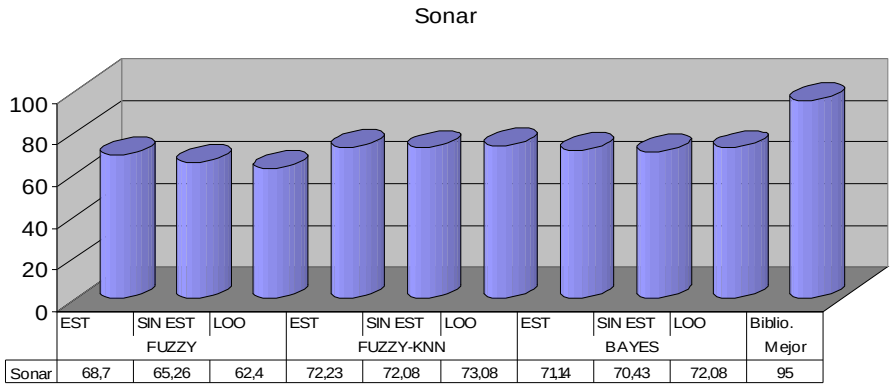


Figura 4.5: Resultados para Heart

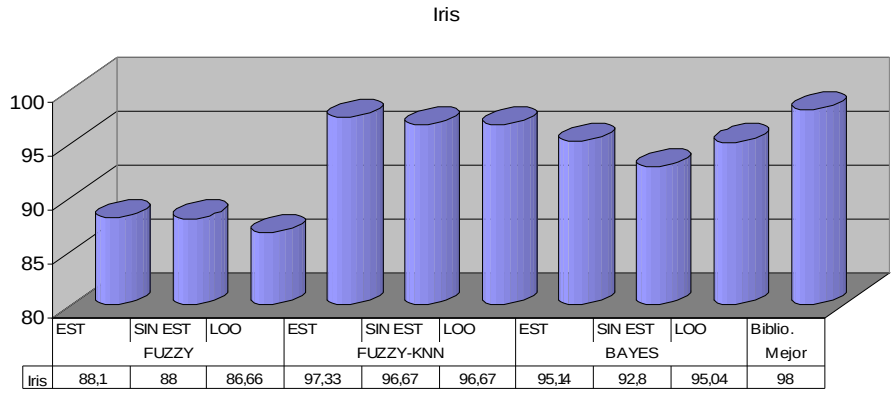
Como se demuestra a lo largo del estudio, no existe un método universal que destaque sobre los demás en cuanto a resultados obtenidos, con *Heart* el mejor resultado ha sido un 80% y se ha obtenido utilizando Fuzzy con estratificación, siendo igual al mejor ofrecido por los datos bibliográficos.



	FUZZY			FUZZY-KNN			BAYES			
	EST	SIN EST	LOO	EST	SIN EST	LOO	EST	SIN EST	LOO	Biblio. Mejor
Sonar	68,7	65,26	62,4	72,23	72,08	73,08	71,14	70,43	72,08	95

Figura 4.6: Resultados para Sonar

Otra vez, en esta ocasión con la base de datos *Sonar*, se han obtenido los mejores resultados utilizando Vecinos más cercanos Fuzzy, superando en todos los casos el 72%. No es comparable al 95% de la bibliografía, sin embargo y como pasó con Bupa, el resto de los artículos de la bibliografía no superan el 80%, lo que hace pensar que tan elevado resultado se debe a un buen ajuste en la parametrización.



	FUZZY			FUZZY-KNN			BAYES			
	EST	SIN EST	LOO	EST	SIN EST	LOO	EST	SIN EST	LOO	Biblio. Mejor
Iris	88,1	88	86,66	97,33	96,67	96,67	95,14	92,8	95,04	98

Figura 4.7: Resultados para Iris

Iris es una de las bases de datos más utilizadas a lo largo de la bibliografía debido a sus características y buen comportamiento. En los resultados el clasificador utilizado refleja un elevado índice de acierto con casi un 97%, muy próximo al 98% que es el valor máximo que se ha obtenido entre los artículos manejados.

