Instituto de Ciencias del Seguro

FACTORES DE RIESGO Y CÁLCULO DE PRIMAS MEDIANTE TÉCNICAS DE APRENDIZAJE

Carlos Bousoño Calzón
Antonio Heras Martínez
Piedad Tolmos Rodríguez-Piñero

FUNDACIÓN MAPFRE

Carretera de Pozuelo 52
28220 Majadahonda. Madrid

www.fundacionmapfre.com/cienciasdelseguro
publicaciones.ics@mapfre.com

ISBN: 978-84-9844-099-7
Depósito Legal: SE-2569-2008

PRESENTACIÓN

Desde 1992 FUNDACIÓN MAPFRE realiza anualmente una convocatoria de becas destinadas a promover estudios monográficos en materia de Riesgo y Seguro, incluyendo áreas temáticas relacionadas específicamente con el seguro iberoamericano.

Su objetivo es facilitar apoyo económico para la realización de trabajos de investigación en las áreas antes mencionadas y están dirigidas a titulados universitarios y profesionales del mundo del seguro, de cualquier nacionalidad, que deseen desarrollar programas de investigación.

Para la realización de este trabajo, FUNDACIÓN MAPFRE concedió a sus autores una Beca de Investigación Riesgo y Seguro.

Carlos Bousoño Calzón es Ingeniero en Telecomunicación y Doctor Ingeniero en Telecomunicación por la Universidad Politécnica de Madrid. Ejerce como profesor de la Universidad Carlos III de Madrid, donde desarrolla docencia e investigación en el ámbito de las tecnologías de la información, con especial énfasis en aplicaciones multidisciplinares.

Antonio Heras Martínez es Licenciado en Matemáticas y Doctor en Ciencias Económicas y Empresariales y Catedrático del Departamento de Economía Financiera y Contabilidad I (Economía Actuarial y Financiera) de la Universidad Complutense de Madrid,, con docencia e investigación relacionadas con la Matemática Actuarial en general y con temas de reaseguro y tarificación en particular.

Piedad Tolmos Rodríguez-Piñero es Licenciada en Matemáticas y Doctora en Ciencias Económicas y Empresariales. Es profesora titular del Departamento de Economía Financiera y Contabilidad II (Matemáticas) de la Universidad Rey Juan Carlos de Madrid, y su investigación está relacionada con la aplicación de diversas técnicas de Inteligencia Artificial al campo de la Economía en general, y de la ciencia actuarial en particular.

ÍNDICE

Página

PRÓLOGO

Uno de los problemas más importantes, y al mismo tiempo más interesantes, de las Ciencias Actuariales, es el de la Tarificación o cálculo de primas. En este proceso, una de las fases principales es la de la selección de los factores de riesgo, características de los asegurados que están correlacionadas con la siniestralidad y que conjuntamente explican un gran porcentaje de la varianza de la misma.

Tradicionalmente, los factores de riesgo se seleccionan utilizando técnicas estadísticas, habitualmente de tipo multivariante que permiten organizar procesos de selección teniendo en cuenta simultáneamente el conjunto de factores.

En este caso hemos empleado otro tipo de herramientas, que si bien hasta ahora no se habían aplicado en este tipo de problemas actuariales, sí se había hecho en otros problemas similares de otros ámbitos, con excelentes resultados. Se trata de técnicas de *Aprendizaje Máquina* que han demostrado sus aptitudes como clasificadores, y se han aplicado con éxito en multitud de problemas complejos, como el del reconocimiento de caracteres escritos, la "limpieza" de imágenes, minería de datos, diagnósticos médicos, etc.

Sin embargo, en el campo de los seguros, su uso se ha centrado casi exclusivamente en el estudio de la predicción de insolvencia en empresas de seguros y temas relacionados.

El presente trabajo es el resultado de nuestros esfuerzos en cuanto a abrir una nueva vía al uso de estas importantes técnicas, con el objeto de que los problemas actuariales de selección-clasificación como los aquí tratados se beneficien de su indudable potencial. Esperamos que los excelentes resultados que hemos obtenido animen a otros investigadores a hacerlo así.

Esta obra tiene su origen en la Tesis doctoral de la coautora Piedad Tolmos Rodríguez-Piñero "*Selección de factores de riesgo y predicción de siniestros en el seguro del automóvil mediante métodos de aprendizaje máquina*", dirigida por los coautores Antonio Heras Martínez y Carlos Bousoño Calzón, que fue defendida en Marzo de 2007 en la Universidad Rey Juan Carlos de Madrid, obteniendo la máxima calificación.

El desarrollo de la tesis se benefició de las facilidades que la Fundación MAPFRE nos dio gracias a la Beca de Riesgo y Seguro que nos otorgó en su convocatoria 2006. Deseamos por ello agradecer sinceramente a los miembros de la Fundación la concesión de dicha Beca, así como al miembro de MAPFRE automóviles que fue nuestro tutor durante la duración de la misma, Juan Antonio Rodrigo, y a los asimismo miembros de MAPFRE automóviles Juan Armijo y Alfonso Ortiz, su inestimable colaboración y ayuda.

INTRODUCCIÓN

El presente trabajo está dedicado a la aplicación de ciertas técnicas de Aprendizaje Máquina a la resolución de un problema del ámbito actuarial: dada una base de datos con los asegurados de una Compañía del seguro del automóvil en un periodo de un año, queremos clasificar a los asegurados atendiendo a si han tenido o no siniestros durante ese año, y realizar además una selección de los factores de riesgo más influyentes en la siniestralidad de entre los que describen dichos clientes.

No hemos encontrado apenas referencias del uso de las estrategias de *aprendizaje* (fuera de las clásicas como el Análisis Discriminante) en el campo actuarial. Técnicas tales como las Redes Neuronales Artificiales, o las Máquinas de Vectores Soporte han demostrado ser unos clasificadores excelentes en términos de la llamada *tasa de clasificación*, y se han aplicado con éxito en multitud de problemas complejos, como el del reconocimiento de caracteres escritos, la "limpieza" de imágenes, minería de datos, diagnósticos médicos, etc. Sin embargo, en el campo de los seguros, su uso se ha centrado casi exclusivamente en el estudio de la predicción de insolvencia en empresas de seguros y temas relacionados [Shapiro 2001], [Shapiro 2002]. Es por ello, por la aplicación novedosa que suponen, y la confianza en obtener buenos resultados, que nos ha parecido interesante el emplearlas en la resolución de nuestro problema de clasificación.

Las cuestiones que nos planteamos resolver son parte de lo que se conoce como proceso de *Tarificación*, para la obtención de la prima a pagar por un asegurado.

Es un hecho que el seguro, sea del tipo que sea, se basa en que cada asegurado del colectivo acuerda con la compañía pagar una cierta cantidad de dinero, conocida como *prima*, de manera que la suma de todas las primas recogidas se emplean para pagar los posibles siniestros ocasionados por el conjunto de los asegurados durante el periodo de tiempo pactado. De esta forma, el asegurado evita el desembolso de lo que constituiría la cuantía de sus siniestros, evidentemente desconocida, sustituyéndola por el pago de esa cantidad cierta de dinero, la prima.

Lo deseable sería que el colectivo de asegurados que comparten sus gastos pagando una misma prima, sea lo más homogéneo posible en cuanto a sus riesgos se refiere, pues si no fuera así, los asegurados que generan más gastos a la compañía deberían pagar primas mayores. En caso contrario podría

darse una *selección desfavorable* de los asegurados que ponen en peligro la solvencia de la empresa. Por ello resulta esencial que la compañía aseguradora sea capaz de clasificar a sus clientes del modo más homogéneo posible atendiendo al riesgo, de manera que los asegurados pertenecientes a un mismo grupo paguen idéntica prima.

Para lograr ese objetivo, se recurre a técnicas estadísticas que permiten detectar los denominados *factores de riesgo*, características de los asegurados que están correlacionadas con la siniestralidad y que conjuntamente explican un gran porcentaje de la varianza de la misma. Esa es la vía para que los asegurados con similares factores de riesgo pertenezcan finalmente al mismo grupo.

El número de factores de riesgo necesario para una correcta clasificación difiere en el seguro de vida y en el de automóvil. Así, mientras que en el primero es suficiente generalmente con considerar la edad y el sexo de los asegurados, en el segundo no es así. Tradicionalmente se utilizan factores como la edad, el sexo, la provincia de residencia, profesión del conductor, antigüedad de su carné de conducir y del vehículo, potencia del motor, etc. Y se recogen muchos más, como el color del vehículo, o el estado civil. Pese a todo, estos factores de riesgo están escasamente correlacionados con su siniestralidad, y explican sólo una pequeña parte de su varianza. Lo ideal sería utilizar factores como la rapidez de reflejos del conductor, sus hábitos, su carácter al volante, si consume alcohol, drogas, etc. Pero esto no resulta posible, ya que esos factores no son observables. La clasificación atendiendo sólo a los factores observables es lo que se conoce como *clasificación o tarificación a priori*. Una parte esencial del proceso de tarificación, será la de la selección de los factores de riesgo más influyentes a la hora de presentar siniestro, de entre todos los dados. Esa será precisamente la primera parte de nuestro trabajo.

El sistema de tarificación a priori tiene la "pega" de que las clases resultantes son altamente heterogéneas. Es por ello que las compañías tienden a considerar una variable que haga el papel de los factores inobservables: el número de siniestros de cada póliza durante los últimos años. Cuando se tiene en cuenta el dato de la siniestralidad pasada del asegurado, con el objeto de que la clasificación sea lo más homogénea posible, y el cálculo de la prima del año próximo sea más preciso, se dice que el método de *tarificación* es *a posteriori*. Para el estudio de este problema existe una rama de la Ciencia Actuarial conocida como *Teoría de la Credibilidad.* Cuando el número de pólizas es lo suficientemente grande, el desarrollo de un sistema de clasificación es el primer paso para lograr una prima justa. Los sistemas de tarificación basada en la experiencia en general, y los métodos de la teoría de la credibilidad en particular, constituyen un segundo paso en la determinación de la prima adecuada. Nuestro trabajo se centra exclusivamente en ese proceso previo de tarificación *a priori*[1].

[1] Conviene aclarar que, a pesar de que no sea propio de la tarificación a priori, en nuestro

En la literatura, se pueden encontrar soluciones a esta tarea empleando métodos estadísticos Las técnicas del análisis estadístico multivariante son las que permiten organizar procesos de selección teniendo en cuenta simultáneamente el conjunto de factores. Recordemos que el primer objetivo (antes de clasificar) es obtener un conjunto equilibrado de factores de riesgo, aquel que mejor explique la estructura del riesgo. Si realizáramos la selección considerando separadamente las variables que una a una están más asociadas con el riesgo, es posible que el conjunto de variables seleccionadas resultante contuviera información redundante, o lo que es más importante, que no tuviéramos incorporadas variables que de manera conjunta con otras resulten significativas. Es por ello que precisamos hacer el estudio teniendo en cuenta a la vez todos los factores potenciales de riesgo e "idealmente" todas sus interacciones.

Un criterio de clasificación de este tipo de métodos se puede definir según las fases del proceso de tarificación *a priori* que nos permiten cubrir. En este caso, dividimos las herramientas empleadas en *predictivas*, que en principio cubrirán todo el proceso de tarificación, y *no predictivas*, que abarcarán sólo alguna fase. Entre las primeras están los Modelos de Regresión, y las Técnicas de Segmentación, y en las segundas encontramos el Análisis Cluster y el Análisis Discriminante.

De entre todas las técnicas anteriores, hemos seleccionado el *Análisis Discriminante* para clasificar a nuestros asegurados y comparar así lo obtenido con los que alcancemos vía las técnicas de Aprendizaje Máquina. De este modo, podremos comprobar la fiabilidad de los resultados que hayamos logrado con las nuevas herramientas.

La elección de este método y no otro, se justificará por su relación directa con las técnicas que hemos denominado de aprendizaje máquina, ya que el Análisis Discriminante, tal y como lo concibió Fisher, se puede considerar un algoritmo de aprendizaje de entre los "estadísticos clásicos".

Por otra parte, debido a la naturaleza de las variables que vamos a tratar, deberemos llevar a cabo un *Análisis Factorial* antes de proceder a discriminar.

Efectivamente, el análisis discriminante es una herramienta que se aplica solamente a variables cuantitativas viniendo caracterizados los grupos a través de una variable categórica, que en nuestro caso será la variable *siniestros*. Como veremos, nuestras variables son en su mayor parte de tipo cualitativo. Se hará necesario por tanto, para aplicar un análisis discriminante, transformar previamente el conjunto de variables en variables cuantitativas/continuas.

Para ello se procederá de la siguiente forma:

---

trabajo deberemos utilizar el dato de la siniestralidad (la información de si ha habido o no siniestro) para entrenar el clasificador.

1. Se realizará un Análisis Factorial Múltiple (AFM) sobre el conjunto completo de las variables.

2. Selección de aquellos factores resultantes del AFM que expliquen el 100% de la variabilidad total.

3. Realización del Análisis Discriminante (AD) con los factores anteriormente seleccionados.

El AFM es una técnica de análisis factorial que trata tablas en las cuales un conjunto de individuos viene descrito por varios grupos de variables. En el seno de un mismo grupo, las variables deben ser del mismo tipo (continuo o nominal) pero de un grupo a otro, las variables pueden ser de diferentes tipos. En nuestro experimento se lleva a cabo un AFM sobre el conjunto de individuos descritos a través de dos tablas, la primera formada por las variables continuas y la segunda formada por las variables nominales.

El método es en realidad un análisis factorial del conjunto de grupos (llamado análisis global). Para las variables continuas, el AFM se comporta como un Análisis en Componentes Principales; para las variables nominales como un Análisis de Correspondencias Múltiples. Es la introducción de pesos en las variables, que equilibran las inercias axiales máximas de los grupos, lo que hace posible la presencia simultánea de variables continuas y nominales. El objetivo final es obtener los principales factores de variabilidad de los individuos, estando éstos descritos de forma equilibrada por varios grupos de variables.

Obsérvese que, en este caso, no nos valdremos de la técnica estadística para comparar resultados con las de aprendizaje, como hacemos con el AD, sino para dar el paso necesario antes de proceder a efectuar el análisis discriminante.

Hasta aquí, lo que hemos realizado sería una aproximación clásica al problema. Pero decíamos que donde realmente pretendemos introducir una experiencia nueva es en la aplicación de herramientas tomadas de la Teoría del Aprendizaje a este problema de Tarificación en seguros del automóvil. El enfoque estadístico “clásico” para abordar un problema de clasificación de este estilo, en el que se cuenta con una gran cantidad de datos, consiste en asumir que tales datos están generados por una *distribución de probabilidad* subyacente que nos es desconocida y a partir de la cual diseñaremos el *clasificador*. Sin embargo, existen otras aproximaciones al problema, como la que nos proponen Vapnik y otros autores de la Teoría del Aprendizaje. La idea básica es diseñar el clasificador directamente desde los datos mediante determinados algoritmos, que en su caso se basan en esta Teoría del Aprendizaje.

Este modo de plantear el problema nos conducirá a la necesidad de analizar la información que comprenden los grandes conjuntos de datos. La habilidad de

extraer el conocimiento que se encuentra escondido entre esos datos, y utilizarlo convenientemente, está teniendo una importancia creciente en el mundo contemporáneo. Las aproximaciones más recientes para desarrollar modelos a partir de los datos se han inspirado en las capacidades de *aprendizaje* de los sistemas biológicos, y, en particular, en las de los humanos.

De hecho, los sistemas biológicos aprenden a hacer frente a la desconocida naturaleza estadística del entorno conducidos por los datos. Los humanos, como los animales, tienen la capacidad superior de *reconocer patrones*, como las de identificar caras, voces u olores. El campo del reconocimiento de patrones tiene como objetivo el construir sistemas artificiales que imiten las habilidades de reconocimiento de los humanos, y avanzan hacia él basándose en principios de ingeniería y estadística.

En un sentido amplio, cualquier método que incorpore información de las muestras de entrenamiento en el diseño de un clasificador emplea *aprendizaje*. La creación de clasificadores implica el planteamiento de una forma general de modelo, o de clasificador, y el uso de los datos de entrenamiento para *aprender* o estimar los parámetros desconocidos del modelo. Cuando hablemos aquí de *aprendizaje* lo haremos como una forma de algoritmo para reducir el error sobre el conjunto de entrenamiento. Concretamente, un *algoritmo de aprendizaje* es aquel que toma los datos de entrenamiento como entrada (*input*) y selecciona la hipótesis (la función candidata de entre todas a ser la que relaciona las salidas con las entradas, esto es, la *función de decisión*) de entre todas las posibles.

En estadística, la tarea del aprendizaje predictivo a partir de las muestras la desempeña la que conocemos como *estimación estadística*. Tiene por objeto estimar las propiedades de cierta distribución estadística (desconocida) desde las muestras o los datos de entrenamiento. La información contenida en los datos de entrenamiento (experiencia pasada) se puede utilizar asimismo para responder preguntas acerca de las futuras muestras. Por lo tanto, distinguimos dos etapas en un sistema de aprendizaje [Cherkassky 1998]: (1) aprendizaje/estimación (a partir de las muestras de entrenamiento) y (2) operación/predicción, cuando las predicciones se hacen para las muestras futuras o de Test. Esta descripción asume que tanto los datos de entrenamiento como los de test parten de la *misma* distribución estadística subyacente. En otras palabras, esa distribución (desconocida) es fija.

La aproximación *estadística clásica*, tal y como la propone Fisher (1952) con su *Análisis Discriminante*, divide el problema de aprendizaje en dos partes: especificación y estimación. La especificación consiste en determinar la forma paramétrica de las distribuciones subyacentes desconocidas, mientras que la estimación es el proceso de determinar los parámetros que caracterizan las distribuciones específicas. La teoría clásica focaliza el problema de la estimación, y deja de lado el de la especificación. Las aproximaciones clásicas dependen de asunciones más estrictas que las que se plantean en la formulación general del aprendizaje, ya que asumen que las funciones se

especifican con un número fijo de parámetros. Esto plantea serias limitaciones en muchas de las aplicaciones. Se tiende entonces a introducir un gran margen de error modelizado, la discrepancia entre modelo paramétrico asumido y la verdad desconocida. Como los métodos no paramétricos clásicos, operan sólo en el caso asintótico, que requieren muestras de un tamaño enorme, y nunca tendremos muestras de tamaño suficiente para satisfacer esas condiciones asintóticas.

Las limitaciones de las aproximaciones clásicas motivan la aparición de los métodos adaptativos o flexibles. El enfoque que hemos denominado de aprendizaje máquina recoge métodos de la Inteligencia Artificial. La que conocemos como *Máquinas de Vectores Soporte* (SVM) es un procedimiento de aprendizaje universal basado en la teoría del aprendizaje. El término "universal" significa que las SVM se pueden utilizar para aprender una variedad de representaciones, como las redes neuronales (con la *función de activación* sigmoide habitual), las *funciones de base radial*, los *estimadores polinomiales* y los *splines*. Considerándolas de un modo muy general, las SVM proporcionan una nueva manera de parametrización de funciones, por lo que pueden emplearse tanto en aprendizaje como en predicción.

Volviendo al problema planteado, nuestro objetivo final es, como hemos dicho, el de *clasificar* los asegurados en dos categorías atendiendo al riesgo de tener un accidente. Para ello, realizamos un estudio en el que detectaremos qué asegurados van a tener un siniestro y quiénes no, atendiendo a unos ciertos factores de riesgo. Esos factores son el resultado de una *selección* de entre todos los dados, en la que se conservan los que resulten relevantes para el problema. De este modo, dado un nuevo asegurado, seremos capaces de predecir, con una probabilidad, si tendrá o no siniestro en un año. Además, podremos indicar cuáles de los factores de riesgo que se han recogido son realmente esenciales para determinar esa predicción, aliviando de este modo a la compañía aseguradora. Efectivamente, veremos que la probabilidad de asignar a un asegurado a la clase correcta es prácticamente igual utilizando todos los factores de riesgo que sólo los seleccionados. Estamos aumentando, por tanto, la capacidad de generalización de nuestro clasificador.

Es así como la elección de los factores de riesgo, además del interés que proporciona por sí misma en cuanto al aporte de información para mejorar el proceso de recogida de datos, redunda en un mejor funcionamiento del sistema. Ese proceso forma parte de uno de los principales campos de la Teoría del Aprendizaje, el llamado *Problema de Selección de Características*, esto es, la selección de los factores o rasgos que permitan desechar aquellos elementos que se revelen como irrelevantes para el estudio que se desea realizar.

En los problemas de clasificación como el que nos ocupa, el objetivo de la selección de características es escoger un subconjunto de variables de entrada (factores de riesgo) que sean los que preserven o mejoren la capacidad del clasificador [Weston et al. 2000].

Entre los distintos modos de tratar este problema, el más frecuente es el siguiente: dado un conjunto de datos del espacio $\mathbb{R}^n$, extraer un subconjunto de *m* variables (*m* < *n*) que posean el error de clasificación menor [Weston et al. 2000].

Evidentemente, la aplicación de estos métodos debe precederse de una descripción teórica del funcionamiento de los mismos. Para ello, debe plantearse el problema desde un punto de vista matemático. Vamos a predecir la siniestralidad de un asegurado, o lo que es lo mismo, vamos a separar a todos los clientes en dos clases: la de los que tendrán siniestros, y la de los que no. Por ello, se considera como un problema de clasificación con múltiples atributos "simple", esto es, con sólo dos clases. Los conjuntos de vectores de $\mathbb{R}^n$ $\{\mathbf{x}_i\}, i \in \{1,...l\}$ representarán a los asegurados, descritos por un conjunto de factores de riesgo (cada componente de $\mathbf{x}_i$, $\mathbf{x}_{ij}$, es un factor), y las *etiquetas* $y_i \in \{-1,1\}$ indicarían la clase, -1 si no tendrán siniestros, y 1 en caso contrario.

El objetivo es que se clasifiquen correctamente los ejemplos nuevos que se presenten $(\mathbf{x}, y)$ generados por la misma distribución de probabilidad *"subyacente"* $P(\mathbf{x}, y)$ que los datos utilizados para el entrenamiento del clasificador. Esta labor se puede realizar tanto seleccionando los factores de riesgo como no, y nosotros lo haremos de las dos formas, para comparar los resultados.

Para llevar a cabo las tareas que nos proponemos, nos vamos a valer de tres técnicas de Aprendizaje Máquina: Máquinas de Vectores Soporte para la clasificación, y Algoritmos Genéticos y Árboles de Clasificación para la selección de factores. De todas ellas haremos una descripción bastante detallada, que constituye una parte importante del trabajo.

La *Máquina (red) de Vectores Soporte* es una nueva técnica de clasificación, que ha demostrado sobradamente su capacidad de resolución frente a problemas de elevado grado de complejidad. Diseñadas en principio para tratar problemas de clasificación binarios (en dos grupos), se trata de una máquina de aprendizaje que implementa la siguiente idea: cuando no sea posible separar los datos en el espacio de entrada con un hiperplano lineal, trasladar, mediante una aplicación no lineal, los vectores de entrada a un nuevo espacio de dimensión muy alta. En este nuevo espacio se construirá una superficie de decisión lineal. Las especiales propiedades que poseerá esta superficie garantizarán que la capacidad de generalización de la máquina de aprendizaje sea alta. Aunque esta idea se empleó en los primeros experimentos para datos que podían separase sin errores, se puede extender para el caso no separable con notable éxito. La parte conceptual del problema la resolvió Vapnik en 1965 para el caso de *hiperplanos óptimos* para clases separables. En este contexto, Vapnik definió un hiperplano óptimo como una función de decisión lineal con el margen de separación máximo entre los vectores de las dos clases. Se

observó entonces que para construir tal hiperplano, uno sólo debía tener en cuenta una cantidad pequeña de los datos de entrenamiento, los llamados *vectores soporte*, quienes determinaban ese *margen*.

Los *Algoritmos Genéticos* (Holland, 1975) son un logro más de la Inteligencia Artificial en su intento de replicar comportamientos biológicos mediante la computación. Se trata de algoritmos de búsqueda basados en la mecánica de la selección natural y de la genética. Utilizan la información histórica para encontrar nuevos puntos de búsqueda de una solución óptima del problema planteado, con esperanzas de mejorar los resultados.

Inspirados en el proceso de la evolución biológica, los métodos evolutivos emplean la búsqueda estocástica en el diseño de un clasificador óptimo. Una ventaja "computacional" de estos métodos es que admiten una implementación natural mediante ordenadores paralelos. Básicamente, proceden del siguiente modo: primero, se crea una *población* (compuesta por varios clasificadores en nuestro caso), cada *individuo* de la misma levemente diferente de los otros; a continuación, se *puntúa* cada individuo en base a su comportamiento en la tarea que han de ejecutar, observando por ejemplo su precisión ante una serie de ejemplos ya etiquetados. En aras de preservar la analogía biológica, al dato resultante se le suene nombrar *capacidad*. A continuación se ordenan los elementos de la población atendiendo a su capacidad, seleccionando los mejores, que representarán una porción de la población total. Es lo que en la teoría de la evolución biológica se conocería como *supervivencia del más capacitado*. En ese momento, se alteran estocásticamente los individuos para crear la siguiente generación (los hijos o *descendencia*). Algunos de ellos tendrán una puntuación más alta que la de sus progenitores, otros la tendrán peor. Por ello, el proceso completo (que comprende los operadores de *reproducción*, *cruce* y *mutación*) se repite para esta la siguiente generación, y para las siguientes, hasta que se logre que un individuo posea una capacidad que supere un valor definido para el criterio que se desea alcanzar.

El Aprendizaje por *Árboles de Decisión* es una de las técnicas de inferencia inductiva más utilizadas y prácticas de las que abarca la Teoría del Aprendizaje. Se han aplicado con mucho éxito a un amplio abanico de tareas, desde el diagnóstico médico, al estudio de riesgo en la concesión de créditos bancarios. Una de las principales novedades que presentan los árboles es que permiten el tratamiento de datos no numéricos de modo "directo".

Hablando de forma general, podemos considerar el aprendizaje mediante árboles como un procedimiento para aproximar funciones objetivo valoradas de forma discreta, en los que la función que se va a aprender se representa por medio de un árbol de decisión. Los árboles de aprendizaje permiten también una segunda representación como conjuntos de reglas *if-then* ("si-entonces") para mejorar la legibilidad humana.

Un árbol de decisión o de clasificación, o, sencillamente, árbol, es básicamente un diagrama que representa un sistema de clasificación o un modelo predictivo

como el arriba descrito. El árbol se estructura como una secuencia de preguntas sencillas, cuyas respuestas trazan un camino que lleva hacia abajo en él árbol. El punto final alcanzado (*hojas*) determina la clasificación o predicción hecha por el modelo, que puede constituir una respuesta tanto cualitativa como cuantitativa.

Veamos para finalizar cuál será la estructura del trabajo. Tras esta Introducción, pasaremos a dar una brevísima Introducción a la Tarificación en el Capítulo 2. Allí, tras dar unos conceptos básicos relativos a los Seguros, describiremos los procesos de Tarificación, deteniéndonos levemente en el de Tarificación a priori. Estudiaremos los factores de riesgo, y tras exponer el tipo de herramientas estadísticas que se emplean en la Tarificación a priori, finalizaremos detallando el tipo de datos que manejaremos en nuestros experimentos, ya con todo el vocabulario actuarial en nuestro poder

Los *Capítulos 3 y 4* están dedicados a las técnicas utilizadas en nuestro estudio, y a la aplicación de las mismas. Para ello, enmarcaremos el trabajo en la Teoría del Aprendizaje, y pasaremos a describir, en apartados separados, cada uno de los métodos empleados. Sí nos gustaría señalar que la descripción del Análisis Discriminante será menos detallada que la de las otras técnicas, ya que su objeto es realmente el de actuar de "juez" sobre la calidad de los resultados. Terminaremos cada uno de esos apartados con la ejecución del experimento por medio de la correspondiente herramienta. Los Capítulos finalizarán con la comparación entre los resultados obtenidos con las diferentes técnicas.

Esperaremos a concluir el Capítulo 4 para dar unas pinceladas sobre las líneas futuras de investigación que puede dejar abiertas el presente trabajo.

Finalmente, añadiremos un *Apéndice* para describir la técnica del Análisis Factorial, utilizado, como hemos apuntado, en el desarrollo de la aplicación del Análisis Discriminante al problema.

CAPÍTULO 1
INTRODUCCIÓN A LA TARIFICACIÓN

Como comentábamos en el Capítulo anterior, el objeto del presente trabajo es aplicar diversas técnicas de Aprendizaje Máquina a un problema propio del campo actuarial, la clasificación de asegurados atendiendo a si han tenido o no siniestros en un periodo determinado de tiempo, en nuestro caso de un año. Para poder hacer comprensible nuestro estudio, necesitaremos dar al menos unas pinceladas sobre diferentes conceptos del área de la Ciencia Actuarial, de manera que así se alcance una aproximación mejor al problema, su significado e implicaciones, en el momento de realizar los experimentos. Nos gustaría volver a señalar, sin embargo, que el objetivo primordial del trabajo es la introducción de estas innovadoras técnicas a la resolución de problemas relativos a los Seguros, y no se trata de un trabajo propiamente actuarial. De este modo, tras una breve Introducción, el primer punto del Capítulo, estará dedicado a dar una breve descripción de los Sistemas de Tarificación. Nos detendremos entonces a estudiar con más detenimiento el de la Tarificación a priori, así como los Factores de Riesgo, básicos para el desarrollo de la misma. Por último, describiremos brevemente las Técnicas Estadísticas que se emplean habitualmente en el proceso de Tarificación. Finalizaremos detallando los datos que utilizaremos en nuestro experimento.

1. INTRODUCCIÓN

El tipo de estudio que queremos abordar está englobado dentro de lo que se conoce como proceso de *tarificación a priori*, que matemáticamente hablando, es considerable como un problema general de clasificación multiatributo con múltiples clases (aunque en nuestro caso, dedicaremos la atención sólo al problema binario, una clase para los que han tenido un siniestro, y otra para los que no), donde $\{\mathbf{x}_i\}, i \in \{1,...I\}$ representan a los asegurados, descritos por un conjunto de *factores de riesgo*: cada componente de $\mathbf{x}_i$, $\mathbf{x}_{ij}$, es un factor y la salida viene dada por $y_{ik} = \begin{cases} +1 & \text{si } \mathbf{x}_i \text{ está en la clase } k \\ -1 & \text{en otro caso} \end{cases}$.

El conjunto de factores de riesgo empleados para representar a un asegurado puede llegar a ser muy amplio, si bien no todos esos factores se utilizan finalmente en la práctica (porque realmente resultan irrelevantes para describir

su comportamiento) para determinar la clase a la que pertenecerá ese elemento (función, en última instancia, del número de siniestros que vaya a tener, y de la cuantía de los mismos). Además, de cara a la máquina, el uso de demasiadas variables puede redundar en un pobre funcionamiento del sistema.

Por ello, una parte complementaria a la resolución de nuestro problema va a ser el *seleccionar* aquellos factores que consideremos esenciales para el proceso de clasificación, eliminando los que no resulten determinantes, o lo sean menos, para describir a un asegurado. Este problema, como trataremos a continuación, es una parte esencial del proceso de tarificación, y resulta determinante a la hora de realizar los cálculos de las *primas*. La selección que llevaremos a cabo, sin embargo, se realizará sobre los factores de riesgo ya escogidos por la aseguradora, de manera que nuestro experimento sería como un segundo filtro a partir del ya realizado.

2. NOCIONES BÁSICAS DE TARIFICACIÓN

2.1 Sistemas de Tarificación

Un *sistema de tarificación* es el conjunto de principios técnicos en que se basa la elaboración de una tarifa. El objeto que persigue todo sistema de tarificación es la obtención de primas equitativas para cada riesgo, teniendo siempre en cuenta que la solvencia del ente asegurador debe estar garantizada.

Respetando el *principio de equidad*, del que luego hablaremos con detalle, en la elaboración de las tarifas deberemos considerar los factores de riesgo más significativos, es decir, los que más *explican* el comportamiento de la siniestralidad como variable endógena del modelo. Hay que añadir que estos factores deben contemplarse en los niveles adecuados para evitar una excesiva dispersión de la siniestralidad en las clases de riesgo que finalmente figuren en la tarifa.

En cuanto a la solvencia, ha de garantizarse que las primas sean *suficientes,* esto es, que permitan hacer rentable, en condiciones de estabilidad a largo plazo, a la empresa aseguradora. El siguiente cuadro aclara cuáles son los sistemas de tarificación [Nieto y Vegas 1993]:

<table>
<tr><th rowspan="2">TARIFICACIÓN A PRIORI</th><th colspan="3">TARIFICACIÓN A POSTERIORI (EXPERIENCE RATING)</th></tr>
<tr><td rowspan="2">Depoid
o class rating</td><td colspan="3">Principio de eficacia en la tarificación</td><td>Principios de eficacia y de estabilidad</td></tr>
<tr><td>Bonus-Malus</td><td>Merit-rating</td><td>Retrospective-rating</td><td>Distribución de dividendos</td></tr>
</table>

Procedamos a explicar detenidamente el esquema:

Como se puede apreciar, se distinguen dos sistemas de tarificación, *a priori*, o class-rating, y *a posteriori* o experience rating. En la *Tarificación a priori*, la prima se calcula sin tener necesariamente información sobre la siniestralidad previa de la póliza. En su lugar, la tarificación se basa en otras características observables relacionadas con la siniestralidad. Como acabamos de comentar, a esas características se las denomina *factores de riesgo.*

La rama de la Ciencia Actuarial que estudia la elección de los factores de riesgo que han de incorporarse a una tarifa es la Estadística Actuarial. El criterio estadístico que se emplea para realizar la selección, a grandes rasgos, es que la media de daños resultante en cada clase sea distinta, y que la dispersión dentro de cada clase sea mínima.

En la *Tarificación a posteriori,* por el contrario, la tarifa se va modificando a lo largo del tiempo de acuerdo con la siniestralidad observada de la póliza, de forma que las pólizas con excesiva siniestralidad sufran penalizaciones, y al contrario, las pólizas con poca siniestralidad consigan bonificaciones en su tarifa. Es bien sabido que las técnicas de tarificación a priori no pueden eliminar totalmente la heterogeneidad de las distintas clases de asegurados, debido a que algunos de los factores de riesgo más importantes son inobservables. Este hecho justifica el uso de los sistemas de tarificación a posteriori por parte de las compañías de seguros, para conseguir ajustar, en la medida de lo posible, las primas a la experiencia de la siniestralidad de los asegurados. Sin embargo, los sistemas de tarificación a posteriori no sustituyen, sino que complementan a los sistemas de tarificación a priori. En efecto, estos últimos siguen siendo necesarios para el cálculo de la tarifa inicial sobre la que girarán las bonificaciones o penalizaciones anteriormente mencionadas.

2.2 Sistema a priori o class-rating

Veremos en este subapartado más extensamente el Sistema de Tarificación *a priori* que introducimos brevemente en el punto anterior, puesto que es al que dedicamos la atención en el presente trabajo.

1. El proceso de Tarificación

El proceso, tal y como indican [de Wit 1986], [van Eeghen, Greup y Nijssen 1983], sigue los siguientes pasos [Boj et al 2004]:

- **Determinación de la estructura de tarifa**

Para ello, deberemos seleccionar las variables tarificadoras (determinar los factores de riesgo), y subdividirlas para definir las clases de tarifa, obteniéndose así los correspondientes grupos de riesgo lo más homogéneos posible.

- **Cálculo de un nivel adecuado de prima para cada grupo de tarifa**

Para ello deberemos calcular la siniestralidad esperada asociada con cada clase de tarifa, a la que se suele denominar *la prima pura*. Posteriormente esta prima pura se incrementa con diferentes tipos de recargos para gastos.

- **Implementación de la tarifa en un mercado competitivo.**

Es el proceso de adecuación a la práctica de la tarifa obtenida, teniendo presente la competencia del mercado asegurador y los posibles clientes a los que se dirige el producto.

2 Los factores de riesgo

Aunque no es labor del presente trabajo el calcular la prima, sino que se centrará en la fase preliminar de selección de los factores de riesgo más influyentes de entre los dados, y la posterior predicción de siniestros, sí conviene comentar en qué consiste básicamente el cálculo de la prima pura, pues éste influye en el tipo de factores de riesgo incluidos en la cartera.

Consideremos una cartera de riesgos, en un intervalo de tiempo que habitualmente se considera un año. Nos interesa estudiar la siniestralidad de la cartera respecto del riesgo cubierto por el seguro en cuestión (en el presente trabajo el seguro del automóvil).

Lo que conocemos como *proceso de riesgo* es el proceso estocástico asociado al acaecimiento de los siniestros y a sus respectivas cuantías [Hossack 1999].

Sean

$$\begin{aligned} &N : \text{Variable aleatoria número de siniestros en el intervalo } \tau \\ &X_i : \text{Variable aleatoria cuantía del siniestro i - ésimo para } i = 1,..., N \end{aligned} \tag{2.1}$$

El coste total por indemnizaciones de siniestros acaecidos, *S*, en el periodo τ, resulta ser

$$S = \begin{cases} X_1 + X_2 + \cdots + X_N & si\ N > 0 \\ 0 & si\ N = 0 \end{cases} \tag{2.2}$$

Si estudiamos el coste total por póliza, su valor esperado viene dado por

$$E[S] = E[N]E[X] \tag{2.3}$$

valor que por definición coincide con la *prima pura*.

$$P = E[S] \tag{2.4}$$

Esta prima pura *P* es la componente fundamental del precio del seguro, ya que está destinada a acumular la recaudación suficiente para hacer frente a los siniestros esperados. Si la cartera fuera infinita, unos riesgos se compensarán con otros y los ingresos obtenidos mediante la prima pura serían suficientes para asegurar la solvencia de la empresa. Sin embargo, en la práctica, no se dispone de una cartera infinita, no siendo además totalmente fiable la información estadística, por lo que es necesario añadir un *recargo de seguridad* a la prima pura para garantizar la solvencia. Gracias a este recargo, se constituye una reserva técnica que permite hacer frente a las desviaciones imprevistas de la siniestralidad que pueden comprometer la solvencia.

Finalmente, es necesario añadir también un *recargo para gastos de gestión interna* (destinado a cubrir los gastos de administración y gestión), un *recargo para gastos de gestión externa* (destinado a cubrir los gastos de producción) y un *recargo para beneficios*, obteniéndose así la *prima comercial o de tarifa***.**

Las primas que resultan una vez considerados todos los recargos deben cumplir los así llamados principios de equidad, solidaridad y suficiencia. Según el *principio de equidad*, las primas pagadas por los asegurados deben ajustarse al riesgo, lo que implica la adecuada selección de los factores de riesgo y la precisión del proceso de cálculo de las tarifas. El *principio de solidaridad* se refiere al pago de una prima idéntica por parte de los asegurados pertenecientes a un grupo homogéneo. Por último, el *principio de suficiencia* pretende que la cuantía total de los ingresos por primas de la empresa aseguradora sea suficiente para hacer frente a los siniestros.

Como hemos dicho antes, los factores del riesgo son las posibles variables independientes o explicativas correlacionadas con la siniestralidad y que permiten explicarla y predecirla. Su conocimiento y cuantificación es un requisito absolutamente esencial para que el proceso de tarificación anteriormente descrito resulte satisfactorio.

En general, el conjunto de factores será una mezcla de variables cuantitativas (discretas o continuas) y cualitativas. En ocasiones los factores continuos se encuentran discretizados de antemano, lo que no necesariamente es una ventaja, pues puede llevarnos a obtener peores resultados que en el caso de haber estado agrupados de otra forma. Es frecuente que durante el proceso de tarificación acabemos realizando una discretización de los datos (la edad del conductor se suele agrupar en intervalos, por ejemplo) para obtener los grupos de tarifa finales, pero no se debe hacer hasta no haber confirmado su relación con la siniestralidad. Por otro lado, el criterio que se utilice para efectuar tal agrupación, es de suma importancia, y objeto de análisis por sí mismo.

Los factores de riesgo podrán hacer referencia tanto a características del objeto asegurado como a otros condicionamientos de éste: características del asegurado, del tomador, condiciones socio-económicas que lo rodean, etc. El número de factores de riesgo necesario para una correcta clasificación difiere en según el tipo de seguros. Así, mientras que en el de vida es suficiente

generalmente con considerar la edad y el sexo de los asegurados, en el del automóvil no es así. Tradicionalmente se utilizan factores como la edad, el sexo, la provincia de residencia, profesión del conductor, antigüedad de su carné de conducir y del vehículo, potencia del motor, etc. Y se recogen muchos más, como el color del vehículo, o el estado civil. Pese a todo, estos factores de riesgo están escasamente correlacionados con su siniestralidad, y explican sólo una pequeña parte de su varianza. Lo ideal sería utilizar factores como la rapidez de reflejos del conductor, sus hábitos, su carácter al volante, si consume alcohol u otro tipo de drogas, etc. Pero esto no resulta posible, ya que esos factores no son observables, o bien el asegurado no está dispuesto a responder, o incluso atentan contra su confidencialidad. Puede incluso que algunos de los factores que tradicionalmente se incluían, y que influyen realmente en la siniestralidad, no pueda recogerse en un momento dado. Es el caso, por ejemplo, del sexo. Este factor, claramente relevante, ha pasado a ser un dato privado sobre el que legalmente no va a estar permitido preguntar. Todo esto puede llegar a complicar bastante el proceso de tarificación.

En el seguro del automóvil, y dependiendo de la cobertura, los factores generalmente considerados son [Boj 2004]:

- *Relativos al vehículo asegurado:* valor, antigüedad, categoría, clase, tipo, marca, modelo, potencia, peso (o relación potencia-peso), color, etc.
- *Relativos al conductor*: edad, sexo, antigüedad del carnet, estado civil, profesión, número de hijos, posibilidad de conductores ocasionales, resultado de la experiencia en el pasado, etc.
- *Relativos a la circulación*: zona de circulación, uso del vehículo, Km. anuales, etc.

2.3 Técnicas estadísticas más utilizadas para la selección de variables de tarifa

Ya hemos establecido que el primer paso dentro del proceso de tarificación a priori es el de selección de las variables de tarifa y sus clases, a partir de unos factores potenciales de riesgo. Vamos a continuación a comentar, muy brevemente, cuáles son las técnicas estadísticas que se emplean usualmente para desempeñar esta labor, terminando así de enmarcar el problema que nos planteamos en la aplicación, al que, como ya dijimos en la Introducción de este trabajo, vamos a sacar de su contexto habitual, y a llevarlo a un nuevo terreno, el de la Inteligencia Artificial. Será en el siguiente Capítulo donde nos detendremos en una de las herramientas que vamos a resumir a continuación, el Análisis Discriminante, que es la elegida para resolver el problema planteado y comparar los resultados con los obtenidos vía la otra clase de métodos.

Las técnicas del análisis estadístico multivariante son las que permiten organizar procesos de selección teniendo en cuenta simultáneamente el conjunto de factores. Recordemos que el objetivo es obtener un conjunto equilibrado de variables de tarifa, aquel que mejor explique la estructura del

riesgo. Si realizáramos la selección considerando separadamente las variables que una a una están más asociadas con el riesgo, es posible que el conjunto de variables seleccionadas resultante contuviera información redundante, o lo que es más importante, que no tuviéramos incorporadas variables que de manera conjunta con otras resulten significativas. Es por ello que precisamos hacer el estudio teniendo en cuenta a la vez todos los factores potenciales de riesgo e "idealmente" todas sus interacciones.