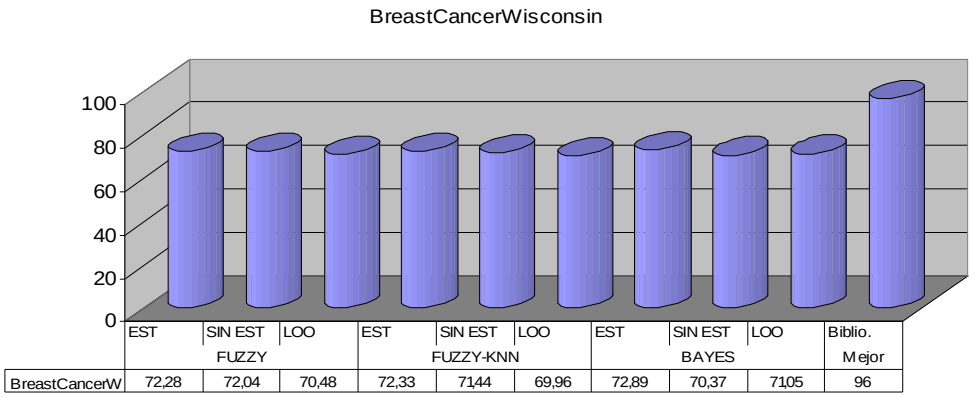


Figura 4.8: Resultados para Breast Cancer

En *Breast Cancer Wisconsin* nuestros resultados están por debajo del nivel máximo obtenido en la bibliografía, sin embargo vuelve a demostrarse el comportamiento estable de nuestro algoritmo, que con valores optimistas supera en la mayoría de las ocasiones el 70% de acierto.

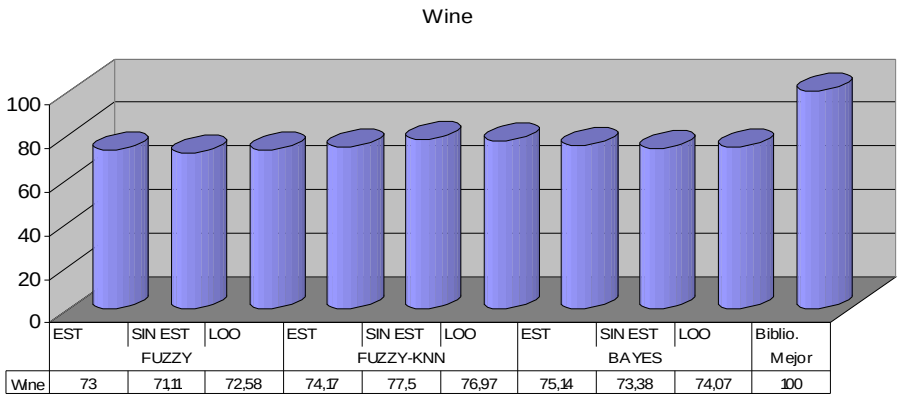


Figura 4.9: Resultados para Wine

En *Wine* vuelve a demostrarse un elevado porcentaje de acierto superando el 75% de acierto en varios casos, principalmente en Vecinos más cercanos Fuzzy, sin embargo nuestros resultados están por debajo del 100% de acierto que se obtiene en el mejor de los artículos estudiados en la bibliografía.