Un criterio de clasificación de este tipo de técnicas es según las fases del proceso de tarificación *a priori* que nos permiten cubrir [Boj 2003]. En este caso, dividimos las técnicas en *predictivas*, que en principio cubrirán todo el proceso de tarificación, y *no predictivas*, que abarcarán sólo alguna fase:

- ***Técnicas predictivas***

    - *Modelos de regresión:* los más utilizados son el Modelo Lineal Generalizado y la Regresión basada en las Distancias. Se incluyen todos los modelos de credibilidad basados en técnicas de regresión que nos permiten realizar una estimación de la siniestralidad a partir de unos grupos homogéneos de riesgo. Ambos modelos permitirán cubrir todas las fases del proceso de tarificación hasta la estimación de la prima pura. Las técnicas de regresión en general consisten en la estimación de la respuesta a partir de una serie de variables explicativas o predictores. Para aplicarlo a la selección de variables de tarifa, se escoge un modelo concreto acorde con los datos disponibles, y se busca mediante un proceso de selección de predictores la "mejor" combinación de ellos para la estimación del riesgo, que pasarán a ser las variables de tarifa.

    - *Técnicas de segmentación:* Cubren todas las fases, aunque su predicción está limitada a las clases ya existentes de los factores categóricos seleccionados. Entre ellas, las más conocidas son Chi-squared Automatic Interaction Detector (CHAID), Theta Automatic Interaction Detector (THAID) y Extended Automatic Interaction Detector (XAID).

- ***Técnicas no predictivas***

    - *Análisis cluster* (*jerárquico aglomerativo* y *no jerárquico, jerárquico de Ward*) Véase [Boj 2003].

    - *Análisis Discriminante:* Resulta útil para la selección de variables y para la formación de los grupos de tarifa que mejor discriminen las poblaciones. Clasifica a los individuos en dos o más poblaciones previamente establecidas según los valores de siniestralidad. Posteriormente, con un proceso de selección de predictores escogemos aquellos que "mejor" discriminan a las poblaciones.

Lo normal es distinguir dos poblaciones, la que no conlleva riesgo, y la que sí lo hace, aunque lo podemos extrapolar a más de dos poblaciones. Es habitual además, como será nuestro caso, basarse en la experiencia del número de siniestros, de modo que la población sin riesgo será la que no tenga siniestros, y la de riesgo la que tenga al menos un siniestro. Al seleccionar las variables que mejor discriminan las poblaciones, lo que cubrimos es el paso de selección de variables de tarifa. Es la misma filosofía que utilizaremos en la aplicación de las técnicas de Aprendizaje Máquina al caso práctico que nos ocupa, y por ello, la seleccionada para realizar la comparación a la hora de clasificar a un individuo en cuanto a si tendrá o no siniestros.

3. DESCRIPCIÓN DE DATOS

Para concluir el Capítulo, procederemos a detallar los datos que serán empleados para ilustrar la aplicación de los métodos de Aprendizaje Máquina a la Selección de Factores de Riesgo, y a la posterior clasificación de los asegurados según hayan tenido o no siniestros durante el año seleccionado.

3.1 Datos relativos a 2003

Se han realizado dos estudios, empleando diferentes técnicas durante el proceso de Selección de factores, y para ambos se han utilizado distintas muestras de la base de datos de la Cartera de Clientes 2003, y Siniestros 2003 que nos proporcionó la compañía MAPFRE, dentro de la "Beca de Riesgos y Seguros 2006". Ambas bases vienen "ordenadas" según el nº de póliza (dato del que evidentemente hemos prescindido), y en primer lugar, nos detuvimos a enlazarlas, de modo que de cada cliente de la muestra sabemos si ha tenido o no siniestro. Las carteras venían dadas en ficheros de Access, con un total de 3943904 asegurados, en un periodo de observación del 1 de Enero de 2003 al 31 de Diciembre de 2003.

Tras depurarlas, se tomaron dos muestras aleatoriamente, una con 58238 asegurados, y la otra con 113.127 clientes, resultado de añadir a la anterior otros 54889 datos, para que este segundo estudio quedara más completo. Hay que mencionar que el objetivo del presente trabajo no es obtener resultados comparativos de las diferentes técnicas, sino observar cómo se comportan estas herramientas en la resolución de un problema del campo actuarial.

La Cartera de Clientes viene descrita por 13 factores de riesgo, dados por la compañía, y sobre cuyo proceso de recogida no hemos tenido ninguna influencia. Son los siguientes: -fecha de carné, -fecha de nacimiento, -permiso, -sexo, -estado civil, -profesión, -fecha de fabricación, -uso, -zona de circulación, -potencia, - valor y marca.

Sin embargo, hay que separar algunos de ellos. Concretamente, las variables consideradas *categóricas* (aquellas en las que el valor no representa cantidad sino clase; ejemplos serían uso, profesión, etc.) y las que no tienen valores numéricos (sean categóricas o no). Además, eliminamos el factor Marca, por la dificultad en cuanto a su tratamiento[2].

De este modo, las variables (109 en total) quedan:

- antigüedad de carnet
- edad del conductor
- permiso: Permiso tipo C, Permiso tipo C1, permiso motocicletas, permiso de conducción de ciclomotor, permiso de conducción de coche, Permiso tipo B2, Permiso tipo D, y Permiso tipo D1
- sexo
- estado civil: soltero, casado, viudo o divorciado
- profesión del conductor
- uso del vehículo
- antigüedad del vehículo
- zona de circulación: Andalucía (Sevilla), Aragón, Asturias, Baleares, Canarias, Cantabria, Castilla-La Mancha, Castilla y León, Cataluña (Barcelona), Extremadura, Galicia, Madrid, Murcia, Navarra, País Vasco, La Rioja y Comunidad Valenciana (Valencia).
- potencia: medida en CV Din -0 significa desconocido-
- valor expresado en decenas de euro: 0 significa que no se han contratado con berturas de daños y no es necesario el dato.

La clasificación de la profesión del conductor siguiendo el criterio de MAPFRE es:

10: AGRICULTORES Y SUS EMPLEADOS
20: INDUSTRIALES, COMERCIANTES, PROFESIONES LIBERALES
(Sin desplazamiento. Profesional habitual)
21: INDUSTRIALES, COMERCIANTES, PROFESIONES LIBERALES
(Desplazamiento. Profesional habitual urbano)
22: INDUSTRIALES, COMERCIANTES, PROFESIONES LIBERALES
(Desplazamiento. Profesional habitual interurbano)
31: VIAJANTES Y REPRESENTANTES URBANOS
32: VIAJANTES Y REPRESENTANTES INURBANOS
40: FUNCIONARIOS Y ADMINISTRATIVOS
(Sin desplazamiento profesional habitual)
41: FUNCIONARIOS Y ADMINISTRATIVOS
(Desplazamiento profesional habitual urbano)
42: FUNCIONARIOS Y ADMINISTRATIVOS
(Desplazamiento profesional habitual interurbano)
53: ESTUDIANTES

[2] La Potencia y el Valor del vehículo recogen gran parte del significado de la Marca.

54: JUBILADOS

55: OBREROS MANUALES

56: SIN PROFESIÓN
57: EMPLEADOS QUE CONDUCEN CON EXCLUSIVIDAD VEHÍCULOS DE LA SOCIEDAD
63: CONDUCTORES DE CAMIÓN VEHÍCULO. INDUSTRIAL DE SU PROPIEDAD
64: CONDUCTOR DE CAMIÓN VEHÍCULO. INDUSTRIAL DE TERCEROS
65: CONDUCTOR DE CAMIÓN VEHÍCULO INDUSTRIAL CONDUCIDOS POR PROPIETARIOS Y EMPLEADOS

La clasificación del uso del vehículo atendiendo a la clasificación de MAPFRE puede ser:

110: TURISMO DE USO PARTICULAR - Uso 110
111: TURISMOS MATRICULADOS A NOMBRE DE EMPRESA - Uso 111
113: TURISMOS ANTIGUOS PARA DESFILES - Uso 113
114: MICROBUSES HASTA 9 PLAZAS USO PROPIO - Uso 114
116: MICROBUSES MATRICULADOS A NOMBRE DE EMPRESA - Uso 116
117: VEHÍCULOS DESTINADOS A USO PROFESIONAL O COMERCIAL - Uso 117
118: TURISMOS USO PARTICULAR +5 HASTA 9 PLAZAS - Uso 118
119: AMBULANCIAS, VEHÍCULOS DE SERVICIO DE URGENCIAS (POLICÍA Y BOMBEROS) - Uso 119
131: TAXI SIN TAXÍMETRO - Uso 131
133: TAXI CON TAXÍMETRO CONDUCIDO EXCLUSIVAMENTE POR PROPIETARIO - Uso 133
135: TAXI CONDUCIDO POR EMPLEADOS - Uso 135
137: MICROBUSES HASTA 9 PLAZAS SERVICIO PÚBLICO - Uso 137
138: TAXI +5 HASTA 9 PLAZAS - Uso 138
141: VEHÍCULOS DE ALQUILER SIN CONDUCTOR - Uso 141
142: VEHÍCULOS DE ALQUILER CON CONDUCTOR - Uso 142
143: VEHÍCULOS DE ALQUILER SIN CONDUCTOR +5 HASTA 9 PLAZAS - Uso 143
144: VEHÍCULOS ALQUILER CON CONDUCTOR +5 HASTA 9 PLAZAS - Uso 144
145: MICROBUSES HASTA 9 PLAZAS DE ALQUILER - Uso 145
150: TURISMOS DE AUTO-ESCUELA - Uso 150
152: MICROBUSES HASTA 9 PLAZAS DE AUTO-ESCUELA - Uso 152
160: VEHÍCULO TODO TERRENO - Uso 160
168: TODO TERRENO +5 HASTA 9 PLAZAS - Uso 168
190: TURISMOS (POLIZA DE PROBADORES) - Uso 190
199: PLACAS DE PRUEBA Y TRANSPORTE 1ª CATEGORÍA – Uso 199
210: FURGONETAS HASTA 3500 KGS - Uso 210
211: FURGONETAS DE TRANSPORTE DE FRUTAS Y HORTALIZAS, RADIO MENOR DE 300 KMS U.P. - Uso 2
212: FURGONETAS DE TRANSPORTE DE FRUTAS Y HORTALIZAS, RADIO MAYOR DE 300 KMS U.P. - Uso 2

213: FURGONETAS TRANSPORTE DE PESCADO EN RADIO MENOR DE 150 KMS U.P. - Uso 213
214: FURGONETAS TRANSPORTE DE PESCADO EN RADIO DE 150 A 300 KMS U.P. - Uso 214
215: FURGONETAS TRANSPORTE DE PESCADO EN RADIO MAYOR DE 300 KMS U.P. - Uso 215
216: FURGONETA TRANSPORTE DE LÍQUIDOS EMBOTELLADOS U.P. - Uso 216
217: FURGONETAS HASTA 500 KGS DE CARGA USO PROPIO - Uso 217
218: VEHÍCULOS DOTADOS DE GRUA USO PROPIO - Uso 218
219: FURGONETAS DE REPARTO URBANO Y AUTOVENTA - Uso 219
220: FURGONETAS DE USO RURAL HASTA 500 KGS DE CARGA USO PROPIO - Uso 220
221: FURGONETAS TRANSPORTE DE FRUTAS Y HORTALIZAS RADIO MENOR DE 300 KMS S.P. - Uso 2
222: FURGONETAS TRANSPORTE DE FRUTAS Y HORTALIZAS RADIO MAYOR DE 300 KMS S.P. - Uso 2
223: FURGONETAS TRANSPORTE DE PESCADO EN RADIO MENOR DE 150 KMS S.P. - Uso 223
224: FURGONETA Transp.. PESCADO RADIO 150 A 300 KMS S.P. - Uso 224
225: FURGONETAS TRANSPORTE DE PESCADO EN RADIO MAYOR DE 300 KMS S.P. - Uso 225
226: FURGONETAS TRANSPORTE DE LÍQUIDOS EMBOTELLADOS SERV. PUBLICO - Uso 226
228: FURGONETAS SERVICIO PUBLICO +5 HASTA 9 PLAZAS - Uso 228
230: FURGONETAS DE USO RURAL HASTA 3500 KGS - Uso 230
231: FURGONETAS DE TRANSPORTE DE MERCANCÍAS NO PELIGROSAS HASTA 150 KMS S.P. - Uso 231
232: FURGONETAS TRANSPORTE MERCANCÍAS NO PELIGROSAS RADIO MAYOR DE 150 KM SP - Uso 23
234: VEHÍCULO. DOTADO DE GRUA HASTA 150 KMS SERV. PUB - Uso 234
235: VEHÍICULO DOTADO DE GRUA MAS 150 KMS SERV. PUBL - Uso 235
238: FURGONETAS USO PARTICULAR +5 HASTA 9 PLAZAS - Uso 238
240: TODO TERRENO USO AGRARIO - Uso 240
241: FURGONETAS USO AGRARIO +5 HASTA 9 PLAZAS - Uso 241
242: TODO TERRENO USO AGRARIO +5 HASTA 9 PLAZAS – Uso 242
250: FURGONETAS DE AUTO-ESCUELA - Uso 250
258: FURGONETAS DE ALQUILER +5 HASTA 9 PLAZAS - Uso 258
290: FURGONETAS (PÓLIZA DE PROBADORES) - Uso 290

La mayoría tienen valores 0 o 1 (por ejemplo, si el cliente tiene un 1 en la variable casado, significa que su estado civil es casado), salvo antigüedad carnet, edad conductor, profesión, uso, antigüedad vehículo, potencia y valor, que tienen las cantidades numéricas que deben indicar.

En el caso del sexo, tomamos 1 si eran hombres, 0 si eran mujeres. el tipo de asignación 0 1 se tratará detenidamente por las implicaciones que ello pueda tener.

En el caso de las variables antigüedad carnet, edad conductor y antigüedad vehículo, hubo que realizar además una transformación[3] para pasar del formato fecha en que nos los proporcionaron, al de "años, meses" en que decidimos presentarlo. Por último, detenernos un momento en el factor *zona de circulación*. Los datos venían dados en función del código postal de cada ciudad española. Sin embargo, separar en base a esto, significaba añadir 52 nuevas variables a la base, lo que redundaría en un peor funcionamiento del sistema.

Por ello, decidimos agruparlos por provincias españolas, quedando así reducido ese número a 17. Pero asociar las zonas de circulación de este modo obliga a pagar un precio, el de no distinguir realmente el riesgo de siniestro dentro de una provincia. Todos somos conscientes de que no se circula igual en todas las ciudades de una misma provincia. Desde luego, hay un mayor número de siniestros en Barcelona, por ejemplo, que en el resto de Cataluña. La solución que adoptamos fue de tipo medio, distinguiendo al menos las grandes capitales de provincia del resto. Así lo hicimos con Barcelona y Valencia, pero, curiosamente, no fue posible con la capital del país, Madrid. Es evidente que la posibilidad de tener un accidente no es la misma en esta ciudad que en Fuenlabrada, por ejemplo. El problema está en que en la Comunidad de Madrid todas las ciudades tienen el mismo código postal, 28, por lo que no es posible diferenciar.

La variable de salida, lo que queremos predecir o en base a lo que clasificamos los clientes, es la de Siniestro, cuyo valor es -1 si no se tiene siniestro, y 1 en caso contrario (el dar estos valores y no otros es por exigencias del programa). Hay que indicar que, aunque sólo hemos tomado el dato de la existencia o no de accidentes, de esta variable MAPFRE nos ha proporcionado además la "Culpa" y el "Importe" (también la "Clase de expediente", pero eso es un dato interno). Las dos se podrían utilizar en ulteriores análisis, muy interesantes a la hora de calcular la prima a pagar por los grupos de asegurados.

En el primer estudio teníamos 28.319 clientes que no habían tenido siniestros en 2003, un poco menos de la mitad. Esta selección no es en absoluto realista en la Cartera de la que disponíamos, ya que la proporción entre el nº de asegurados sin siniestros es muyo mayor que la de con siniestros.

Esto es lo que ocurre además en la realidad de cualquier compañía, pues la probabilidad de presentar siniestro a lo largo de un año es verdaderamente baja. Sin embargo, escogimos de esta manera los clientes para poder entrenar correctamente los clasificadores, pues en otro caso, se nos podría presentar fácilmente la situación de que la tasa de clasificación para el grupo de siniestro fuera muy baja.

[3] La fecha aparecía como "añomes", sin separación. Por ejemplo, la Fecha de Carnet 197704, corresponde a un asegurado que lo obtuvo en Abril de 1977. Para calcular los años y meses que han pasado hasta Diciembre de 2003 (fecha en la que se obtuvieron nuestros datos), hicimos la operación: 2003,12-fecha/100, que en este caso sería 2003,12-197704/100=26,08, esto es, su carnet tiene una antigüedad de 26 años y 8 meses.

En el segundo experimento, nos dimos cuenta después de haber realizado la selección de factores, de que la muestra que habíamos tomado no estaba equilibrada en cuanto al n º de asegurados con y sin siniestros, acorde con la realidad de la que hablábamos antes. Este "error" se resolvió realizando un nuevo experimento en el que se penalizó más clasificar erróneamente un cliente con siniestro que uno sin siniestro, concretamente tratando los errores de siniestro como 2'88 veces más importantes que los demás.

De nuevo, señalar que este tipo de distribución no es realista, en cuanto a que lo habitual es no presentar siniestro. Si lo tomamos así fue en aras de que la aplicación funcionara lo mejor posible, lo que no se hubiera podido lograr tomando los datos tal y como estaban, con la evidente y lógica desproporción en cuanto a la frecuencia de siniestros.

3.2 Datos relativos a 2005

Comentaremos a continuación únicamente las diferencias respecto a los datos anteriores. Éstas se centran especialmente en los factores de riesgo, bastante diferentes de los que manejamos para 2003. En este caso, los factores de riesgo recogidos eran: tipo de vehículo, uso naturaleza, CV, tara, plazas, código postal, nivel BM, ámbito, acuerdo, repara; ocasional, antigüedad del vehículo, edad, antigüedad de carnet, sexo, zona de circulación y diesel.

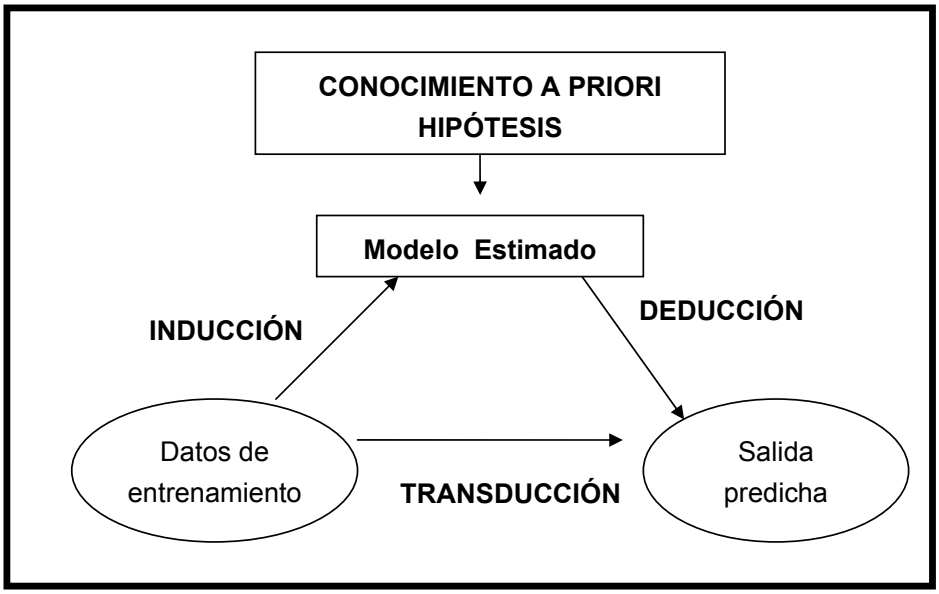
De igual modo que se hizo con los datos de 2003, se procedió a categorizar las variables, generándose dos nuevas bases de datos: una incluyéndose los niveles de Bonus-Malus, y otra sin incluirlos. El motivo de la distinción fue posibilitar el estudio de la influencia de esta nueva variable a la hora de seleccionar los factores, y a la de clasificar.

Por otro lado, en esta ocasión, al permitirlo los datos que se nos había proporcionado, se tomaron muestras teniendo en cuenta el número de siniestros de cada conductor. Esto se utilizó a la hora de clasificar, aunque sin implicaciones relevantes en cuanto a la tasa de clasificación.

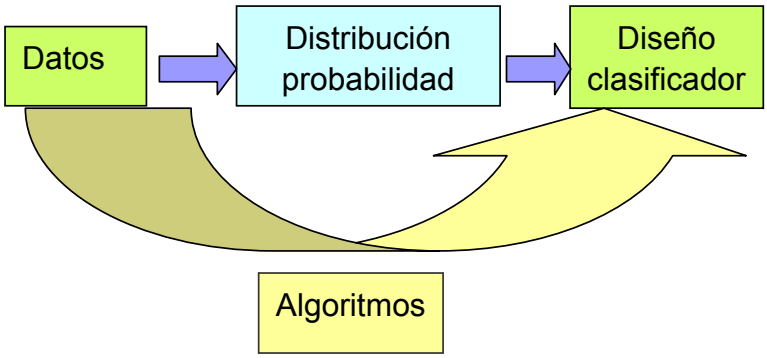
CAPÍTULO 2
TÉCNICAS DE CLASIFICACIÓN

1. INTRODUCCIÓN Y MOTIVACIÓN

El problema que planteamos en el capítulo anterior presenta la misma dificultad inherente a la gran cantidad de datos que necesitamos manejar que la mayoría de los problemas de los negocios, la ciencia y la ingeniería. Recordemos que nuestro objetivo principal es *clasificar* los asegurados de una cartera con casi 10 millones de clientes (en la base original) atendiendo a si han presentado o no siniestro durante un año. Lo ideal sería poder reducir el número de datos necesario para resolver el problema, conservar sólo los que aportan la información más importante. El enfoque estadístico "clásico" a esta situación consiste en asumir que los datos están generados por una *distribución de probabilidad* subyacente que nos es desconocida y que se ha de estimar, dando lugar al diseño del *clasificador*. Eso implica realizar ciertas hipótesis (normalidad, homocedasticidad, etc.) que no sabemos si serán o no ciertas, con el riesgo que eso supone. Sin embargo, existen otras aproximaciones al problema, como la que nos propone Vapnik. La idea básica es diseñar el clasificador directamente desde los datos mediante determinados algoritmos, que en su caso se basan en la Teoría Estadística del Aprendizaje. El siguiente cuadro [Cherkassky 1998] aclara ambas posturas:



Los dos pasos intermedios se corresponden con dos tipos clásicos de inferencia conocidos como inducción (esto es, progresar desde los casos particulares -datos de entrenamiento-, a los generales -dependencia estimada o modelo-) y la *deducción* (progresar de lo general –modelo- a lo particular –valores de salida-). Lo que Vapnik concibe como *transducción* es estimar la salida directamente desde los datos[4]. En el siguiente cuadro, podemos observar el enfoque de Vapnik en verde, y el tradicional en amarillo.