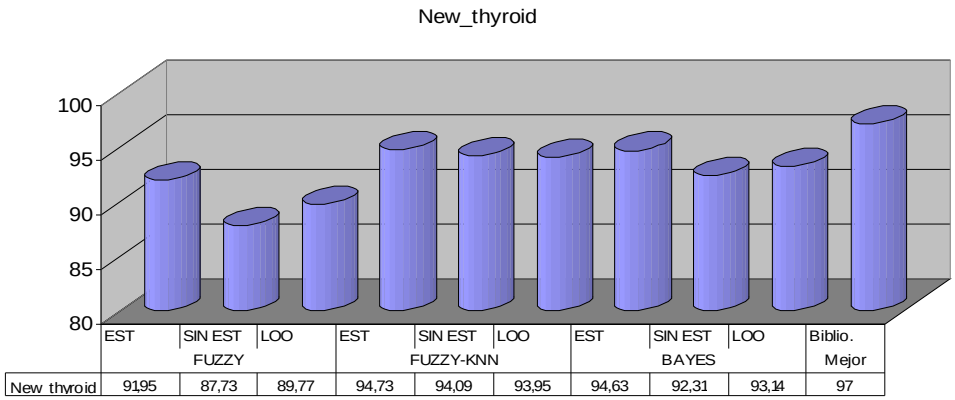


Figura 4.10: Resultados para New thyroid

New thyroid es una base de datos que ha ofrecido un elevado porcentaje de acierto en todas las pruebas realizadas.

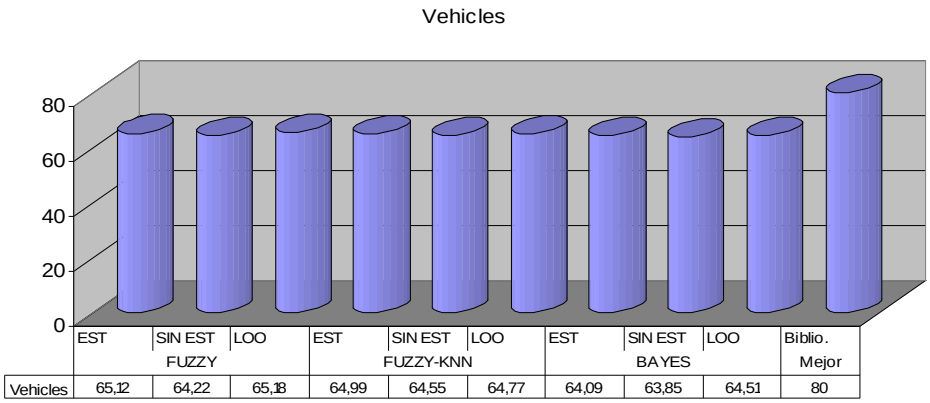


Figura 4.11: Resultados para Vehicles

En la base de datos de *Vehicles* podemos observar que la técnica basada en los Vecinos más cercanos con Fuzzy ofrece casi un 65% de acierto, que es bastante optimista y está relativamente próximo al 80 % de la mejor de las bases de datos de la bibliografía.

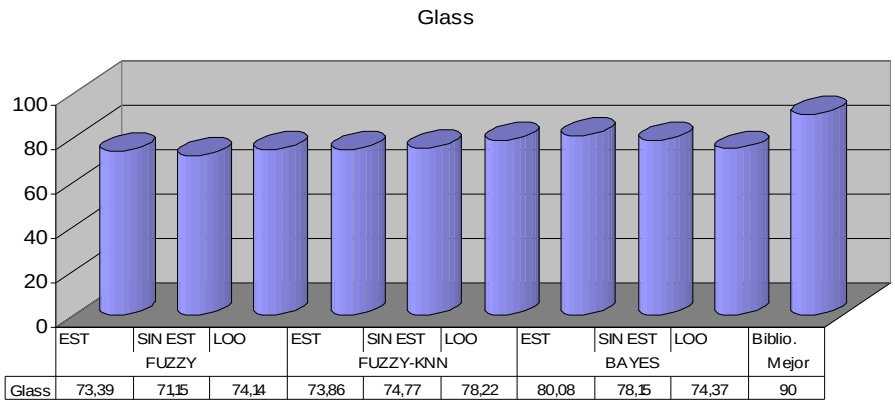


Figura 4.12: Resultados para Glass

Pese a las siete clases que tiene la base de datos de *Glass*, el mejor artículo de la bibliografía refleja un 90% de acierto siendo esto una excepción, ya que el resto de artículos estudiados generalmente no pasan del 80%. Los resultados obtenidos en este trabajo reflejan que el mejor valor supera el 80% de acierto.

El estudio se ha realizado sobre el resto de conjuntos de muestras descritos en la tabla 4.1, que han generado un amplio número de resultados. Aunque se han obtenido destacables porcentajes de acierto para algunas de las bases de datos mencionadas, sin embargo sólo se han mostrado las gráficas con los resultados más relevantes. El conjunto de resultados de todas las bases estudiadas se muestran en la tabla 4.2.

4.4.3. Evaluación de los resultados obtenidos

Entre todas las pruebas realizadas se ha comprobado que todos los métodos reflejan un porcentaje de acierto muy similar, aunque destacan ligeramente el método de Vecinos más Cercanos Fuzzy cuando utiliza la estratificación y el Bayesiano, siendo el método Fuzzy con estratificación y sin ella, el que generalmente peores resultados ofrece.

Al elegir entre 22 artículos aquél que ofreciera el mejor resultado para realizar el análisis comparativo, se observa que el valor proporcionado en el artículo correspondiente en cada base de datos, es a menudo superior a los obtenidos por nuestra estrategia, sin embargo no nos centramos en la búsqueda del mejor resultado en términos de una base de datos determinada, sino en ofrecer valores competitivos para el mayor número de bases de datos posibles.

Mediante la combinación se elige en todo momento el mejor resultado de entre los métodos estudiados, ayudando a que la clasificación realizada responda eficientemente a las expectativas deseadas, pues produce unos valores comparables con cualquier resultado obtenido en los artículos de la bibliografía, para las bases de datos estudiadas.

Capítulo 5

5. Conclusiones y trabajo futuro

A la vista de los resultados mostrados en la tabla 4.2, algunos de los cuales se muestran gráficamente, se pueden extraer las siguientes concusiones:

1. Continúa sin existir un método válido con carácter general para cualquier tipo de datos y de forma totalmente automática.
2. Para conseguir buenos resultados para un tipo de datos concretos, no se está exento del previo y necesario ajuste de parámetros, lo que dificulta la automatización de los procesos de aprendizaje y clasificación.
3. Las estrategias combinadas con las técnicas de clasificación utilizadas en este trabajo muestran unos resultados relativamente estables en todas las bases de datos evaluadas, ofreciendo fiabilidad respecto a los demás algoritmos. En unos casos la estabilidad aparece asociada a buenos resultados y en otros casos a no tan buenos con respecto a los otros algoritmos estudiados.
4. Los mejores resultados dentro de los experimentos realizados en este trabajo se obtienen con el método Fuzzy con Vecinos más cercanos, utilizando la técnica de Leave-One-Out. Aunque existen casos en los que su resultado empeora levemente, por norma muestra el mejor comportamiento en términos de porcentaje de errores y una constante estabilidad con un relativo elevado número de aciertos frente a todos los demás.
5. A medida que el número de muestras aumenta se verifica que los resultados son mejores (menor número de errores). Esto es debido a que en el aprendizaje del proceso es tanto mayor cuantas más muestras se procesan durante el entrenamiento.
6. La calidad y complejidad de los atributos de las muestras es de vital importancia a la hora de obtener mejores resultados. Sin embargo, debido a que tratamos con muestras de distinta complejidad, no ha podido demostrarse que el aumento de clases afecte negativamente al porcentaje de aciertos.
7. Los porcentajes de acierto obtenidos no constituyen el óptimo, sin embargo haciendo la media de todos los resultados obtenidos para todas las bases de datos

se ha calculado un acierto del 78,93%, que si bien resulta un tanto baja, en términos globales puede considerarse aceptable.