Esta aproximación proporciona, en principio, estimaciones mejores que con el enfoque clásico de inducción/deducción [Vapnik, 1995].

Este modo de abordar el problema nos conduce a la necesidad de analizar la información que comprenden los grandes conjuntos de datos. La habilidad de extraer el conocimiento que se encuentra escondido entre esos datos, y utilizarlo convenientemente, está teniendo una importancia creciente en el mundo contemporáneo. Las aproximaciones más recientes para desarrollar modelos a partir de los datos se han inspirado en las capacidades de *aprendizaje* de los sistemas biológicos, y, en particular, en las de los humanos.

De hecho, los sistemas biológicos aprenden a hacer frente a la desconocida naturaleza estadística del entorno conducidos por los datos. Los humanos, como los animales, tienen la capacidad superior de *reconocer patrones*, como las de identificar caras, voces u olores a partir de los "datos", de la experiencia. El campo del reconocimiento de patrones tiene como objetivo el construir sistemas artificiales que imiten las habilidades de reconocimiento de los humanos, y avanzan hacia él basándose en principios de ingeniería y estadística.

Lo que llamamos un *método de aprendizaje* es un algoritmo (implementado habitualmente en software) que estima una aplicación desconocida

[4] Un caso especial de transducción es el de la conocida *estimación local* , donde la predicción se hace en un único punto.

(dependencia) entre las entradas y las salidas del sistema desde los datos disponibles, esto es, entre las muestras (entrada, salida) conocidas. Una vez que se ha estimado con seguridad tal dependencia, puede emplearse para la predicción de futuras salidas del sistema a partir de los valores conocidos de las entradas.

En estadística, la tarea del aprendizaje predictivo a partir de las muestras recibe el nombre de *estimación estadística*. Tiene por objeto estimar las propiedades de cierta distribución estadística (desconocida) desde las muestras o los datos de entrenamiento. La información contenida en los datos de entrenamiento (experiencia pasada) se puede utilizar asimismo para responder preguntas acerca de las futuras muestras. Por lo tanto, distinguimos dos etapas en un sistema de aprendizaje [Cherkassky 1998]: (1) aprendizaje/estimación (a partir de las muestras de entrenamiento) y (2) operación/predicción, cuando las predicciones se hacen para las muestras futuras o de Test. Esta descripción asume que tanto los datos de entrenamiento como los de test parten de la *misma* distribución estadística subyacente. En otras palabras, esa distribución (desconocida) es fija.

Las tareas específicas de aprendizaje incluyen las siguientes:

- **Clasificación** (reconocimiento de patrones) o estimación de las fronteras de las clases de decisión.
- **Regresión** o estimación de una función continua desconocida desde muestras ruidosas.
- **Estimación** (desde las muestras) de la densidad de la probabilidad.

Nuestro problema entra en la primera de estas categorías. Para abordarlo, como para cualquier tipo de problema de aprendizaje, podemos seguir los siguientes enfoques: el estadístico clásico, o el que nosotros hemos llamado de "aprendizaje máquina".

La aproximación *estadística clásica*, tal y como la propone [Fisher 1952] con su *Análisis Discriminante*, divide el problema de aprendizaje en dos partes: especificación y estimación. La especificación consiste en determinar la forma paramétrica de las distribuciones subyacentes desconocidas., mientras que la estimación es el proceso de determinar los parámetros que caracterizan las distribuciones específicas. La teoría clásica focaliza el problema de la estimación, y deja de lado el de la especificación. Las aproximaciones clásicas dependen de asunciones más estrictas que las que se plantean en la formulación general del aprendizaje, ya que asumen que las funciones se especifican con un número fijo de parámetros. Esto plantea serias limitaciones en muchas de las aplicaciones. Se tiende entonces a introducir un gran margen de error modelizado, la discrepancia entre modelo paramétrico asumido y la verdad desconocida. Como los métodos no paramétricos clásicos, operan sólo en el caso asintótico, que requieren muestras de un tamaño enorme, y nunca tendremos muestras de tamaño suficiente para satisfacer esas condiciones asintóticas.

Las limitaciones de las aproximaciones clásicas motivan la aparición de métodos adaptativos o flexibles. El enfoque que hemos denominado de aprendizaje máquina recoge métodos de la Inteligencia Artificial. La que conocemos como *Máquinas de Vectores Soporte* (SVM) es un procedimiento de aprendizaje universal basado en la teoría estadística del aprendizaje (Vapnik 1995). El término "universal" significa que las SVM se pueden utilizar para aprender una variedad de representaciones, como las redes neuronales (con la *función de activación* sigmoide habitual), las *funciones de base radial*, los *estimadores polinomiales* y los *splines*. Considerándolas de un modo muy general, las SVM proporcionan una nueva manera de parametrización de funciones, por lo que pueden emplearse tanto en aprendizaje como en predicción.

Las SVM fueron desarrolladas inicialmente por V.C. Vapnik para el problema de la clasificación de datos separables. Más tarde se mejoraron para manejar datos no separables, y se adaptaron para resolver el problema de la regresión. Como veremos después, las soluciones del problema pueden ser de tipo lineal (Teorema de Cover) o *kernel* (Vapnik).

La estrategia que adoptan las SVM es la de insertar los datos en un espacio tal que los *patrones*[5] puedan ser descubiertos como relaciones lineales. Ese espacio recibirá el nombre de espacio de características. De este modo, las relaciones lineales se verán entre las imágenes de los datos en el nuevo espacio. Para pasar al nuevo espacio nos valdremos de una función *kernel* (núcleo), que dependerá del tipo de datos que manejemos, así como de las características que pretendamos detectar. Veremos además que el algoritmo que diseñemos se implementará de modo que no precisará las coordenadas de los puntos, sino únicamente el producto interno entre ellos. Esos productos podrán computarse de una manera eficiente directamente a partir de los datos originales con la función kernel.

Por último, y volviendo a lo que comentamos al principio, entre los métodos de aprendizaje que hemos seleccionado, están aquellos dedicados a resolver la tarea inherente a los problemas como el de la clasificación, que es la de la Selección de Características. Entre ellos hemos escogido los Algoritmos Genéticos y los Árboles de Clasificación, que describiremos en el siguiente capítulo. Resolveremos entonces el problema de clasificación creando un algoritmo "híbrido" con una de estas técnicas para la selección de características (tratando así de reducir la dimensionalidad de los datos, y mejorando el rendimiento de la SVM), y la SVM para la clasificación final. Se trata de una práctica muy habitual, como veremos más adelante.

Este marco para nuestro problema da lugar a la estructura de los Capítulos 3 y 4 del presente trabajo. Tras explicar los procesos de Aprendizaje, pasaremos a describir las SVM como técnica de aprendizaje innovadora para aplicar en la

[5] Por patrones entendemos cualquier relación, regularidad o estructura inherente a una fuente de datos.

resolución de nuestro problema de clasificación. Daremos paso entonces al enfoque estadístico clásico con el Análisis Discriminante, que como veremos, está además íntimamente relacionado con el planteamiento dado por las SVM. Hemos de advertir, sin embargo, que la técnica del Análisis Discriminante ha sido elegida para *comparar* los resultados obtenidos en la clasificación con la SVM, ya que se trata de una herramienta habitual en la resolución de los problemas de clasificación en los procesos de tarificación. Por ello, y aunque entra dentro de la teoría del aprendizaje tal y como la hemos planteado, la descripción que desarrollaremos de ella no será tan extensa como la del resto de los métodos que se emplean en este Trabajo.

Dejaremos para el Capítulo 4 los procesos de Selección de Características con los Algoritmos Genéticos y los Árboles de Clasificación, que, como ya hemos apuntado, tiene para nosotros un interés añadido al de la cuestión del número de datos: el de seleccionar los factores de riesgo relevantes para el proceso de tarificación.

Una vez concluida la exposición teórica, procederemos a comentar los resultados prácticos. Lo haremos con cada técnica por separado, comparando las soluciones en el último apartado de cada capítulo.

Para comprobar el comportamiento de nuestros clasificadores, se calculará la *tasa de error* dividiendo nuestro conjunto de datos en dos partes, el conjunto de entrenamiento y el de test. Para el cómputo de la tasa de error existen varios métodos, y nosotros hemos seleccionado el conocido como *Validación Cruzada k-fold* (en " *k* pliegues" o "*k* frente al resto), por ser la técnica que mejores resultados produce. Se trata de un método de estimación basado en un "re-muestreo".

El algoritmo consiste en separar las observaciones en *k* subconjuntos de aproximadamente el mismo tamaño, clasificarlos *k* veces, reservando en cada ocasión uno de los conjuntos de entrenamiento para utilizarlo en el cómputo de la tasa de error. La implementación es la siguiente:

1. Separar las muestras en *k* subconjuntos de aproximadamente el mismo tamaño $\mathbf{Y} = \{\mathbf{y}_i, i = 1,...,k\}$.

2. Generar el clasificador con el conjunto de datos dejando cada vez fuera del proceso uno de los *k* subconjuntos $\mathbf{Y}_j^C$, donde $\mathbf{Y}_j$ es el *k*-ésimo subconjunto e $\mathbf{Y}_j^C$ su subconjunto complementario

$$\mathbf{Y}_j = \left\{ \mathbf{y}_i \; i = 1,..., \left\lfloor \frac{1}{k} \right\rfloor \right\}$$

$$\mathbf{Y}_j{}^C = \left\{ \mathbf{y}_i \; i = 1,..., \left\lfloor \frac{n(k-1)}{k} \right\rfloor \right\}, \quad \mathbf{Y} = \mathbf{Y}_j \cup \mathbf{Y}_j{}^C$$

3. Calcular la *j*-ésima tasa de error *k-fold* e_j^k usando sólo el subconjunto excluido.

Repetir estos pasos *k* veces, siendo la tasa de error finalmente $e^k = \frac{1}{k}\sum_{j=1}^{k} e_j^k$.

Con este procedimiento, se puede además construir lo que se conoce como Matriz de Confusión. Esta matriz contiene información sobre las clasificaciones hechas por el clasificador, tanto las reales como las que va a predecir. El comportamiento de este tipo de sistemas se suele evaluar utilizando los datos de la matriz. Un ejemplo de lo que sería la matriz de confusión para un clasificador de dos clases como el nuestro la podemos contemplar en la siguiente tabla.

PREDICCIÓN CORRECTA	**MAL (M)**	**BIEN ((B)**
MAL (M)	*a*	*b*
BIEN (B)	*c*	*d*

a y d representan el número correcto de predicciones de mal y bien, respectivamente, y b y c el número incorrecto de predicciones de mal y bien.

En los experimentos que realicemos, utilizaremos esta matriz para expresar los resultados siempre que sea posible.

2. LA TEORÍA DEL APRENDIZAJE Y LA CLASIFICACIÓN

2.1 Introducción

Las máquinas de clasificación, representadas principalmente por las Redes Neuronales, se han empleado para la resolución de problemas ligados principalmente con la Minería de Datos, en áreas tan dispares como la Ingeniería de Telecomunicaciones, el reconocimiento de caracteres, la Economía y las Finanzas, en multitud de ocasiones durante las dos últimas décadas. Así, ya es habitual el encontrarse en problemas de estudio de la insolvencia en empresas (predicción de la situación de quiebra, clasificación de empresas en solventes y no solventes, etc.), rating (clasificación de empresas atendiendo a determinados criterios contables), e incluso en campos tan intrépidos como el Mercado de Valores [Tolmos *et al*.] con soluciones vía este tipo de técnicas. Es la primera de estas áreas en la que parecen haber encontrado un campo especialmente abonado los investigadores del mundo del aprendizaje, y es efectivamente en la que se pueden hallar un mayor número de trabajos. Incluso se han realizado ya aplicaciones de la última y (de momento) más avanzada de estas técnicas, las SVMs [Segovia-Vargas *et al* (2004)]. Sin embargo, fuera de este aspecto, su empleo en el campo actuarial es escaso. Es por ello que nos ha parecido interesante el utilizar estas

herramientas que han demostrado sobradamente su eficacia en las áreas económicas arriba mencionadas, y especialmente, en problemas complejos de la ciencia en general.

La construcción de máquinas capaces de *aprender* de la experiencia ha sido objeto de discusiones tanto filosóficas como científicas. Más allá de la ciencia ficción, el aprendizaje máquina es hoy un hecho, gracias a la irrupción en escena de los ordenadores electrónicos, cuya enorme capacidad ha probado que las máquinas pueden mostrar un nivel de aprendizaje más que significante (si bien los límites de esa habilidad estén aún lejos de definirse con claridad).

En un sentido amplio, cualquier método que incorpore información de las muestras de entrenamiento en el diseño de un clasificador emplea aprendizaje. La creación de clasificadores implica el planteamiento de una forma general de modelo, o de clasificador, y el uso de los datos de entrenamiento para *aprender* o estimar los parámetros desconocidos del modelo. Cuando hablemos aquí de *aprendizaje* lo haremos como una forma de algoritmo para reducir el error sobre el conjunto de entrenamiento. Concretamente, un *algoritmo de aprendizaje* es aquel que toma los datos de entrenamiento como entrada (*input)* y selecciona la hipótesis (la función candidata de entre todas a ser la que relaciona las salidas con las entradas, esto es, la *función de decisión*) del espacio de hipótesis.

El poder disponer de sistemas de aprendizaje seguros tiene una gran importancia, ya que, al no tener la posibilidad de contar con modelos matemáticos adecuados, son muchos los problemas que no se alcanzan a resolver con las técnicas clásicas de programación. Es el caso, por ejemplo, del reconocimiento de caracteres escritos, campo en el que las técnicas de aprendizaje que luego mostraremos, han probado sobradamente su habilidad.

El problema del aprendizaje es tan general que casi cualquier cuestión que se haya discutido en la ciencia estadística tiene su homóloga en la teoría del aprendizaje. Es más, algunos resultados generales de relevancia fueron establecidos en el contexto del aprendizaje y luego reformulados en términos de estadística. Por ejemplo, fue en este entorno en el que por vez primera se destacó la importancia del tamaño pequeño de la muestra estadística: se probó que si se tenía en cuenta el tamaño de la muestra, los resultados a muchos problemas de estimación de funciones eran mejores que los obtenidos con las técnicas estadísticas clásicas. La teoría abstracta del aprendizaje establece condiciones para la generalización que son más generales que aquellas que se tratan en el seno de la estadística clásica. La comprensión de esas condiciones inspiró aproximaciones algorítmicas nuevas a los problemas de estimación de funciones. Históricamente, el problema del aprendizaje a partir de los datos era el equivalente filosófico al de la “inferencia inductiva”. Aunque pueda parecer absurdo, no fue hasta el siglo XX el que se reconoció la inducción pura como tarea imposible salvo que se asumiera un conocimiento a priori, logro conceptual debido fundamentalmente al trabajo de Karl Popper [Popper 1932].

En el ámbito de la estadística, podemos establecer el origen del estudio de este problema en las aproximaciones de Fisher a la clasificación en los años 30 del pasado siglo. Pero es en el área de la inteligencia artificial donde encontrará su más fructífero campo de cultivo. Aquí los investigadores comenzaron a lidiar con el problema del aprendizaje desde su comienzo. Alan Turing [Turing 1950] propuso la idea de las máquinas de aprendizaje en 1950, refutando la afirmación de Lady Lovelace en cuanto a que "las máquinas sólo pueden hacer lo que sabemos cómo ordenarlas hacer".

Turing comenta además en ese escrito una de las principales críticas que se les ha realizado a estos modelos, por la que reciben el apelativo de "caja negra": "Un hecho importante de una máquina de aprendizaje es que su profesor a menudo será enormemente ignorante sobre cómo va todo dentro, aunque puede que sea capaz, hasta cierto punto, de predecir el comportamiento de su alumno."

Tan sólo unos pocos años después verán la luz las primeras máquinas de aprendizaje, como el juego de damas de Arthur Samuel [Samuel 1959] (ejemplo pionero del aprendizaje de refuerzo), o el Perceptrón de Rosenblatt. Son los nacimientos de estas máquinas los que caracterizan la historia de la investigación del problema del aprendizaje, que pasa de la construcción de las primeras máquinas de aprendizaje, a la de los fundamentos teóricos, siguiendo con la creación de las redes neuronales, y, finalmente con la construcción de las alternativas a las redes neuronales.

El análisis matemático de los procesos de aprendizaje comenzó realmente con la creación del primer modelo de máquina de aprendizaje, el Perceptrón. Hacía más de treinta y cinco años que F. Rosenblatt había propuesto este modelo, basado en una idea que ya había sido discutida en la literatura de la neurofisiología durante muchos años. Lo novedoso del enfoque de Rosenblatt fue que describió el modelo como un programa de ordenador, y demostró con experimentos sencillos que ese programa se podía generalizar.

La teoría estadística del aprendizaje (más formalmente conocida como teoría VC, de Vapnik-Chervonenkis) enlaza con lo anterior irrumpiendo en escena a finales de 1960. Tan pronto como los experimentos del Perceptrón se afianzaron y se dieron a conocer, surgieron otros tipos de máquinas de aprendizaje (como el *MADALINE* de B. Widrow, o las *matrices de aprendizaje* de K. Steinbuch, aunque se construyeron más bien como hardware especial de aprendizaje). Sin embargo, a diferencia del Perceptrón, se consideró a esas máquinas como herramientas para resolver problemas de la vida real, más que como un modelo general del fenómeno del aprendizaje.

Con el mismo objetivo se diseñaron muchos programas de ordenador, incluyendo software para construir funciones lógicas de diferentes tipos (por ejemplo, árboles de decisión, elaborados originalmente para sistemas expertos), o modelos de Markov ocultos. Pero estos programas, como las máquinas, tampoco afectaron al estudio del fenómeno general del aprendizaje.

El siguiente paso en la generación de un tipo general de máquina de aprendizaje fue dado en 1986, cuando se empleó la técnica llamada de retropropagación[6], que marcó una nueva era en la historia de las máquinas de aprendizaje.

A diferencia del análisis aplicado, donde no ocurrió nada relevante entre la construcción del perceptrón (1986)y la implementación de la técnica de retropropagación (1989), estos fueron años extremadamente fructíferos para el desarrollo de la teoría del aprendizaje, estableciéndose conceptos como el de la entropía VC y la dimensión VC, que dieron paso al desarrollo de la Ley de los Grandes Números, y al principio de minimización del riesgo estructural., así como al método de máxima verosimilitud, ya en 1989.

La comprensión de la esencia del proceso de aprendizaje no se benefició demasiado de los más de diez años de investigación en las redes neuronales. Habría de esperarse a la siguiente década para que la exploración en este campo comenzara a dar los frutos esperados.

En los últimos años se presta más atención a las alternativas a las redes, como puedan ser las funciones de base radial. Como en 1960, las redes neuronales recuperan su nombre de perceptrones multicapa. Las partes más avanzadas de la teoría estadística del aprendizaje atraen en estos últimos tiempos a más investigadores. Parece como si todo regresara a sus orígenes.

Además, la teoría estadística del aprendizaje juega ahora un papel más importante: tras concluir el análisis general del proceso de aprendizaje, ha comenzado la investigación en el área de la síntesis de los algoritmos óptimos. Sin embargo, estos estudios todavía no forman parte de la historia. Son procesos vivos, objeto de investigaciones que se desarrollan en la actualidad.

En los últimos años han surgido nuevas ideas en la metodología del aprendizaje, inspiradas en la teoría estadística del aprendizaje. Ahora ya no tratan de emular a la biología en el proceso de aprendizaje, sino que su inspiración procede de los intentos de minimizar los límites teóricos de la tasa de error que se obtiene como resultado del análisis formal del proceso de aprendizaje. Estas ideas han dado lugar a algoritmos que no sólo cuentan con propiedades matemáticas deseables, sino que además muestran un funcionamiento excelente. Son las Máquinas de Vectores Soporte, que trataremos más adelante en este capítulo.

Los métodos prácticos son en la actualidad el resultado de un análisis teórico profundo de los límites estadísticos más que el resultado de una heurística más o menos ocurrente, lo que ha conducido a una situación metodológica nueva. Es un hecho que ha transformado el carácter del problema de aprendizaje.

[6] Este algoritmo, que luego enunciaremos, se utiliza para encontrar simultáneamente los pesos de varias neuronas, proceso que hasta el momento se realizaba fijando los pesos de todas las neuronas menos de la última, cuyo coeficiente se establecía durante el proceso de aprendizaje.

2.2 Tipos de aprendizaje

El aprendizaje puede darse en diferentes formas:

- *Aprendizaje Supervisado*

El elemento esencial es la disposición de un "profesor externo" con conocimiento del entorno, que se representa mediante unos *ejemplos de entrada-salida.*

Ese entorno es sin embargo desconocido para el algoritmo de aprendizaje. Supongamos que profesor y algoritmo son sometidos a un vector de entrenamiento (*ejemplo*); el profesor proporciona entonces al sistema la respuesta deseada para ese vector de entrenamiento, que representa la acción óptima a desarrollar por el modelo. Los parámetros del modelo se ajustan bajo la influencia combinada del vector de entrenamiento y la señal de error (diferencia entre la respuesta actual y la deseada). Este ajuste se lleva a cabo de modo iterativo, con la intención de que el algoritmo eventualmente emule (será óptimo en algún sentido estadístico) al profesor.