8. La aplicación de técnicas previas de preparación de los datos para mejorar el rendimiento es una ventaja que puede utilizarse en un momento dado para elevar los porcentajes de acierto.
9. En trabajos futuros el objetivo es claro, debe plantearse la búsqueda de un algoritmo que con un ajuste inicial de sus parámetros, produzca de forma estable porcentajes de aciertos tan elevados como sea posible para hacer desaparecer el debate sobre el clasificador a elegir.
10. La cantidad de clasificadores aplicables en la combinación puede aumentarse para añadir algún otro clasificador junto a los estudiados, siempre que mostrara resultados aceptables y estables para un gran número de bases de datos. Es algo que queda abierto para su estudio en el futuro.

Bibliografía

1. Alatas B. and Akin E. (2005). “FCACO: Fuzzy Classification Rules Mining Algorithm with Ant Colony Optimization”. ICNC 2005, LNCS 3612, pp. 787 – 797.
2. Alippi C., Fuhrman M. and Roveri M. (2008). “k-NN classifiers: investigating the k=k(n) relationship”, Technical Report.
3. Amari S., Murata N., Müller K., Finke M. and Yang H. (1996). “Statistical theory of overtraining – is cross-validation asymptotically effective?”. Advances in neural information processing systems, 8:176–182.
4. Asuncion, A. and Newman, D.J. (2009). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science, available on-line http://www.ics.uci.edu/~mlearn/MLRepository.html
5. Balasko B., Abonyi J. and Feil B. (2008). “Fuzzy Clústering and Data Analysis Toolbox for Use with Matlab”. Veszprem University, Hungary.
6. Battiti, R. and Colla, A.M. (1994). Democracy in neural nets: voting schemes for classification. Neural Networks, 7, 691–707.
7. Baumann K. (2003). “Cross-validation as the objective function for variable-selection techniques”. Trends in Analytical Chemistry. pp. 395-406.
8. Bezdek J. C. (1981). “Pattern Recognition with Fuzzy Objective Function Algorithms”. Kluwer, Plenum Press, New York.
9. Bouckaert R. R. (2008). “Practical Bias Variance Decomposition”. In Proc 21st Australasian Joint Conference on Artificial Intelligence Auckland, New Zealand.
10. Cano J. R., Herrera F. and Lozano F. (2004). “Stratification for scaling up evolutionary prototype selection”. Pattern Recognition Letters 26, pp. 953–963
11. Castiello C., Castellano G., Fanelli A. M. (2008). “MINDFUL: A framework for Meta-INDuctive neuro-FUzzy Learning”. Information Sciences 178, pp. 3253–3274
12. Castro P. A. D. and Von Zuben F. J. (2006). “Bayesian Learning of Neural Networks by Means of Artificial Immune Systems”. 2006 International Joint Conference on Neural Networks
13. Castro P. D., Coelho G. P., Caetano M. F. and Von Zuben F. J. (2005). “Designing Ensembles of Fuzzy Classification Systems: An Immune-Inspired Approach”. ICARIS 2005, LNCS 3627, pp. 469–482.
14. Cordella, L. P., De Stefano C., Fontanella F. and Marcelli A. (2005). “Genetic Programming for Generating Prototypes in Classification Problems”.

15. Cover T. M. and Hart P. E. (1967). “Nearest neighbor pattern classification” IEEE Trans. Inform. Theory, vol. IT-13, pp. 21-027.

16. Creusere C. D. and Hewer G. (1994). “A Wavelet-Based Method of Nearest Neighbor Pattern Classification Using Scale Sequential Matching”. Naval Air Warfare Center Weapons Division, pp. 1123 – 1127.

17. Dam H. H. and Abbass H. A. (2008). “Neural-Based Learning Classifier Systems”. IEEE transactions on knowledge and data engineering, vol. 20, no 1.

18. Dasarathy B. V. and Sánchez J. S. (2000). “Tandem Fusion of Nearest Neighbor Editing and Condensing Algorithms – Data Dimensionality Effects”. pp. 692-695.

19. Deng D. and Zhang J. (2006). “Combining Multiple Precision-Boosted Classifiers for Indoor-Outdoor Scene Classification” Internal Report Nº 2006/09, Dpt. Information Science, University of Otago, Dunedin, New Zeland.

20. Dietterich T. G. (2000). “The divide-and-conquer manifesto”. Proceedings of the Eleventh International Conference on Algorithmic Learning Theory, pp. 13-26.

21. Duda R. O., Hart P. E. and Stork D. S. (2000). Pattern Classification, Wiley.

22. Fakhrahmad S. M., Zare A. and Jahromi M. Z. (2007). “Constructing Accurate Fuzzy Rule-Based Classification Systems Using Apriori Principles and Rule-Weighting”. LNCS 4881, pp. 547–556.

23. Fernandes S., Kamienski C., Kelner J., Mariz D. and Sadok D. (2008). “A stratified traffic sampling methodology for seeing the big picture”. Computer Networks.

24. García S., Cano J. R. and Herrera F. (2008). “A memetic algorithm for evolutionary prototype selection: A scaling up approach”. Pattern Recognition 41, pp. 2693 – 2709

25. Giacinto G., Roli F. and Bruzzone L. (2000). “Combination of neural and statistical algorithms for supervised classification of remote-sensing image”. Pattern Recognition Letters, vol. 21, no. 5, pp. 385-397.

26. Gonçalves L. B. and Vellasco M. B. R. (2006). Member, IEEE, Marco Aurélio Cavalcanti Pacheco, and Flavio Joaquim de Souza. “Inverted Hierarchical Neuro-Fuzzy BSP System: A Novel Neuro-Fuzzy Model for Pattern Classification and Rule Extraction in Databases”. IEEE Transactions on systems, man, and cybernetics – Part C: Applications and reviews, Vol 36, no.2.

27. Grim J., Kittler J., Pudil P. and Somol P. (2002). “Multiple Classifier Fusion in Probabilistic Neural Networks” Pattern Analysis and Applications, 5, 221-233.

28. Gu L., Wu H. (2008). “A kernel-based fuzzy greedy multiple hyperspheres covering algorithm for pattern classification”. School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, Jiangsu, China.

29. Guijarro, M. (2009). Combinación de clasificadores para identificación de texturas en imágenes naturales: nuevas estrategias locales y globales. Tesis Doctoral. Facultad de Informática. UCM.

30. Guijarro, M. and Pajares, G. (2009). On combining classifiers through a fuzzy multicriteria decision making approach: applied to natural textured images. Expert Systems with Applications, 36(3P2), 7262-7269.

31. Guijarro, M., Pajares, G. and Herrera, P. J. (2008). On combining classifiers by relaxation for natural images in images. Innovations in Hybrid Intelligent Systems (HAIS08). Advances in Soft Computing (Corchado, E., Corchado, J.M. and Abraham, A. Eds.), Lecture Notes in Artificial Intelligence, Springer-Verlag Berlin Heidelberg, September, 5271, 345-352, doi: 10.1007/978-3-540-87656-4_43.

32. Guijarro, M., Pajares, G., Abreu, R., Garmendia, L. and Santos, M. (2007a). Design of a Hybrid Classifier for Natural Textures in Images from the Bayesian and Fuzzy Paradigms. In Proc. IEEE International Symposium on Intelligent Signal Processing (WISP07). In Conference Proceedings Book (J. Ureña, J.J. García, Eds.), 431-436, Alcalá de Henares, Madrid, October 3-5, doi: 10.1109/WISP.2007.4447562

33. Guijarro, M., Abreu, R. and Pajares, G. (2007b). A New Unsupervised Hybrid Classifier for Natural Textures in Images. Innovations in Hybrid Intelligent Systems (HAIS07). Advances In Soft Computing. Lecture Notes in Artificial Intelligence (Eds. Emilio Corchado, Juan M. Corchado, Ajith Abraham), 44, 280-287, Springer-Verlag Berlin Heidelberg, November, doi:10.1007/978-3-540-74972-1_37.

34. Guijarro, M., Abreu, R. and Pajares, G. (2007c). On combining Learning Vector Quantization and the Bayesian classifiers for natural textured images. Proc. II Congreso Español de Informática. V Taller Nacional de Minería de Datos y Aprendizaje, (TAMIDA2007), 195-201, Zaragoza, Spain, September.