La desventaja más importante de este método de aprendizaje es el hecho de que sin profesor el modelo no puede aprender nuevas estrategias para situaciones particulares que no están cubiertas por el conjunto de ejemplos utilizados para entrenar el modelo. Esta limitación la subsana el tipo de aprendizaje que veremos a continuación.

- *Aprendizaje de refuerzo*

Es el aprendizaje on-line de una aplicación de entrada salida a través de un proceso de ensayo y error diseñado para maximizar la *señal de refuerzo*. Más correctamente, lo podemos definir del siguiente modo: *"Si una acción tomada por un sistema de aprendizaje es seguida por un estado satisfactorio de sucesos, entonces la tendencia del sistema a producir esa acción particular se ve reforzada. En otro caso, esa tendencia se debilita".* [Sutton et al.,1991];[Barto 1992]. La idea básica que está detrás del aprendizaje reforzado es construir una función que sea la medida natural del desarrollo del sistema de aprendizaje, la *función de evaluación* y aprender de ella de modo que seamos capaces de predecir el refuerzo acumulativo que se recibirá en el futuro.

- *Aprendizaje no supervisado*

En el aprendizaje no supervisado, o *autorganizado*, no hay profesor externo que supervise el proceso. En otras palabras, no hay ejemplos de la función de los que el modelo pueda aprender. Es más, esa provisión se realiza mediante una medida *de tarea independiente* de la calidad de la representación que se

pretende que el modelo aprenda, y los parámetros libres del modelo se optimizan respecto a esa medida. Una vez que el modelo ha comenzado a coger el tono dentro de las regularidades estadísticas de los datos de entrada, desarrolla la habilidad de formar representaciones internas para figuras codificadas de las entradas, y así crear nuevas clases automáticamente.

2.3 Los tres principales problemas de aprendizaje

Muchos de los problemas que se plantean en el entorno de la Inteligencia Artificial son extremadamente complejos, haciendo muy difícil, e incluso imposible, su resolución explícita de un modo programado. Los Sistemas de Aprendizaje proporcionan una metodología alternativa a la hora de abordarlos. Al explotar el conocimiento extraído de una muestra de datos, a menudo son capaces de adaptarse para inferir una solución a esas tareas.

Lo que consideramos en este trabajo como un "problema de aprendizaje" es el problema de encontrar una dependencia deseada utilizando un número limitado de observaciones. Podemos describir asimismo el modelo general de aprendizaje a partir de los ejemplos, en el sentido expresado en el apartado anterior, a través de tres componentes:

(i) Un generador (G) de vectores aleatorios $x \in \Re^n$, extraídos independientemente de una función de distribución de probabilidad $F(x)$ fija pero desconocida.

(ii) Un supervisor (S) que devuelve un valor de salida *y* a cada vector de entrada *x*, de acuerdo a una función de distribución condicional $F(y/x)$ (esto en el caso más general, que incluye también el caso de que el supervisor emplee una función $y = f(x)$), también fija pero desconocida.

(iii) Una máquina de aprendizaje (MA) capaz de implementar un conjunto de funciones $f(x,\alpha), \alpha \in \Lambda$, donde Λ es un conjunto determinado de parámetros[7].

El problema del aprendizaje consiste en escoger del conjunto de funciones dado $f(x,\alpha), \alpha \in \Lambda$, aquella que mejor aproxime la respuesta del supervisor.

La selección de la función deseada está basada en un conjunto de entrenamiento de *l* observaciones independientes e idénticamente distribuidas extraídas de acuerdo a $F(x,y) = F(x)F(y/x)$:

$$(x_1, y_1),...,(x_l, y_l) \tag{2.1}$$

[7] Obsérvese que, al ser los parámetros α de cualquier tipo (ni siquiera deben necesariamente ser vectores), estamos considerando cualquier conjunto de funciones.

Con el objeto de resolver el problema del aprendizaje tal y como indicamos antes, uno mide la *pérdida*, o discrepancia, $L(y, f(x,\alpha))$ entre la respuesta *y* del supervisor a una entrada dada *x*, y la que proporciona la máquina, $f(x,\alpha)$. Consideremos el valor esperado de la pérdida, dado por el *funcional de riesgo*

$$R(\alpha) = \int L(y, f(x,\alpha))\, dF(x,y) \tag{2.2}$$

El objetivo es encontrar la función $f(x,\alpha_0)$ que minimiza el funcional del riesgo $R(\alpha)$ (sobre la clase de funciones $f(x,\alpha), \alpha \in \Lambda$) cuando la función de distribución de probabilidad $F(x,y)$ es desconocida, y la única información disponible es la contenida en el conjunto de entrenamiento (2.1).

En esta línea de razonamiento, concretamos un poco más y establecemos los que se consideran los tres principales problemas de aprendizaje: el Reconocimiento (o análisis) de Patrones, la estimación de la regresión, y la estimación de la densidad.

- *Análisis de Patrones*

El Análisis de Patrones es el proceso para detectar automáticamente relaciones generales entre los datos de un conjunto dado, y forma parte del corazón de multitud de disciplinas, desde las Redes Neuronales hasta la conocida como Reconocimiento de Patrones Sintáctico, desde el reconocimiento de patrones estadístico hasta el aprendizaje máquina y el "data mining" (minería de datos). Sus aplicaciones abarcan desde la bioinformática a la recuperación de documentos. Cuando nos referimos a "datos", lo hacemos pensando en el resultado de cualquier observación, medida o registro. Por patrones entendemos cualquier relación, regularidad o estructura inherente a una fuente de datos. Al hallar patrones significantes en los datos disponibles, el sistema tiene la posibilidad de hacer predicciones sobre datos nuevos procedentes de la misma fuente. En este sentido el sistema ha adquirido el poder de generalización que proporciona el *aprendizaje* de algo sobre la fuente que generó los datos. Los patrones detectados son de muchos tipos, tales como clasificación, regresión, análisis cluster, extracción de características, inferencia gramatical, etc.

Aunque el campo usual de aplicación se encuentra en el ámbito de la Ingeniería, el análisis de patrones resulta muy útil en multitud de disciplinas. Tomemos por ejemplo un problema financiero clásico, el de la morosidad en los préstamos, esto es, la predicción de cuándo el prestatario va a dejar de pagar el préstamo basada en la información disponible hasta el momento sobre ese préstamo. No es posible el realizar una predicción exacta, debido a la multitud de factores extrínsecos e incontrolables que se pueden dar, sí podemos encontrar relaciones que se den con una cierta probabilidad. Ese tipo de relaciones se llaman "estadísticas", y se encuentran donde los sistemas de aprendizaje han probado sus dotes con éxito.

Tradicionalmente existe una distinción entre el reconocimiento de patrones *estadístico* y el *sintáctico*. El primero maneja básicamente vectores y lo hace bajo hipótesis estadísticas en cuanto a su distribución, mientras que el segundo trabaja con objetos estructurados (lenguajes formales, secuencias,…) y se basa mucho menos en el análisis estadístico. Sin embargo, se trata de dos direcciones reconciliables, como luego veremos.

Queremos, por tanto, diseñar algoritmos de análisis de patrones con vistas a emplearlos para realizar predicciones sobre datos nuevos, que no se le habían presentado al sistema con anterioridad. El conjunto que recoge tales datos es conocido con el nombre de *conjunto de test*. El buen funcionamiento del algoritmo se evalúa analizando su comportamiento sobre tal conjunto.

El objetivo más frecuente del análisis de patrones es el de predecir una característica de los datos como función de los otros valores de las mismas. El análisis suele aislar la característica que tenemos intención de predecir, de modo que los datos de entrenamiento aparecen de la forma : $(\mathbf{x}, y)$

donde y (*salida* o *etiqueta*) es la característica que el sistema pretende predecir, y $\mathbf{x}$ (*entrada*) es el vector que contiene los valores de las restantes características. El análisis de patrones cuyos datos están descritos de esta forma es *supervisado*, en cuanto a que cada entrada tiene asociada su correspondiente etiqueta.

En los términos en que expusimos antes los problemas de aprendizaje, *y* es la salida del supervisor. Consideramos $f(x,\alpha), \alpha \in \Lambda$ el conjunto de funciones indicador.

Definimos la *función de pérdida* $f(\mathbf{x}, y) = L(y, g(\mathbf{x}))$ con *g* la función de predicción.

Para esta función, el funcional definido en (2.2) determina la probabilidad de las diferentes respuestas dadas por el supervisor y por la función característica $f(\mathbf{x}, y)$. Denominaremos *error de clasificación* el caso de respuestas diferentes. Obviamente, esperamos que la pérdida tome un valor cercano al cero cuando se detecte una característica. Cada vez que se presenta un dato nuevo, la etiqueta no está disponible, y nos valdremos de $f(\mathbf{x}, y)$ para averiguar el valor de *y* para la entrada dada **x** usando *g*(**x**). La predicción de que $f(\mathbf{x}, y)$ = 0 indica que la discrepancia entre *g*(**x**) e *y* es pequeña.

De este modo, el problema consiste en encontrar una función que minimice la probabilidad del error de clasificación cuando la medida de probabilidad $F(x, y)$ es desconocida, pero los datos (2.1) están dados.

El tipo de salida *y* permite distinguir entre las diferentes aplicaciones del análisis de patrones. Así, en la *Clasificación Binaria y* toma sólo dos valores, que habitualmente son $\{-1,1\}$, de modo que si el vector **x** pertenece a la

categoría escogida obtendremos *y* = +1, y si no es así, $y = -1$. En este caso, haremos uso de una función de pérdida discreta, que devolverá 1 si los dos argumentos difieren y -1 en otro caso. Cuando los datos de entrenamiento están etiquetados para pertenecer a una de *N* clases, y la misión del sistema es aprender a asignar datos nuevos a su correspondiente clase, entonces escogeremos *y* dentro del conjunto $\{1,2,...,N\}$, y estaremos hablando de una *Clasificación Multiclase*.

Por último, se tratará de Regresión cuando la característica a determinar toma valores reales, $y \in \mathbb{R}$, incluso cuando la etiqueta es un vector ($\mathbf{y} \in \mathbb{R}^n$).

Veamos a continuación, muy brevemente, una descripción de los otros dos problemas de aprendizaje a los que nos referíamos antes.

- *Estimación de regresión*

Sea la respuesta del supervisor *y* un valor real, y sea $f(x,\alpha), \alpha \in \Lambda$ el conjunto de funciones reales que contienen a la función de regresión

$$f(x,\alpha_0) = \int y \, dF(y/x)$$

Se sabe que la función de regresión es la única que minimiza el funcional (3.2) con la siguiente función de pérdida

$$L(y, f(x,\alpha)) = \big(y - f(x,\alpha)\big)^2 \tag{2.3}$$

En este caso, el problema de la estimación de regresión es el de minimizar el funcional de riesgo (2.2) con la función de pérdida (2.3) cuando, de nuevo, la medida de probabilidad $F(x,y)$ es desconocida, pero los datos (2.1) están dados.

- *Estimación de densidad (Ajuste de Fisher-Wald)*

Por último, consideremos el problema de la estimación de la densidad a partir del conjunto de densidades $p(x,\alpha), \alpha \in \Lambda$. Para resolverlo, definimos la función de pérdida,

$$L(p(x,\alpha)) = -\log p(x,\alpha) \tag{2.4}$$

Es un hecho conocido que la densidad deseada minimiza el funcional (2.2) con esta función de pérdida. Por ello, otra vez, para estimar la densidad a partir de los datos, debemos minimizar el funcional bajo la condición de que la correspondiente medida de probabilidad $F(x)$ es desconocida pero los datos independientes e idénticamente distribuidos (2.1) sí están dados.

Como se puede apreciar, los tres problemas comparten un objetivo en común, tal y como avanzábamos antes cuando describíamos el problema general del aprendizaje: minimizar el funcional del riesgo. Sin embargo, con vistas a la aplicación que realizaremos en el capítulo final, vamos a dedicar un poco más de atención al primero de los problemas que hemos descrito, el del reconocimiento de patrones.

2.4 Clasificación

1. Métodos de Clasificación basados en el Aprendizaje

Como ya hemos visto, el Aprendizaje juega un papel crucial en el diseño de los clasificadores. La aproximación general a un problema de esta índole es especificar un modelo que cuenta con determinados parámetros a estimar a partir de los datos. Ha de tenerse en cuenta que cuanto más complicado sea el modelo, menor será el conocimiento a priori, y cuanto menor sea el conjunto de entrenamiento, mayor será la confianza que tengamos que depositar en una búsqueda sofisticada para encontrar unos parámetros aceptables para el modelo. Por ello, la máquina de aprendizaje que se construya debe controlar la generalización, minimizando el funcional del riesgo y teniendo en cuenta para ello el tamaño del conjunto de entrenamiento, evitando asimismo la sobrealimentación. Por supuesto, la máquina de aprendizaje ideal para el problema será aquella que presente el menor error en la clasificación de los objetos nuevos.

Con estos objetivos en mente, vamos a estudiar diferentes modelos que se pueden aplicar en problemas de clasificación. Primero trataremos algunos de los principales exponentes de los "algoritmos individuales", que abarcan técnicas tales como las Redes Neuronales (supervisadas o no), el Razonamiento Probabilística, la Lógica Difusa, los Algoritmos Genéticos y Evolutivos, el Razonamiento Basado en casos, la Teoría de la Regresión y la Reducción de Datos, los Sistemas Expertos, los Árboles de Decisión y regresión, las Técnicas Clustering, la Vida Artificial, o las Máquinas de Vectores Soporte. Junto a ellas están los *Sistemas Híbridos,* en los que se usa más de una técnica de las anteriores para resolver un problema, y que vamos a comentar brevemente a continuación.

- *Técnicas individuales*

Como sucede en multitud de ocasiones, la aproximación lineal a un problema, además de fácil de comprender, resulta extremadamente útil para resolver los casos más sencillos, y servir de base para la posterior generalización. Por ello, comenzaremos por describir dos máquinas de aprendizaje que son generalizaciones de aquellas máquinas con funciones indicador lineal que se construyeron en los años 1960: las *Redes Neuronales Artificiales*, inspiradas en la analogía biológica con el cerebro, y las *Máquinas de Vectores Soporte*, que

surgen de la teoría estadística del aprendizaje. Son lo que se conocen como máquinas de aprendizaje lineal, que luego extenderemos al caso no lineal.

Ambas suelen trabajar en un entorno de aprendizaje supervisado, en el que se les proporciona un conjunto de ejemplos de entrenamiento con unas etiquetas (valores de salida) asociadas, que clasificaran (en dos clases en general, pero veremos también la extensión a múltiples clases) empleando diferentes algoritmos.

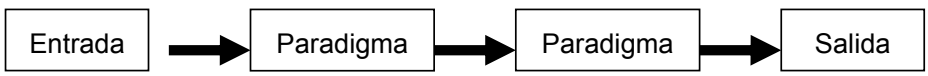
Otro tipo de métodos son los que llamamos *estocásticos*, en los que la aleatoriedad juega un papel crucial para la búsqueda y el aprendizaje. Dentro de ellos cabe destacar dos clases de métodos: aquellos basados en conceptos y técnicas de la física (concretamente de la mecánica estadística), representados por el *aprendizaje de Boltzmann*, y los que se basan en ideas de la biología (la teoría matemática de la evolución), cuyo máximo exponente son los *Algoritmos Genéticos.*

En todos los métodos anteriores, el problema de reconocimiento que se plantea se fundamenta en patrones representados por vectores evaluados por números reales y discretos. Pero supongamos que el problema de clasificación trabaja con valores *nominales*, por ejemplo descripciones que son discretas y sin ninguna noción de similaridad u orden aplicable. Un modo natural e intuitivo de clasificar esas muestras descritas por listas de atributos es mediante una secuencia de preguntas, en la que la siguiente pregunta dependa de la respuesta de la anterior. Es lo que se conoce con el nombre de *árbol de decisión,* o clasificación, que también incluiremos en las técnicas utilizadas en nuestros experimentos.

- *Sistemas Híbridos*

La colaboración entre los paradigmas empleados para dar o mejorar la solución al problema planteado puede prestarse de tres formas básicas[8]:

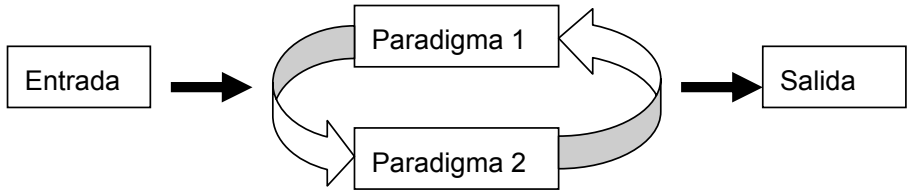
1. *Hibridación secuencial:* es la forma más débil de hibridación, y en ella uno pasa la salida a otro:



Un ejemplo clásico sería dar la salida de un pre-procesador estadístico como entrada a una Red Neuronal.

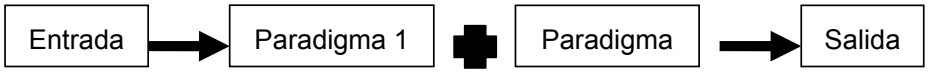
2. *Hibridación auxiliar:* el nivel de hibridación es más alto pues en este caso dos sistemas existen y el primero llama al segundo:

[8] Suponemos dos paradigmas, aunque puede ser mayor que dos.



Es muy habitual encontrarse esta estructura cuando una RNA llama a un Algoritmo Genético para optimizar su estructura (calcular o mejorar sus pesos, etc.).

3. *Hibridación Embutida:* en este último caso un paradigma contiene a otro.



Un ejemplo lo encontramos en una red neuronal que simula algún aspecto de un Sistema Difuso.

Entre los modelos híbridos más utilizados hallamos el de la combinación de las Redes Neuronales y los Algoritmos Genéticos, empleados principalmente para:

- Aprender los pesos de las conexiones: pueden sustituirse los algoritmos basados en el gradiente por los Algoritmos Genéticos, o realizarse un entrenamiento mixto, en el que los Algoritmos Genéticos se usan para buscar zona o pesos iniciales buenos, y se afina mediante un algoritmo de retropropagación.

- Optimizar la estructura y/o las reglas de Aprendizaje de la Red Neuronal

- Selección evolutiva de las características.

- Tomar una Red como estimador de la capacidad en el Algoritmo Genético.

- Diseñar varias Redes Neuronales Evolutivas y conservar la mejor.

- Valerse de un Algoritmo Genético para tener buenas reglas de aprendizaje por refuerzo.

Otros modelos híbridos reúnen a las RNA con la estadística, ya sea mediante comparaciones entre ambos, el uso de métodos estadísticos para definir el modelo de la Red o la implementación de métodos estadísticos por medio de Redes Neuronales.

También nos gustaría destacar la simbiosis entre las Redes Neuronales y las SVM, y entre las SVM y los Algoritmos Genéticos

2 Representación matemática de un problema de clasificación

Matemáticamente, en el mismo sentido en el que definíamos antes el problema de aprendizaje, la representación de un problema de clasificación con múltiples atributos es la siguiente [Schölkopf 1999]: se quiere estimar una función (de decisión) *f:* $\mathbb{R}^n \to \{\pm 1\}$ empleando datos (observaciones u *objetos*) del conjunto de observaciones que se utilizará para *entrenar* lo que hemos llamado "máquina de clasificación".

Consideraremos para ello una serie de objetos $\{\mathbf{x}_i\}, \mathbf{x}_i \in \mathbb{R}^n, \quad i \in \{1,...,l\}$ generados por cierta función de distribución de probabilidad desconocida $P(\mathbf{x}, y)$, a los que asociaremos una etiqueta y_i del conjunto $\{-1,1\}$. De este modo, el conjunto de datos considerados sería:

$$(\mathbf{x}_1, y_1),...,(\mathbf{x}_l, y_l) \in \mathbb{R}^n \times \{\pm 1\}, \tag{2.5}$$

y el objetivo es que la función *f* clasifique correctamente los ejemplos nuevos que se la presenten $(\mathbf{x}, y)$, esto es, que $f(\mathbf{x}) = y$ para ejemplos $(\mathbf{x}, y)$ generados por la misma distribución de probabilidad *subyacente* $P(\mathbf{x}, y)$ que los datos utilizados para el entrenamiento.

El concepto de la función *f* se puede *ampliar* para dar lugar al de *máquina de clasificación*, que queda así definida como el conjunto de todas las posibles aplicaciones $\mathbf{x} \to f(\mathbf{x}, \alpha)$ donde cada elección del parámetro α genera una "máquina de entrenamiento". Se supone que la máquina es determinista, esto es, para cada $\mathbf{x}$ dado, y un cierto α, la salida de $f(\mathbf{x}, \alpha)$ es siempre la misma. Podemos considerar que α se corresponde con los parámetros "libres" que determinan una máquina concreta; en el caso de una red neuronal con una arquitectura dada, serían los pesos y *bias* de la red, o en el de un algoritmo genético α comprenderá el número de nodos, ramas y funciones de los nodos que definen el árbol final.

El *error de entrenamiento* se puede definir como:

$$R_{emt}[\alpha] = \frac{1}{l}\sum_{i=1}^{l}\frac{1}{2}|f(\mathbf{x_i}, \alpha) - y_i| \tag{2.6}$$

El error de test (*riesgo*) esperado para una máquina de entrenamiento se establece, tal y como se veía en (1.2) mediante

$$R[\alpha] = \int \frac{1}{2}|f(\mathbf{x}, \alpha) - y| dP(\mathbf{x}, y) \tag{2.7}$$

La cantidad $\frac{1}{2}|f(\mathbf{x}_i,\alpha)-y_i|$ recibe el nombre de *pérdida*.