35. Handley S., Langley P. and Rauscher F. A. (1998). “Discovery and Data Mining”. New York: AAAI Press. Learning to Predict the Duration of an Automobile Trip

36. Hanmandlu M., Madasu V. K. and Vasikarla S. (2004). “A Fuzzy Approach to Texture Segmentation”. Proc. of the IEEE International Conference on Information Technology: Coding and Computing (ITCC’04), The Orleans, Las Vegas, Nevada, USA, pp. 636-642.

37. Ho, T.K., Hull, J.J. and Srihari, S.N. (1994). Decision combination in multiple classifier systems. IEEE Trans. Pattern Analysis and Machine Intell., 16, 66-75.

38. Hurtado C. (2007). “Evaluación de modelos de clasificación”. Departamento de ciencias de computación, Universidad de Chile.

39. Kerwin M. (2005). “A Fuzzy K-Nearest Neighbor Algorithm”. Review and Critical Analysis.

40. Kim D. W., Lee K. H. and Lee D. (2003). “Fuzzy Clúster validation index based on inter-clúster proximity” Pattern Recognition Letters, vol. 24, pp. 2561-2574.

41. Kim M. W. and Ryu J. W. (2005). “Optimized Fuzzy Classification Using Genetic Algorithm”. LNAI 3613, pp. 392 – 401,

42. Kittler J., Hatef M., Duin R. P. W. and Matas J. (1998). “On Combining Classifiers” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 226-239.

43. Kohavi R. (1995). “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. Computer Science Department, Stanford University.

44. Kumar S., Ghosh J. and Crawford M. M. (2002). “Hierarchical Fusion of Multiple Classifiers for Hyperspectral Data Analysis” Pattern Analysis and Applications, 5, pp. 210-220.

45. Kuncheva, L.I., Bezdek, J.C. and Duin, R.P. (2001). Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition, 34(2), 299-314.

46. Kuncheva L. I.. (2004). “Combining Pattern Classifiers: Methods and Algorithms, Wiley”.

47. Lachenbruch P. A. and Mickey M. R. (1968). “Estimation of error rates in discriminant analysis” Technometrics, vol. 10, pp. 1-10.

48. Lakshmanan, V., Fritz A., Smith T., Hondl K. and Stumpf G. J. (2007). “An automated technique to quality control radar reflectivity data”. Applied Meteorology, 46, 288–305

49. Lam, L. and Suen, C.Y. (1997). Application of majority voting to pattern recognition: an analysis of its behaviour and performance. IEEE Trans. Systems, Man and Cybernetics, 27, no. 5, 553-568.

50. Lin, X., Yacoub, S., Burns, J. and Simske, S. (2003). Performance analysis of pattern classifier combination by plurality voting. Pattern Recognition Letters, 24, no. 12, 1959-1969.

51. Liu D., Sun J., Wei G., Liu X. (2008). “RBF Neural Networks and Cross Validation-based Signal Reconstruction for Nonlinear Multi-functional sensor”. Department of Automatic Measurement and Control, Harbin Institute of Technology.

52. Merwe N. T. and Hoffman A. J. (2001). School for Electrical and Electronic Engineering.

53. Misra B. B., Dehuri S., Dash P. K. and Panda G. (2008). “Reduced Polynomial Neural Swarm Net for Classification Task in Data Mining”. 2008 IEEE Congress on Evolutionary Computation.

54. Pajares G. and Cruz J. M. (2002). “Clasificación de Texturas Naturales mediante K-Means”. Revista Electrónica de Visión por Computador http://revc.uab.es/revista/06/, no. 6, pp. 1-18.

55. Pajares G., Cruz J. M. and Moreno V. (2002). “Clasificación de texturas naturales mediante agrupamiento borroso”. Ingeniería Civil. Centro de Estudios y Experimentación de Obras Públicas (CEDEX).- Ministerio de Fomento, nº 127, pp. 83-89.