Nótese que, a diferencia del riesgo, $R_{emt}[\alpha]$ no depende de la distribución de la probabilidad; es fijo para cada α y cada conjunto de entrenamiento $\{\mathbf{x}_i, y_i\}$. Además, es interesante observar que un valor mínimo para el error de entrenamiento, no implica necesariamente un riesgo pequeño. De hecho, la teoría estadística del aprendizaje demuestra que hay que necesariamente restringir la clase de funciones *f* susceptibles de ser elegidas a aquellas que cuenten con la *capacidad* adecuada a la cantidad disponible de datos de entrenamiento. La teoría VC proporciona además cotas al error de test que nos podemos permitir. La capacidad de una máquina determina la complejidad de la máquina, de igual modo que el factor de riesgo viene a medir su calidad. Para lograr una capacidad de generalización alta, hay que alcanzar el equilibrio entre esos dos factores, y esa es precisamente la tarea de un problema de clasificación multi-atributo: encontrar la máquina que *aprenda* la aplicación $x_i \mapsto y_i$ con la habilidad de generalización más alta posible [Schölkopf 2002]

La anterior es la formalización para el problema de dos clases. La extensión natural para el de *K* clases, dependerá de la filosofía concreta de cada clasificador, pero lo más frecuente es construir funciones de decisión binarias para cada clase *k* del siguiente modo:

$$f_k: \mathbb{R}^n \rightarrow \{\pm 1\} \qquad \begin{cases} +1 & \text{para cada muestra en la clase } k \\ -1 & \text{para el resto} \end{cases} \tag{2.8}$$

Otro modo muy popular de abordar el problema de *K* clases, es el de la *clasificación a pares*, donde se trata de usar *K**(*K*-1)/2 clasificadores para cubrir todos los pares de clases, en vez de emplear sólo *K* clasificadores binarios con la filosofía de "una clase frente al resto". Concretaremos éstos y otros métodos más adelante.

2.5 Referencias de las aplicaciones en problemas actuariales

Durante los años 90, el interés por la aplicación a la Ciencia Actuarial de herramientas de aprendizaje tales como RNA o AG, ha ido en aumento [Shapiro, A. 2001]. Sin embargo, lo usual es encontrar aplicaciones de una técnica individual, en perjuicio de la utilización de los que ya hemos nombrado algoritmos híbridos, mucho más eficientes, por otra parte [Shapiro, A. 2002].

Encontrar problemas actuariales en los que se haya utilizado una SVM para su resolución no es tarea fácil. Entre las que hemos encontrado destacan dos, *Analysis of workplace accidents using machine learning techniques*, de J. M. Matías et al (2006), y *Prediction of insolvency in non-life insurance companies using Support Vector Machines, Genetic Algorithms*, *and Simulated Annealing* de Segovia-Vargas M. J., Salcedo-Sanz S, Bousoño-Calzón C.(2003).

En el primero, se analizan los tipos y causas de accidentes laborales empleando técnicas de aprendizaje máquina, concretamente, redes Bayesianas entrenadas con diferentes algoritmos con y sin información a priori, árboles de clasificación y SVM. En el segundo, que tiene su colofón en *Genetic programming for the prediction of insolvency in non-life insurance companies* (Sancho Salcedo-Sanz, José Luis Fernández-Villacañas, María Jesús Segovia-Vargas, Carlos Bousoño-Calzón:. Computers & OR 32: 749-765 (2005)), se utiliza una SVM híbrida con un Algoritmo Genético y una técnica de búsqueda global conocida como Simulated Annealing para predecir la insolvencia de compañías de seguros no vida. Muy en la línea de nuestra investigación, emplea la SVM para clasificar las empresas en solventes y no solventes, mientras que el AG y el SA realizan una selección *on-line* entre los ratios contables que describen a las empresas, con vistas a mejorar el funcionamiento de la SVM.

Los artículos relacionados con los seguros y basados en resoluciones vía RNA cubren desde problemas de clasificación, modelos de cash flow, de inversión, predicción de insolvencia, y estudios sobre mortalidad y morbilidad. El hecho de que este sea un nuevo campo de aplicación para las redes, hace que en la mayoría de las referencias encontremos estudios comparativos con los resultados obtenidos mediante las técnicas "clásicas".

La *clasificación* resulta fundamental para el negocio del seguro, como ya hemos observado. Por eso es importante emplear una herramienta eficaz para realizar el proceso. Aunque cada vez es mayor, la aplicación de las RNA en este campo se ha limitado a tres áreas: aseguración de bonos en la construcción, aseguración de clientes de seguros de vida, y la detección de fraude en indemnizaciones de seguros:

Bakheet, M. T. [*Contractors' Risk Assesment System*, 1995. Ph. D. Dissertation, Georgia Institute of Technology] utilizó una red neuronal con algoritmo de retropropagación como herramienta para la aseguración de bonos en la construcción. Las compañías aseguradoras en este sector basan su criterio en la evaluación de un conjunto de factores tales como el carácter, el capital, la capacidad y la continuidad. La conclusión era que el modelo planteado resultaba un método eficiente para manejar la evaluación.
Vaughn et. al (1997) se valieron de un perceptrón multicapa para clasificar los asegurados de una compañía de seguros de vida en riesgo estándar y no estándar. Entonces utilizaron un método de "descubrimiento del conocimiento" (similar al de extracción de características) para identificar los inputs relevantes que usa la red para clasificar a los asegurados. La clasificación de esas entradas permite saber el conocimiento adquirido por la red durante el entrenamiento para presentarlo en forma de relaciones entre los datos y reglas inducidas que demuestran que la red aprende de modo sensible y efectivo cuando se compara con el conjunto de datos de entrenamiento.

Brockett *et. al.* (1998) utiliza dos tipos de redes para su estudio, un *Mapa de Kohonen* para descubrir el fraude en las peticiones de indemnización por

heridas en los siniestros de automóvil, y una red entrenada con el algoritmo de retropropagación para validar la aproximación del Mapa. El objetivo del trabajo era determinar cuándo la reclamación era fraudulenta y el "nivel de sospecha" de fraude detectado en el archivo de demandas. La conclusión a la que llegaron fue que el método empleado mejoraba los utilizados hasta ahora, y se sugería la posibilidad de aplicarlo también a la detección de fraudes en las demandas de seguros médicos.

En el campo de los *modelos de inversión y activos* se pueden encontrar varias referencias a aplicaciones de las RNA, la mayoría recogidas en la obra de Refenes [Refenes 1995.]. En el mismo sentido está el artículo de Boero y Cavalli [*Forecasting the exchange Rage: A Comparison Between Econometric and Neural Network Models.* 1996. AFIR, Vol 2, 981]. en el que se investiga un modelo de RNA para predecir la tasa de cambio entre el Dólar estadounidense y la Peseta española. Comparan el resultado obtenido con cuatro modelos lineales, y se demuestra una clara ventaja en la aplicación del nuevo método cuando se le entrena con datos mensuales, aunque no es el caso cuando los datos son trimestrales.

También podríamos incluir en la anterior categoría los artículos de Tolmos, P relativos a la predicción del índice IBEX 35, y sus sectoriales, en diferentes momentos del tiempo. Las predicciones obtenidas eran esperanzadoras, sorprendentemente ajustadas en algunos casos, dado el carácter impredecible en esencia del Mercado de Valores, sujeto a fluctuaciones incontrolables por variados motivos.

Pero es en el ámbito de la *Insolvencia* donde encontramos un número mayor de aplicaciones. Centrándonos en los casos de compañías aseguradoras, podemos destacar los trabajos de Park [*Bankruptcy Prediction of Banks and Insurance Companies: An Approach Using Inductive Methods*. 1993. Ph. D. Dissertation. University of Texas at Austin], Brockett e*t. al* [Brockett e*t. al* 1994], Huang *et. al.* [*Life Insurer Financial Distress Prediction: A Neural Network Model.* 1995. Journal of Insurance Regulation, Winter, Vol. 13, No 2, pp. 131-167], Jang [*Comparative Analysis Of Statistical Methods And Neural Networks Fornpredicting Life Insurers' Insolvency (Bankruptcy)*. 1997. Ph. D. Dissertation. University of Texas at Austin], y [Tam KY. *Neural network models and the prediction of bankruptcy*. 1991; Omega 19(5):429–45.]

3. MÁQUINAS DE VECTORES SOPORTE. RESULTADOS

Aunque en el breve recorrido histórico que realizamos antes estudiamos el nacimiento de las Redes Neuronales Artificiales (NN), no dimos una definición del funcionamiento de las mismas, lo que haremos someramente a continuación. Esto es importante para conocer la evolución de las NN hacia las Máquinas de Vectores Soporte, y para comprender los rudimentos de estas últimas

3.1 Las Redes Neuronales Artificiales

La *Neurona Artificial* más básica puede modelarse mediante un mecanismo de entrada múltiple no lineal, con interconexiones ponderadas (las intensidades sinápticas[9]) w_{ij}.

El soma o cuerpo de la neurona biológica (donde se realizan casi todas las funciones lógicas de la célula) se representa por una restricción no lineal, la *función de activación* $\Psi(u_j)$.

El modelo más sencillo de una neurona artificial suma las n entradas (inputs) ponderados por los w_{ij}, y transforma el resultado a través de una no linealidad de acuerdo con la ecuación:

$$y_j = \Psi\left[\sum_{i=1}^{n} w_{ji}x_i + \theta_j\right]$$

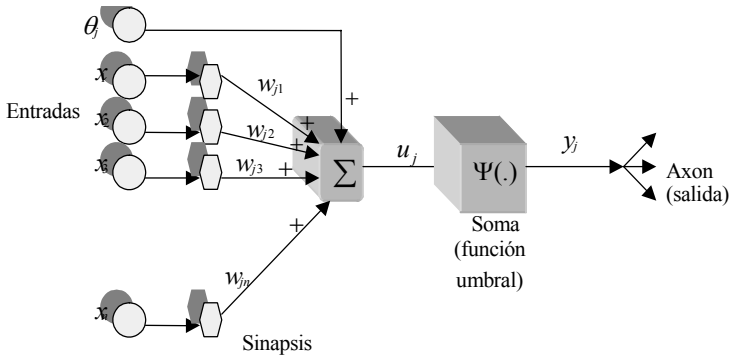
donde Ψ es la *función de activación* (o función umbral, o de transferencia), θ_j es el *umbral externo*, también llamado offset o bias, w_{ij} son los *pesos sinápticos*, x_i las *entradas* (consideramos un n º *n* de ellas) e y_j las *salidas* (output).

Esta ecuación se puede escribir también como:

$$y_j = \Psi\left[\sum_{i=0}^{n} w_{ji}x_i\right] \quad con\ w_{j0} = \theta_j, x_0 = 1$$

[9] La *sinapsis* es el contacto especializado entre una neurona y las demás. Las señales excitatorias e inhibitorias recibidas por una neurona se combinan, y en función de la estimulación total recibida, la neurona toma un cierto nivel de activación, que se traduce en la generación de breves impulsos nerviosos con una determinada frecuencia o tasa de disparo, y su propagación a lo largo del axon hacia las neuronas con las cuales sinapta.

Podríamos representarlo por las figura



La Red Neuronal más conocida, y de la que partieron la mayoría de los trabajos en este campo, es el *Perceptrón*. Su funcionamiento es básicamente el siguiente: Se realiza el producto de las señales de entrada $\{x_i\}$ por un conjunto de pesos sinápticos ajustables $\{w_{ji}\}$ para generar un potencial interno u_j, que es procesado por una función de tipo *signo*[10] , obteniéndose así una salida binaria $y_j \in \{-1,1\}$.

Este es comparado entonces con la señal binaria deseada $\hat{d}_j \in \{-1,1\}$, produciéndose un error cuantificado que se emplea para ajustar los pesos durante el proceso de aprendizaje.

La regla de aprendizaje original para ajustar los pesos del Perceptrón fue desarrollada por Rosenblatt::

$$w_{ji}^{(k+1)} = w_{ji}^{(k)} + \eta \hat{e}_j^{(k)} x_i^{(k)}$$

donde $w_{ji}^{(k)}$ es el valor de w_{ji} en el instante (de tiempo discreto) $t = k\tau$, τ es el periodo de la muestra, y

$$\hat{e}_j^{(k)} = \hat{d}_j^{(k)} - y_j^{(k)}; y_j^{(k)} = sign(u_j^{(k)}); u_j^{(k)} = \sum_{i=0}^{n} w_{ji}^{(k+1)} x_i^{(k)}; x_0 = 1$$

[10] Suele adoptar la forma $\varphi(u_j) = \begin{cases} 1 & si\ u_j \geq \theta_j \\ -1 & si\ u_j < \theta_j \end{cases}$

La tasa de aprendizaje η>0 controla la velocidad de convergencia del algoritmo

El Perceptrón opera básicamente como un mecanismo de clasificación de muestras, y en las aplicaciones modernas, la función *signo* se reemplaza por una función de activación no lineal tal como la sigmoidal donde el escalar positivo γ se usa para modificar la pendiente de la función.

$$y_j = \Psi(u_j) = tanh(\gamma u_j)$$

En esos casos la función más utilizada para minimizar es

$$E_2 = \frac{1}{2}(\hat{e}_j^{(k)})^2 = \frac{1}{2}(\hat{d}_j^{(k)} - y_j^{(k)})^2$$

que no es otra cosa que una estimación instantánea del error cuadrático medio.

El algoritmo que resulta es una forma standard del algoritmo de retropropagación (*back-propagation*) aplicado al *Perceptrón Multicapa*. El Perceptrón individual en la red recibe el nombre de "neurona" o *nodo*, y difiere del Perceptrón de Rosenblatt en que se usa la no linealidad sigmoidal en lugar del *signo*. El vector que contiene la entrada se introduce en cada Perceptrón de la primera capa, la salida de ésta se le da como entrada a la segunda capa de Perceptrones, y así seguidamente. A menudo los nodos están totalmente conectados entre capas, es decir, cada Perceptrón de la capa *l* está conectado con cada uno de la capa *l*+1. A veces el vector de entrada mismo se toma como una capa de la red, por lo que se suele especificar la arquitectura de la misma indicando el n º de capas "ocultas", esto es, el n º de capas que no son entrada ni salida.

3.2 La Máquina (red) de Vectores Soporte

La Máquina (red) de Vectores Soporte[11] es una nueva técnica de clasificación, que ha demostrado sobradamente su capacidad de resolución frente a problemas de elevado grado de complejidad. Diseñadas en principio para tratar problemas de clasificación binarios (en dos grupos), se trata de una máquina de aprendizaje que implementa la siguiente idea: cuando no sea posible separar los datos en el espacio de entrada con un hiperplano lineal, trasladar, mediante una aplicación no lineal, los vectores de entrada a un nuevo espacio de dimensión alta. En este nuevo espacio se construirá una superficie de decisión lineal. Las especiales propiedades que poseerá esta superficie garantizarán que la capacidad de generalización de la máquina de aprendizaje sea alta. Aunque esta idea se empleó en los primeros experimentos para datos que podían separase sin errores, se puede extender para el caso no separable con notable éxito. Pero antes de adentrarnos más en estas máquinas, veamos, volviendo atrás en la historia, cómo y de quién surgió este modelo.
El método de los *vectores* soporte fue descubierto a primeros de 1960 con la

[11] En inglés Support Vector Machine, cuyas siglas, SVM, se emplearán frecuentemente a lo largo del presente trabajo.

idea de aplicarlo en la construcción de hiperplanos de separación en los problemas de reconocimiento de patrones [Vapnik y Lerner, 1963]; [Vapnik y Cherbonenkis, 1962]. Entre 1992 y 1995 se generalizó para construir funciones de separación no lineales (pero que sí lo fueran en el *espacio de características*[12]) [Boser et al., 1992]; [Cortes y Vapnik, 1995]. Ya en 1995 se amplió para estimar funciones con valores reales [Vapnik, 1995]. Por último, en 1996, se aplicó para resolver ecuaciones con operadores lineales [Vapnik et al. 1997]. Sin embargo, la idea que subyace, la de tratar de procurar la linealidad, tuvo su origen y su continuidad en otros fueros. Hace casi 70 años, R.A. Fisher [Fisher 1936] fue el promotor del primer algoritmo para el reconocimiento de patrones. Diseñó un modelo para dos poblaciones con distribuciones normales, y demostró que, aunque la solución óptima (en sentido Bayesiano) era una función de decisión cuadrática, lo más sencillo era emplear una función discriminante lineal con una matriz de covarianzas que definió[13]. Es por ello que los algoritmos de aprendizaje han estado ligados, casi desde su origen, con la construcción de superficies de decisión *lineales*.

En 1962, como vimos antes, Rosenblatt dirigió sus investigaciones hacia un nuevo tipo de máquinas de aprendizaje: los perceptrones o redes neuronales. Puede considerarse que el perceptrón implementa una superficie de separación lineal definida a trozos, puesto que cada neurona calcula su hiperplano de separación. De ahí su conexión con las SVM.

No fue hasta 1986 cuando se halló [Rumelhart, Hinton y Williams, 1986, 1987]; [Parker, 1985]; [LeCun, 1985] un algoritmo que permitía adaptar todos los pesos de la red neuronal para minimizar el error cometido sobre una serie de vectores de un problema de reconocimiento de patrones: el famoso algoritmo de retropropagación. Como ya vimos, la solución implicaba una leve modificación en el modelo matemático para las neuronas. De este modo, las redes neuronales implementan funciones de decisión de tipo *lineal* definidas a trozos. Así, la línea de razonamiento en la construcción de las SVM está clara: en el caso de que los datos no se puedan separar mediante un hiperplano lineal (casi siempre), éstos se trasladan a un espacio de dimensión alta donde se construye una superficie de decisión *lineal*, cuyas especiales propiedades asegurarán la elevada capacidad de generalización de la red.

Sin embargo, nos vamos a encontrar con dos importantes, aunque no irresolubles, escollos en nuestro camino hacia la linealidad. Uno es de carácter conceptual, y otro técnico. El primero es cómo encontrar un hiperplano de separación que generalice bien, puesto que la dimensión del espacio de características puede ser enorme, y no todos los hiperplanos que separen los datos de entrenamiento necesariamente han de mantener la capacidad de generalización que deseamos. El problema técnico hace referencia a cómo

[12] Más adelante se dará una definición adecuada de esta espacio. Por el momento, baste decir que es un nuevo espacio al que se trasladarán los vectores de entrada con el fin de poder definir adecuadamente ciertas propiedades.

[13] Vapnik, V., Cortes, C. Support-Vector Networks. 1995. Machine Learning, 20, 273-297.

manejar computacionalmente esos espacios de alta dimensión (nótese que para construir un polinomio de grado 2 o 5 en un espacio de dimensión 200 sería necesario generar hiperplanos en un espacio de características de dimensión un billón).

La parte conceptual del problema la resolvió Vapnik en 1965 para el caso de *hiperplanos óptimos* para clases separables, dando origen, como comentábamos antes, a la Teoría de Vectores Soporte. En este contexto, Vapnik definió un hiperplano óptimo como una función de decisión lineal con el margen de separación máximo entre los vectores de las dos clases. Se observó entonces que para construir tal hiperplano, uno sólo debía tener en cuenta una cantidad pequeña de los datos de entrenamiento, los llamados *vectores soporte*, quienes determinaban ese *margen*. De este modo, se demostró que si se separaban los vectores de entrenamiento sin cometer errores mediante un hiperplano óptimo con tales características, entonces el valor esperado de la probabilidad de cometer un error con un ejemplo de test está acotado por el ratio entre el valor esperado del número de vectores soporte, y el número de vectores de entrenamiento. De esta cota se deduce que si es posible construir el hiperplano óptimo empleando un número bajo de vectores soporte relativos al tamaño del conjunto de entrenamiento, entonces la capacidad de generalización será alta, incluso en un espacio de dimensión infinita. Pese a todo, incluso aunque el hiperplano óptimo generalice bien, permanece el problema técnico de cómo manejar los espacios de características de dimensión alta. Fue en 1992 cuando se probó ([Boser, Guyon y Vapnik, 1992]) que era posible intercambiar el orden de las operaciones para construir la función de decisión; en vez de realizar una transformación no lineal de los vectores de entrada seguida de ejecutar los productos escalares de los vectores soporte en el espacio de características, se puede comparar primero dos vectores en el espacio de entrada (ya sea tomando su producto escalar u otra medida cualquiera de la distancia), y realizar entonces la transformación lineal sobre el valor del resultado. Esto permite la construcción de una clase rica de superficies de decisión, por ejemplo polinomiales de grado arbitrario. A este tipo de máquina de aprendizaje es a la que se conoce con el nombre de *Red de Vectores Soporte*, enfatizando así lo crucial para estas máquinas de la idea de expandir la solución sobre esa clase de vectores (en el algoritmo de aprendizaje de los vectores soporte la complejidad de la construcción no depende de la dimensión del espacio muestral, sino del número de vectores soporte).

El objetivo entonces está claro. Veamos en los siguiente puntos cómo se realiza la construcción de las Máquinas de Vectores Soporte, comenzando por establecer una representación adecuada de los datos, y la función que represente la similaridad entre los mismos en el espacio inicial de datos, para luego definir el hiperplano óptimo y los vectores soporte, y finalmente, extender todo ello al caso no separable. Todo ello lo haremos para el caso de separación de los datos en dos clases, y en el último apartado extenderemos los resultados para el caso multi-clase.

3.3 Representación de los datos y Similaridad

Uno de los problemas fundamentales que se resuelven dentro de la Teoría del Aprendizaje, como hemos visto, es el de la clasificación, que de hecho, es el que nos ocupa en el presente trabajo. En su caso más "simple", se enunciaría del siguiente modo: dadas dos clases de objetos, ¿a cuál de las dos asignamos el nuevo objeto que se nos presente? La formalización matemática de la cuestión es sencilla:

Consideramos los datos (o patrones) obtenidos de forma empírica

$$(x_1, y_1),...,(x_l, y_l) \in \ X \times \{\pm 1\} \tag{3.1}$$

El conjunto X es el conjunto no vacío al que pertenecen las *observaciones* x_i (también llamadas *entradas*, *casos*, *ejemplos* o *patrones*), y al que nos referiremos como *dominio* o *espacio muestral*. Las *salidas* (*etiquetas*, *objetivos*, o a veces, también *observaciones*) vamos a tomarlas como +1, si la entrada pertenece a una clase determinada, y -1 si se encuentra en a la otra[14]. Las entradas pueden ser de cualquier tipo, y sobre el conjunto X no se establece ninguna restricción, salvo la de ser un conjunto.

En nuestro problema de clasificación de los asegurados, los conjuntos $\{\mathbf{x}_i\}, i \in \{1,...l\}$ representarán a los individuos ($l = 58238$ en el experimento de 2003, por ejemplo) descritos por el conjunto de factores de riesgo (cada componente de $\mathbf{x}_i$, $\mathbf{x}_{ij}$, es un factor) que ya vimos en el capítulo anterior, y las etiquetas $y_i \in \{-1,1\}$ indicarían la clase, -1 si no tendrán siniestros, y 1 en caso contrario.

El número de componentes de cada vector variará en cada una de las aplicaciones que hagamos, dependiendo de si utilizamos todos los factores de riesgo, o realizamos primero la selección de factores.

El objetivo es que se clasifiquen correctamente los ejemplos nuevos que se presenten $(\mathbf{x}, y)$ generados por la misma distribución de probabilidad *"subyacente"* $P(\mathbf{x}, y)$ que los datos utilizados para el entrenamiento del clasificador.

Esta labor, como decimos, se puede realizar tanto seleccionando los factores de riesgo como no, y nosotros lo haremos de las dos formas, para comparar los resultados. Sin embargo, para realizar un estudio correcto del problema de aprendizaje, hay que añadir un nuevo tipo de estructura. La razón es que, en un problema de aprendizaje, nuestra intención será la de *generalizar* para los datos que aún no se han presentado.

[14] Recuérdese que estamos considerando el problema binario, de dos clases. Más adelante se generalizará para el multiclase, rango en el que se encuentra nuestro problema.

En el caso del reconocimiento de patrones, por ejemplo, significa que, dado una nueva muestra $x \in X$, queremos predecir qué etiqueta $y \in \{\pm 1\}$le corresponderá. En este sentido, se trata de elegir *y* de modo que (*x*, *y*) sea *similar* de algún modo a los ejemplos de entrenamiento (3.1).

Esta es la razón (que además posibilita, como comentábamos antes, el resolver el problema técnico sobre la dimensión, al permitir intercambiar las operaciones para la construcción de la función de decisión), por la que se hace imprescindible una definición de *similaridad*, tanto en *X* como en $\{\pm 1\}$.

Cuando se trata de las salidas, es sencillo caracterizar la similaridad: en la clasificación binaria, sólo pueden darse dos situaciones, es decir, dos etiquetas sólo son idénticas o diferentes. No es así para los datos de entrada, donde la elección de una medida adecuada para la similaridad es una cuestión que subyace en el corazón del campo del aprendizaje máquina.

Definamos una medida de la similaridad de la forma:

$$\begin{aligned} k &: X \times X \to \mathbb{R} \\ (x, x') &\mapsto k(x, x') \end{aligned} \tag{3.2}$$

Esta función transforma dos entradas *x* y *x*' en un número real que caracteriza su similaridad. Supondremos que se trata de una función simétrica, y la llamaremos *kernel* No se trata de un nombre casual: deriva del primer uso que se hizo de este tipo de función (núcleo) por parte de Hilbert (y otros) en el campo de los operadores integrales.

El kernel más sencillo por su operatividad, pero que cumple su objetivo en cuanto a reflejar bien la similaridad entre los datos, es el producto escalar (euclídeo), esto es,

$$\langle \mathbf{x}, \mathbf{x}' \rangle = \sum_{i=1}^{N} x_i \cdot x'_i \tag{3.3}$$

La interpretación geométrica del producto escalar, que da lugar a las nociones de ángulo comprendido entre dos vectores, norma (módulo o *longitud*) de un vector, y distancia entre dos vectores, da una idea de cómo caracterizar la similaridad entre dos entradas. Sin embargo, y pese a sus ventajas, el uso de esta función como kernel tiene dos importantes inconvenientes:

Habíamos asumido que los datos de nuestro problema podían ser de cualquier tipo, de modo que puede que ni siquiera estén representados como vectores, o que, aún en ese caso, ni pertenezcan a un espacio dotado de producto escalar.

Aunque se trate de una medida que refleje bien la similaridad entre dos muestras dadas, sigue siendo demasiado sencilla, y, sobre todo, lineal, lo que

impedirá, en muchos casos, representar adecuadamente la semejanza entre dos datos.

La solución consiste en el uso de una aplicación que nos permita representar los datos como vectores en un nuevo espacio H en el que sí esté definido un producto escalar:

$$\begin{aligned} \Phi &: X \to \mathrm{H} \\ x &\mapsto \mathbf{x} := \Phi(x) \end{aligned} \tag{3.4}$$

H recibe el nombre, que ya empleamos en el punto dedicado a la introducción, de *espacio de características* o espacio kernel. La aplicación Φ traslada el espacio en el que se presentan los datos del problema, al nuevo espacio, en el que las muestras aparecerán representadas como vectores.

De este modo, la medida de similaridad quedaría bien definida:

$$k(x,x') := \langle \mathbf{x}, \mathbf{x}' \rangle = \langle \Phi(x), \Phi(x') \rangle \tag{3.5}$$

Además, la definición del espacio H va a permitir manejar las muestras geométricamente, pudiendo emplearse técnicas de álgebra lineal y geometría analítica para el estudio de los algoritmos de aprendizaje.

Por último, la libertad para elegir la aplicación Φ más adecuada a nuestras necesidades, derivará en una gran variedad de medidas de similaridad y algoritmos de aprendizaje. Todo ello es aplicable, por supuesto, al caso en el que el espacio muestral esté ya dotado de producto escalar. En ese caso, se utilizará el producto allá definido como medida de similaridad, si se quiere, pero nada nos impedirá realizar la transformación vía una Φ no lineal, si ello es más aconsejable para nuestros propósitos.

Resaltar, para concluir este punto, las funciones kernel más empleadas en la práctica; son el polinomial, el Gaussiano, y el sigmoide, respectivamente:

$$\begin{aligned} k(x,x') &= \langle x, x' \rangle^d \\ k(x,x') &= \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \\ k(x,x') &= \tanh(\kappa \langle x, x' \rangle + \Theta) \end{aligned} \tag{3.6}$$

Estas funciones, con las elecciones adecuadas de $d \in \mathbb{N}$ y $\sigma, \kappa, \Theta \in \mathbb{R}$ (suponiendo $X \subset ?^N$) conducen a clasificadores de vectores soporte con elevada capacidad de resolución.

3.4 Clasificación mediante hiperplanos óptimos

En esta sección nos adentraremos en el método de los hiperplanos óptimos del que dimos unas pinceladas en la introducción dedicada a las SVM. Se trata, como comentábamos entonces, de separar los datos de entrenamiento "sin errores". Apelando también en lo que establecimos en el punto **1** de este Capítulo, relativo al problema de clasificación, vamos a comenzar por el caso de separación en dos clases de datos *separables*, cuya extensión al de varias clases es teórica, aunque no técnicamente, sencilla.

Sean los datos tal y como los consideramos en (3.1),

$$(\mathbf{x}_1, y_1),...,(\mathbf{x}_l, y_l) \quad y_i \in \{-1,1\} \quad [15] \tag{3.7}$$

Como ya se dijo en las notas relativas a la representación matemática de un problema de clasificación, para diseñar algoritmos de aprendizaje cuya efectividad, en el sentido estadístico, pueda controlarse, se necesita trabajar con funciones en las que su capacidad sea computable. En este sentido, Vapnik y Chervonenkis ([Vapnik y Chervonenkis 1974]), consideró la clase de hiperplanos en el espacio H,

$$\mathbf{w}^t \cdot \mathbf{x} + b = 0 \qquad \text{donde } \mathbf{w} \in \mathrm{H},\ b \in \mathbb{R} \tag{3.8}$$

En este tipo de formulación, **w** es un vector ortogonal al hiperplano. El conjunto (3.8) se corresponde con los vectores que tienen la misma longitud sobre **w**.

Diremos que el par (**w**,*b*) es la *forma canónica del hiperplano* (3.8) respecto a $\{\mathbf{x}_i\} \in \mathrm{H},\ i = 1,...,l$ si se escala de modo que

$$\min_{i=1,...,m} \left|\mathbf{w}^t \cdot \mathbf{x}_i + b\right| = 1 \tag{3.9}$$

Esta condición da lugar a dos hiperplanos, (**w**, *b*) y (-**w**, -*b*), que en el ámbito de los problemas de clasificación, son diferentes, ya que están orientados de distinto modo; se corresponden con dos funciones de decisión, que son la inversa una de la otra:

$$\begin{aligned} f_{\mathbf{w},b} &: \mathrm{H} \to \{\pm 1\} \\ \mathbf{x} &\mapsto f_{\mathbf{w},b}(\mathbf{x}) = sign\left(\mathbf{w}^t \cdot \mathbf{x} + b\right) \end{aligned} \tag{3.10}$$

En ausencia de las etiquetas asociadas a cada clase $y_i \in \{-1,1\}$ correspondientes a los $\mathbf{x}_i$, no habría modo de distinguir los dos hiperplanos. No es el caso de los conjuntos de datos ya "etiquetados": estos hiperplanos realizan asignaciones de clases opuestas. En el tipo de problema que tratamos, este planteamiento conduce a encontrar una solución $f_{w,b}$ que clasifique

[15] Utilizamos ya la notación de vector del espacio de características.

correctamente los ejemplos etiquetados $(\mathbf{x}_i, y_i) \in \mathrm{H} \times \{\pm 1\}$; en otras palabras, que satisfaga $f_{w,b}(\mathbf{x}_i) = y_i$ para todo *i* (diremos entonces que se trata de un conjunto *separable*) o, al menos, para una gran parte de ellos.

De este modo, definimos los patrones (3.7) como *linealmente separables* si existen tales **w** y *b* de manera que se satisfagan, para todos los elementos del conjunto de entrenamiento, las inecuaciones:

$$\mathbf{w}^t \cdot \mathbf{x}_i + b \geq 1 \quad \text{si} \quad y_i = 1$$
$$\mathbf{w}^t \cdot \mathbf{x}_i + b \leq 1 \quad \text{si} \quad y_i = -1 \tag{3.11}$$

El algoritmo diseñado por Vapnik se fundamenta en dos hechos: primero, existe un único hiperplano óptimo, que se caracteriza por que presenta el máximo margen de separación entre cualquier punto del conjunto de entrenamiento y él mismo. Concreta-mente, es la solución de calcular el máximo para $\mathbf{w} \in \mathrm{H}$, $b \in \mathbb{R}$ del

$$\min\left\{\left\|\mathbf{x} - \mathbf{x}_\mathbf{i}\right\| : \mathbf{x} \in \mathrm{H}, \mathbf{w} \cdot \mathbf{x} + b = 0,\ i = 1,...,l\right\} \tag{3.12}$$

Segundo, la *capacidad* de la clase de hiperplanos de separación decrece a medida que se incrementa el margen. Es por ello que existen argumentos teóricos que respaldan el buen comportamiento en generalización que posee el hiperplano óptimo. Además cuenta con la propiedad de ser computacionalmente atractivo, ya que, como se verá después, se puede construir resolviendo un problema de programación cuadrática, para los que hay numerosos algoritmos enormemente eficientes.

El hiperplano óptimo determina la dirección $\mathbf{w}/\|\mathbf{w}\|$ donde la distancia entre las proyecciones de vectores de entrenamiento de dos clases diferentes es máxima. Esto nos va a llevar al concepto de ***margen***, que denota la distancia al hiperplano de separación del punto más cercano al mismo. Veámoslo en el siguiente ejemplo [Schölkopf y Smola 2002]:

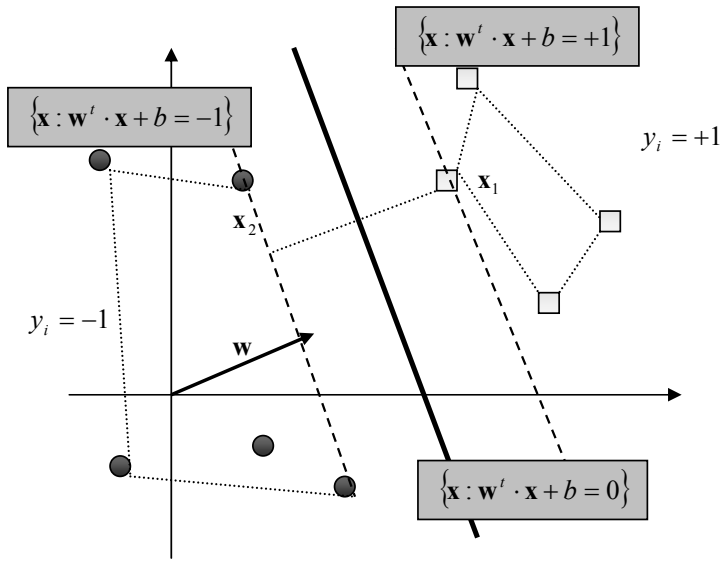


Figura 3.1

En la anterior figura consideramos un caso sencillo que consiste en separar los círculos de los cuadros. Representamos el hiperplano óptimo (3.8) por una línea gruesa. Al tratarse de un problema separable, existen el vector w y el umbral b tales que $y_i\left(\mathbf{w}^t \cdot \mathbf{x}_i + b\right) > 0$ para i= 1,…,l. Si procedemos a reescalar w y b de modo que los puntos más cercanos al hiperplano ($\mathbf{x}_1$ y $\mathbf{x}_2$) verifiquen $\left|\mathbf{w}^t \cdot \mathbf{x}_i + b\right| = 1$, obtendríamos la forma canónica del hiperplano, satisfaciéndose $y_i\left(\mathbf{w}^t \cdot \mathbf{x}_i + b\right) \geq 1$. Hay que observar que en este caso, el margen es $1/\|\mathbf{w}\|$: si tomamos los puntos $\mathbf{x}_1$ y $\mathbf{x}_2$, pertenecientes a lados opuestos del hiperplano, que precisamente verifican $\mathbf{w}^t \cdot \mathbf{x}_1 + b = 1$ y $\mathbf{w}^t \cdot \mathbf{x}_2 + b = -1$, y los proyectamos sobre el vector normal al hiperplano $\mathbf{w}/\|\mathbf{w}\|$, calculando por último la distancia entre ambas proyecciones, tenemos el resultado deseado:

.
$$\left|\frac{\mathbf{x}_1 \cdot \mathbf{w}}{\|\mathbf{w}\|} - \frac{\mathbf{x}_2 \cdot \mathbf{w}}{\|\mathbf{w}\|}\right| = \left|\frac{1-b}{\|\mathbf{w}\|} - \frac{-1-b}{\|\mathbf{w}\|}\right| = \frac{2}{\|\mathbf{w}\|}$$

Este ejemplo clarifica un hecho que va a resultar fundamental: podemos conseguir un margen de separación grande conservando $\|\mathbf{w}\|$ pequeño.

De modo intuitivo, la explicación es la siguiente: la distancia de $\mathbf{x}_i$ al hiperplano se mide como $\frac{\left|\mathbf{w}^t \cdot \mathbf{x}_i + b\right|}{\|\mathbf{w}\|}$,

o lo que es lo mismo, cuanto menor sea $\|\mathbf{w}\|$, mayor será esa distancia o margen *global,* que, como se aprecia en el ejemplo, para los puntos más cercanos al hiperplano, pertenecientes a los hiperplanos $\left|\mathbf{w}^t \cdot \mathbf{x}_i + b\right| = 1$ es, obviamente, el margen $1/\|\mathbf{w}\|$.

Veamos ahora una definición formal de margen. Llamaremos ***margen geométrico*** del punto $(\mathbf{x}, y) \in \mathrm{H} \times \{\pm 1\}$ a:

$$\rho(\mathbf{x}, y) = \frac{y\left(\mathbf{w}^t \cdot \mathbf{x} + b\right)}{\|\mathbf{w}\|} \tag{3.13}$$

que para el conjunto $(\mathbf{x}_1, y_1),...,(\mathbf{x}_l, y_l)$ sería

$$\rho_{(\mathbf{w},b)} = \min_{i=1,...,l} \frac{y_i\left(\mathbf{w}^t \cdot \mathbf{x}_i + b\right)}{\|\mathbf{w}\|} \tag{3.14}$$

Obsérvese cómo, para un punto bien clasificado, el margen es, simplemente, la distancia de **x** al hiperplano (el producto por *y* asegura que el margen sea positivo). Para un punto mal clasificado, obtendríamos la distancia *negativa* al hiperplano. Por último, volvemos a insistir en el hecho de que para los hiperplanos canónicos, el margen es $1/\|\mathbf{w}\|$.

De este modo, queda patente que la longitud de **w** se corresponde con una cantidad con sentido geométrico, que va a jugar un importante papel en el algoritmo de las SVM.

Pero, ¿por qué debemos buscar aquel hiperplano con el mayor margen posible? Sencillamente, porque si somos capaces de separar los datos de entrenamiento con un margen grande, también lo seremos con los puntos del conjunto de test. La idea es la siguiente: lo habitual es que los datos de test se encuentren cerca (en H) de al menos uno de los de entrenamiento.

Supongamos, por ejemplo, que generamos los datos de test añadiendo ruido a los de entrenamiento (práctica muy habitual, por otra parte), y que ese ruido está acotado en norma por un cierto *r*>0.Es obvio que, si somos capaces de separar los datos de entrenamiento con un margen $\rho > r$ podremos clasificar correctamente todos los elementos del conjunto de test. Además, si conociéramos ρ de antemano, podríamos diseñar el algoritmo de clasificación óptimo, pues podríamos aproximar *r* de manera que incluso los datos con ruido se pudieran separar con un margen no cero.

Por último, una ventaja añadida de los clasificadores con un margen alto, es que se comportan bien a la hora de su implementación en hardware, debido a su insensibilidad frente a pequeños cambios en los inputs.

Recapitulemos ahora para diseñar, al fin, el algoritmo que conduce al hiperplano óptimo. Buscamos una función de decisión $f_{\mathbf{w},b}(\mathbf{x}) = sign\left(\mathbf{w}^t \cdot \mathbf{x} + b\right)$ que cumpla que $f_{w,b}\left(\mathbf{x}_i\right) = y_i$. Si existe tal función, la forma canónica (3.9) implica que $y_i\left(\mathbf{w}^t \cdot \mathbf{x}_i + b\right) \geq 1$.

Con el objetivo en mente de lograr el máximo margen de separación, tenemos que, según lo comentado anteriormente, el hiperplano de separación que generalizará bien se puede construir resolviendo el siguiente problema de optimización:

Encontrar $\mathbf{w} \in \mathrm{H}$ y $b \in \mathbb{R}$ que

minimicen $\tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$ (3.15)

sujeto a $y_i\left(\mathbf{w}^t \cdot \mathbf{x}_i + b\right) \geq 1 \qquad \forall\, i = 1,...,l$" (3.16)

Este es el que llamaremos *problema primal*. La función τ de (3.15) es la *función objetivo* y las inecuaciones de (3.16) son las *restricciones de desigualdad*; juntos constituyen un *problema de optimización restringida*. No vamos a entrar a tratar este tipo de cuestiones con detenimiento en este trabajo. Baste decir que en la práctica, normalmente su resolución pasa por transformarlo en el problema *dual*, introduciendo unas nuevas variables conocidas como *multiplicadores de Lagrange*, $\alpha_i \geq 0$, y una nueva función a optimizar, llamada *función de Lagrange* o, simplemente, *Lagrangiana*:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left(y_i\left(\mathbf{w}^t \cdot \mathbf{x}_i + b\right) - 1\right) \tag{3.17}$$

Según el Teorema de Karush-Kuhn-Tucker[16] (KKT) esta función deberá ser maximizada con respecto a los multiplicadores, y minimizada para **w** y *b*. A consecuencia de esto, en ese punto crítico, deben anularse las derivadas de *L* con respecto a las variables primales, lo que nos lleva a :

[16] Teorema de Karush-Kuhn-Tucker para problemas diferenciables convexos: *El punto **x** es solución del problema [Min f(x) s.a.* $c_i(x) \leq 0 \quad \forall i = 1,...,n$ *] con f y* c_i *convexas y diferenciables, si existe algún* $\boldsymbol{\alpha} \in \square^{\mathrm{n}}$ *con* $\alpha_i \geq 0 \quad \forall i = 1,...,n$ *tal que se satisfacen las siguientes condiciones:*

$\partial_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\alpha}) = 0$

$\partial_{\alpha_i} L(\mathbf{x}, \boldsymbol{\alpha}) = c_i(\mathbf{x}) \leq 0$

$\sum_{i=1}^{n} \alpha_i c_i(\mathbf{x}) = 0$

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0, \quad \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \tag{3.18}$$

Añadiendo además unas condiciones complementarias de la teoría de optimización (restricciones de Karush-Kuhn-Tucker), se resuelve el problema, obteniéndose las ecuaciones:

$$\begin{aligned} &\sum_{i=1}^{l} \alpha_i y_i = 0 \\ &\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \end{aligned} \tag{3.19}$$[17]

Obsérvese que aunque la solución **w** es única (debido a la convexidad estricta de la función objetivo, y a la convexidad de (3.16)), los coeficientes α_i no tienen por qué.