56. Pajares G., Moreno V. and Cruz J. M. (2001). “Clasificación de texturas mediante redes neuronales”. Ingeniería Civil. Centro de Estudios y Experimentación de Obras Públicas (CEDEX).- Ministerio de Fomento, nº 123, pp. 61-69.

57. Partridge D. and Griffith N. (2002). “Multiple Classifier Systems: Software Engineered, Automatically Modular Leading to a Taxonomic Overview” Pattern Analysis and Applications, 5, pp. 180-188.

58. Puig D. and García M. A. (2006). “Automatic texture feature selection for image pixel classification,” Pattern Recognition, vol. 39, nº 11, pp. 1996-2009.

59. Rosa J. L. A. and Ebecken N. F. F. (2003). “Data Mining for Data Classification Based on the KNN-Fuzzy Method Supported by Genetic Algorithm”. COPPE, Universidade Federal do Río de Janeiro, Brazil. VECPAR 2002, LNCS 2565, pp. 126-133.

60. Saastamoinen K. and Ketola J. (2006). “Medical Data Classification using Logical Similarity Based Measures”. Lappeenranta University of Technology, Lappeenranta, Finland. CIS.

61. Saastamoinen, K. and Ketola, J. (2006). “Medical Data Classification using Logical Similarity Based Measures”. Lappeenranta University of Technology, Lappeenranta, Finland. CIS 2006.

62. Stefanowski J. (2004). “An experimental evaluation of improving rule based classifiers with two approaches that change representations of learning”.

63. Stefanowski, J. (2004). “An experimental evaluation of improving rule based classifiers with two approaches that change representations of learning examples”. Engineering Applications of Artificial Intelligence 17, 439–445

64. Takagi N., Kikuchi H., and Mukaidono M. (2004). “Applications of Fuzzy Logic Functions to Knowledge Discovery in Databases”. Transactions on Rough Sets II, LNCS 3135, pp. 107–128.

65. Tsai C., Lee C. and Yang W. (2008). “A discretization algorithm based on Class-Attribute Contingency Coefficient”. School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, Jiangsu, China.

66. Tsipouras M. G., Exarchos T. P. and Fotiadis D. I. (2008). “A methodology for automated fuzzy model generation”. Fuzzy Sets and Systems 159, pp. 3201 – 3220

67. Tsipouras M. G., Exarchos T. P. and Fotiadis D. I. (2008). “A methodology for automated fuzzy model generation”. Fuzzy Sets and Systems 159, pp. 3201 – 3220.

68. Tubbs, J.D. and Alltop, W.O. (1991). Measures of confidence associated with combining classification rules. IEEE Trans. Systems, Man, and Cybernetics, 21, 690-692.

69. Asunción y Newman. (2009). UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/

70. Valdovinos R. M., Sánchez J. S. and Barandela R. (2005). “Dynamic and Static weigthing in classifier fusion”. In Pattern Recognition and Image Analysis, Lecture Notes in Computer Science (Marques J.S., Blanca N. P. and Pina P., Eds.), Springer-Verlag, Berlin, pp. 59-66.

71. Wahba G., Lin Y. and Zhang H. (2001). “Margin-Like Quantities and Generalized Approximate Cross-Validation for Support Vector Machines”.

72. Wang J., Keskovic P. and Cooper L. N. (2005). “An adaptive nearest neighbour algorithm for classification”. Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21, pp. 2069-3074.

73. Xu, L., Krzyzak, A. and Suen, C.Y. (1992). Methods of combining multiple classifiers and their application to handwriting recognition. IEEE Trans. System, Man and Cybernetics, 22, 418-435.

74. Zimmermann H. J. (1991). “Fuzzy Set Theory and its Applications”. Kluwer Academic Publishers, Norwell.

75. Zseby T. (2003). “Stratification strategies for sampling-based non-intrusive measurements of one-way delay”. Passive and Active Measurement Workshop Proceedings.