De acuerdo con el Teorema de Karush-Kuhn-Tucker sólo los multiplicadores que no son cero en el punto crítico se corresponden con restricciones (3.16) que se alcanzan. Formalmente, se tiene:

$$\alpha_i \left(\left(y_i \left(\mathbf{w}^t \cdot \mathbf{x}_i + b \right) - 1 \right) \right) = 0 \tag{3.20}$$

Así, la expresión para el vector solución $\mathbf{w}$ es función de un subconjunto de los vectores de entrada, precisamente de aquellos cuyo multiplicador α_i no se anula: son los que llamaremos *vectores soporte*[18]. De acuerdo con (3.20), se encuentran exactamente en el margen. Son por ello los elementos "críticos" del conjunto de datos de entrenamiento, puesto que son los más cercanos a la frontera de decisión; si los eliminásemos, la solución variaría, pero si lo hiciéramos con otros (los que tienen $\alpha_i=0$, razón por la que no aparecen en la expansión (3.19)) y se repitiera el proceso de entrenamiento, se encontraría el mismo hiperplano de separación. Esto nos conduce directamente al cálculo de una cota superior para la capacidad de generalización de los hiperplanos de margen óptimo:

---

[17] El problema dual se obtiene sustituyendo estas expresiones en la Lagrangiana, y tiene la forma del siguiente problema de programación cuadrático:

$$MaxW(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j \right) \text{ para } \boldsymbol{\alpha} \in \square^l$$

sujeto a $\alpha_i \geq 0, i = 1,...,l$ y $\sum_{i=1}^{l} \alpha_i y_i = 0$

[18] El término se corresponde con nociones relacionadas con la teoría de conjuntos convexos, y tiene incluso un significado físico, en cuanto a que son los que ejercen (o soportan) la fuerza sobre un plano sólido que está en el hiperplano para lograr que la solución sea mecánicamente estable.

Proposición ([Schölkopf y Smola 2002]): el valor esperado del número de vectores soporte obtenidos durante el entrenamiento con un conjunto de tamaño *l*, dividido por *l*, es una cota superior de la esperanza de la probabilidad del error de test de la SVM entrenada con conjuntos de tamaño *l*-1.

Finalmente, utilizando la expresión para **w** obtenida en (3.19), tenemos que la función de decisión es de la forma:

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{l} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b\right) \tag{3.21}$$

que, como se puede apreciar, se puede evaluar sencillamente efectuando los productos escalares entre el vector a clasificar y los Vectores Soporte.

Tenemos ahora todas las herramientas necesarias para tratar el problema de la clasificación mediante las Máquinas de Vectores Soporte, lo que haremos en la siguiente sección.

3.5 Clasificación por vectores soporte

Hasta el momento se ha indicado por qué es bueno buscar hiperplanos de separación con un gran margen, e incluso hemos llegado a ver cómo calcularlo. Sin embargo, existe una pega en nuestro argumento anterior: todo lo que hemos hecho hasta aquí es lineal.

Por ello, en alas de lograr superficies de decisión lo más generales posibles, vamos a emplear las funciones *kernel* que definimos en el punto 3.2 para trasladar los puntos $x_1,...,x_l \in X$ al espacio de dimensión superior H que ya definimos entonces empleando la función:

$$\begin{aligned} \Phi : X &\to \mathrm{H} \\ x_i &\mapsto \mathbf{x}_i := \Phi(x_i) \end{aligned}$$

que teníamos en la fórmula (3.4). Cuando nos encontremos en el espacio de características, efectuaremos la separación lineal.
El Teorema de Cover [Scholkopf (2002)] se alude a veces para justificar este procedimiento.

Tal teorema caracteriza el número de separaciones lineales posibles para *l* puntos que se encuentren en una posición general en un espacio de dimensión *N*. Si $m \leq N+1$, entonces todas las 2^m separaciones lineales son posibles –la dimensión VC de de la clase de la función es *n* + 1. Si $m > N+1$, el Teorema de Cover establece que el número de separaciones lineales es:

$$2\sum_{i=0}^{N}\binom{m-1}{i}$$

Cuanto más incrementemos *N*, más términos tendremos en el sumatorio, y mayor será el número resultante.

Este Teorema formaliza lo que intuitivamente es evidente: que el número de separaciones aumenta con la dimensión del espacio en que se hallen los puntos. Requiere, no obstante, que los puntos estén en una posición general, por lo que no establece realmente un resultado de separabilidad para datos de un conjunto dado en un espacio dado.

Existe otro modo de entender intuitivamente porque la aplicación kernel aumenta las posibilidades de separación, en términos de conceptos de la teoría estadística del aprendizaje. Valernos de un kernel equivale a utilizar una clase de funciones mayor, aumentando por tanto la capacidad de la máquina de aprendizaje, interpretando como separables problemas que no lo son para poder comenzar con ellos.

Valiéndonos de la función Φ, y utilizando un kernel definido positivo, como los que concretamos en el punto 2, tenemos que:

$$\langle \Phi(x), \Phi(x_i) \rangle = k(x, x_i) \tag{3.22}$$

que nos lleva a funciones de decisión de la forma,

$$f(x) = sign\left(\sum_{i=1}^{l} \alpha_i y_i k(x, x_i) + b \right) \tag{3.23}$$

Obsérvese (figura 3.2) que de este modo, todo lo dicho en el punto anterior sería válido para el caso general, pues mediante esta transformación nos encontramos en un espacio de alta dimensión en el que hemos podido evaluar el producto escalar necesario (rehecho, esto es lo único fundamental en las definiciones del punto anterior) para considerar las funciones de decisión (3.21). Simplemente, donde antes decíamos $\mathbf{x}$, ahora queremos decir $\Phi(x)$, y cuando utilizábamos el producto escalar ahora nos valemos de un kernel adecuado.

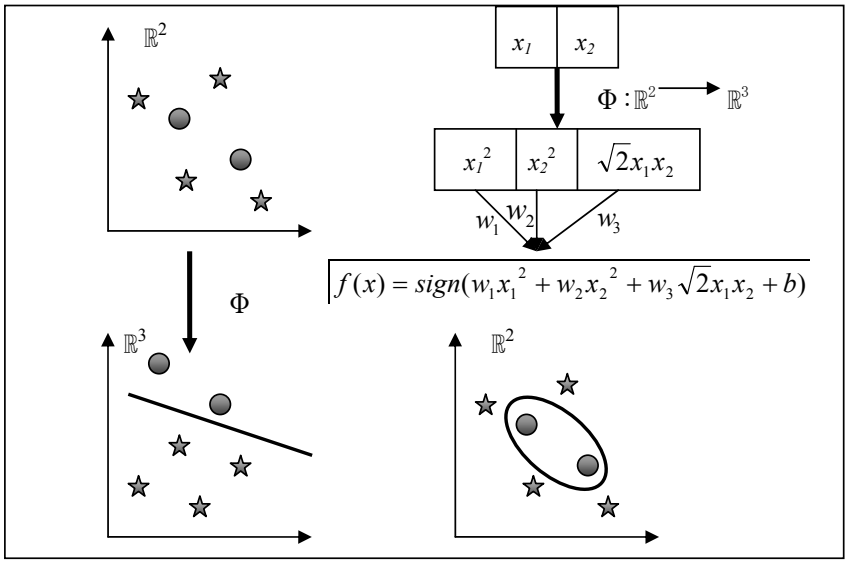


Figura 3.2

La arquitectura de las SVM sería la correspondiente al siguiente esquema:

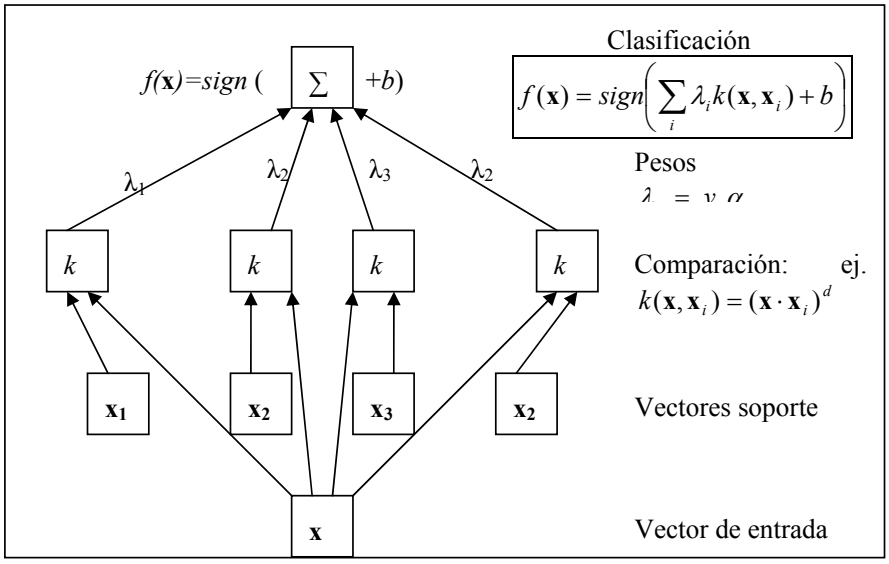


Figura 3.3

Se escoge a priori una función kernel (lo único que se le exige, para asegurar la convexidad del programa cuadrático, es ser definida positiva *condicionalmente*[19]); ésta determinará el tipo de clasificador.

[19] Sea *X* conjunto no vacío, y *K* el cuerpo ? o ?. Se dice que el kernel $k : X \times X \rightarrow \mathrm{K}$ es

El resto de los parámetros (número de unidades ocultas, pesos, umbral *b*) se calculan durante el proceso de entrenamiento resolviendo el problema de programación cuadrática. Como sabemos, los pesos de la primera capa son un subconjunto de los vectores del conjunto de entrenamiento (los vectores soporte), y los de la segunda se conocen a partir de los multiplicadores de Lagrange, $\lambda_i = y_i \alpha_i$.

Si empleamos las funciones kernel que describimos en el punto 2, el algoritmo SVM conduce a una diversidad de máquinas de aprendizaje, algunas de las cuales coinciden con arquitecturas clásicas:

$k(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x} \cdot \mathbf{x}_i)^d$: *Clasificadores polinomiales de grado d*

$k(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\|\mathbf{x} - \mathbf{x}_i\|^2 / c\right)$: *Clasificadores de función de base radial ,* con kernel Gaussiano de tamaño c>0.

$k(\mathbf{x}, \mathbf{x}_i) = \tanh(\gamma(\mathbf{x} \cdot \mathbf{x}_i) + \theta)$: *Redes Neuronales*, con función de activación tangente hiperbólica (los parámetros γ y θ son tal y como los definimos en el apartado correspondiente a las redes neuronales).

Para calcular el umbral *b*, basta con tener en cuenta que, debido a las condiciones de KKT, $\alpha_j > 0$ en (3.20), implica que:

$$y_j\left[\left\langle x_j, \sum_{i=1}^{l} \alpha_i y_i x_i \right\rangle + b\right] - 1 = 0 \underset{(3.22)}{\Rightarrow} y_j\left[k\left(x_j, \sum_{i=1}^{l} \alpha_i y_i x_i\right) + b\right] = 1 \Rightarrow \sum_{i=1}^{l} \alpha_i y_i k(x_j, x_i) + b = 1 / y_j \underset{y_j = \pm 1}{=} y_j$$

por lo cual: $b = y_j - \sum_{i=1}^{l} \alpha_i y_i k(x_j, x_i)$ (3.24)

sobre todos aquellos puntos con $\alpha_j > 0$ (esto es, sobre los vectores soporte).

3.6 El caso no separable

¿Qué ocurre cuando lo anterior no funciona, o no lo hace todo lo bien que esperábamos? En la práctica, el hiperplano de separación no tiene por qué existir, o, si existe, es posible que no sea la solución óptima al problema de clasificación. Y es que aunque sólo sea un punto el que se encuentre aislado (por ejemplo, un dato mal clasificado), la trascendencia para el hiperplano

---

condicionalmente definido positivo, si da lugar a una matriz Gram condicionalmente definida positiva para todo *l*≥2 $x_1, \quad , x_l ? \mathrm{K}$. Esto es, si los elementos k_{ij} de la matriz *K* verifican $\sum_{i,j=1}^{l} c_i c_j k_{ij} \geq 0 \quad \forall c_i \in \mathrm{K}, \text{ con } \sum_{i=1}^{l} c_i = 0$.

resultante es crucial. Por ello, sería deseable tener un algoritmo con una cierta tolerancia hacia este tipo de entradas.

La idea natural sería pedir al algoritmo que devuelva un hiperplano que conduzca al mínimo número de errores de entrenamiento posible. Sin embargo, es ese un problema combinatorio de considerable dificultad.

La solución encontrada por Cortes y Vapnik ([Cortes y Vapnik, 1995]) pasa por introducir unas variables de holgura con objeto de permitir la existencia de ejemplos que violen la restricción (3.16):

$$\xi_i \geq 0 \quad \text{para } i = 1,\ldots,l \tag{3.25}$$

De este modo, las restricciones pasarían a ser:

$$y_i\left(\mathbf{w}^t \cdot \mathbf{x}_i + b\right) \geq 1 - \xi_i \qquad \forall\, i = 1,\ldots,l \tag{3.26}$$

Resulta inmediato que, si hacemos ξ_i suficientemente grande, la restricción para $(\mathbf{x}_i, y_i)$ siempre se puede alcanzar. Por otro lado, no nos interesa que la solución que se alcance ante valores grandes de ξ_i sea la trivial, de manera que introduciremos en la función objetivo una penalización, añadiendo el término $\sum_i \xi_i$.

La formulación del problema quedaría ahora:

Encontrar $\mathbf{w} \in \mathrm{H}$, $\xi_i \in \mathbb{R}$ $i = 1,\ldots,l$ que

minimicen $\tau(\mathbf{w},\boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{l}\sum_{i=1}^{l}\xi_i$ (3.27)

sujeto a $y_i\left(\mathbf{w}^t \cdot \mathbf{x}_i + b\right) \geq 1 - \xi_i$ y $\xi_i \geq 0 \quad \forall\, i = 1,\ldots,l$"

donde C es un parámetro que el clasificador deberá estimar[20].

Además, las variables de holgura determinarán si un punto es un error de margen, esto es, puntos que, o son errores, o caen dentro del margen. Serán aquellos cuyas ξ_i sean estrictamente positivas.

Además, si muchas de las ξ_i toman valores grandes, o dicho de otro modo, si las clases que debemos separar se solapan sobremanera, por ejemplo debido al ruido, entonces el término $\sum_i \xi_i$ representará un valor significadamente mayor que el de la fracción de errores de margen. En tal caso, no existe garantía de que el hiperplano solución generalice bien.

[20] En principio, no hay un modo de seleccionar C a priori. Se suele tomar C/m=10.

Al igual que en el caso separable, la solución que el llamado clasificador C-SV encuentra al problema (3.23) tiene una expansión de la forma:

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \tag{3.28}$$

Donde los coeficientes α_i sólo toman valores estrictamente positivos cuando el ejemplo correspondiente ($\mathbf{x}_i$,y_i) alcanza la restricción (3.26). De nuevo, el problema sólo depende de los productos escalares sobre H, calculable mediante el kernel.

Los coeficientes α_i se hallan resolviendo el programa cuadrático en $\boldsymbol{\alpha} \in \mathbb{R}^l$:

$$\text{“}MaxW(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{3.29}$$

$$\text{sujeto a } 0 \le \alpha_i \le \frac{C}{m} \quad \forall i = 1,...,l \text{ y a } \sum_{i=1}^{l} \alpha_i y_i = 0\text{”} \tag{3.30}$$

Para calcular el valor del umbral *b* hay que tener en cuenta que, debido a la restricción (3.26), para los Vectores Soporte para los que ξ_i = 0, tenemos la expresión (3.24) sobre todos los vectores soporte $\mathbf{x}_j$ con $\alpha_j < C$.

Convergencia

Tal y como hemos visto hasta ahora, la SVM se basa en la minimización de un problema cuadrático, que se resuelve frecuentemente empleando programación cuadrática [Burges 1998]. Recientemente se ha propuesto un algoritmo que construye la SVM resolutora más rápida. Se trata del procedimiento IRWLS (*iterative reweighted least square*) para resolver SVM para la clasificación, que fue introducido por Pérez-Cruz, Navia-Vázquez, Rojo-Álvarez y Artés-Rodríguez (1999), y Pérez-Cruz, Navia-Vázquez, Alarcón-Diana y Artés-Rodríguez (2001), y fue empleado en Pérez-Cruz, Navia-Vázquez, Alarcón-Diana y Artés-Rodríguez (2000) para construir la SVM que resolvía el problema con la mayor velocidad. Solucionaba una secuencia de problemas de mínimos cuadrados ponderados, que, a diferencia de otros procedimientos de mínimos cuadrados como las SVM Lagrangianas (Mangasarian & Musicant, 2000) o la SVM de mínimos cuadrados (Suykens & Vandewalle, 1999; Van Gestel et al., 2004) conduce a la solución correcta de la SVM. Sin embargo, para probar la convergencia de la solución de la SVM, el algoritmo IRWLS, con la formulación que aparece en Pérez-Cruz et al. (1999, 2001), hubo de modificarse (Pérez-Cruz, Bousoño-Calzón, Artés Rodríguez, 2005).

El clasificador por vectores soporte busca computar la dependencia entre el conjunto de patrones $\{\mathbf{x}_i\}, i \in \{1,...l\}$ y sus correspondientes etiquetas $y_i \in \{\pm 1\}$

dada la transformación Φ al espacio de características.

El clasificador resolvía, recordemos, el problema $\min_{\mathbf{w},\xi_i,b}\left\{\frac{1}{2}\|\mathbf{w}\|^2+\frac{C}{l}\sum_{i=1}^{l}\xi_i\right\}$

sujeto a $y_i\left(\mathbf{w}^t\cdot\Phi(\mathbf{x}_i)+b\right)\geq 1-\xi_i$ y $\xi_i\geq 0 \quad \forall\, i=1,...,l$

Este problema es equivalente al siguiente problema sin restricciones, en el que se pretende minimizar el funcional:

$$L_p(\mathbf{w},b)=\frac{1}{2}\|\mathbf{w}\|^2+\frac{C}{l}\sum_{i01}^{n}L(u_i)$$

con respecto a **w** y *b*, donde y $L(u)=\max(u,0)$.

Para probar la convergencia del algoritmo es necesario que $L_p(\mathbf{w},b)$ sea continuo y diferenciable, por lo que se sustituye $L(u)$ por una aproximación que tienda a $\max(u,0)$ cuando *K* tiende a infinito. A continuación, para construir el algoritmo IRWLS hace el desarrollo de Taylor de primer orden de $L(u)$, modificando la ecuación anterior para $L_p(\mathbf{w},b)$. Después, se construye la aproximación cuadrática, bajo ciertas imposiciones, de la nueva ecuación. El procedimiento IRWLS consiste en minimizar esta ecuación, recalculando determinados términos hasta que se alcance la solución. En Pérez-Cruz, Bousoño-Calzón, Artés Rodríguez (2005) se prueba, como decíamos que este procedimiento converge a la solución de la SVM.

3.7 Clasificación en múltiples clases

Hasta el momento, todo lo que hemos establecido se refiere al caso de clasificación en dos clases. Sin embargo, son muchos los problemas reales que requieren una separación de los datos en múltiples clases.[21] Veremos a continuación algunos métodos que tratan de lidiar con esta situación ([Bredensteiner y Bennett 1999], [Weston y Watkins 1999]). Probablemente, no hay ninguno mejor que otro, y la elección del que empleemos finalmente dependerá de la naturaleza intrínseca del problema que tengamos que resolver, de las restricciones que tengamos que manejar, y del tiempo del que dispongamos para su desarrollo y entrenamiento. A pesar de ello, el primero de los métodos resulta, además del más sencillo, el más empleado por los investigadores.

- *Uno Frente al Resto*

La idea es sencilla: para obtener un clasificador en *M* clases, se construyen *M* clasificadores binarios $f^1,...,f^M$ como los tratados en los apartados anteriores,

[21] De hecho, una extensión natural del problema que dejaremos como una línea abierta de investigación, será la de la clasificación de los asegurados en múltiples clases.

de modo que cada uno de ellos separe una clase del resto (si sólo para un *i* $f^i(x)>0$, o $f^i(x)<0$, *x* pertenecerá a la clase *i*, pero si eso ocurre para más de un *i*, entonces la clase *i* será considerado ambiguo), para luego combinarlos realizando la clasificación multi-clase de acuerdo a la máxima salida calculada antes de aplicar la función signo; es decir, tomando

$$\arg\max_{j=1,\ldots,M} g^j(x) \text{ donde } g^j(x) = \sum_{i=1}^{l} \alpha_i^{\,j} y_i k(x, x_i) + b^j \quad [22] \qquad (3.31)$$

Los valores de $g^j(x)$ se pueden emplear también para rechazar decisiones.

Si consideramos la diferencia entre los dos valores mayores de $g^j(x)$ como una medida de la confianza en la clasificación de *x*, podemos compararla con un cierto umbral θ, de manera que si la medida es menor que el umbral, el clasificador rechaza la entrada y no la asigna a ninguna clase (en ese caso, la clasificación la podría hacer un experto humano). Además, la consecuencia inmediata es que se puede lograr una tasa de error menor para las restantes muestras.

La principal pega que tiene la aproximación (3.31), es que posee un cierto toque heurístico. Los clasificadores que se utilizan se obtienen entrenando diferentes problemas de clasificación binarios, y por ello no resulta claro si las salidas que han calculado lo son sobre escalas comparables. Esto puede suponer un problema, pues puede darse la situación de que varios clasificadores asignen el dato a su respectiva clase (o ninguno lo haga); en ese caso, se puede escoger *una* clase comparando las salidas. Además de esto, se critica a estos clasificadores por trabajar con problemas demasiado asimétricos.

- *Clasificación por Parejas*

En este caso, se entrena un clasificador para cada posible par de clases .Para *M* clases, serán un total de (*M*-1)*M*/2 clasificadores binarios, un número habitualmente mayor que el de los clasificadores de "uno frente al resto" (por ejemplo, para *M*=12, tendremos 66 clasificadores, frente a los 12 del otro método).

Aunque pueda parecer que este hecho conlleve un mayor tiempo de entrenamiento, no es así, pues los problemas individuales que hay que entrenar son significativamente más cortos.

Algo parecido podemos alegar en cuanto a la velocidad de ejecución. Cuando tratamos de clasificar un dato de test, evaluamos los 66 clasificadores binarios, y le asignamos la clase que haya recibido más *votos*. Los clasificadores

[22] Obsérvese que $f^j(x) = sign(g^j(x))$, por (2.20)

individuales son, sin embargo, de menor tamaño (tienen menos vectores soporte) de lo que hubieran sido con el procedimiento anterior.

Esto se debe a dos razones: primero, los conjuntos de entrenamiento son más pequeños, y segundo, los problemas a aprender son normalmente más sencillos, puesto que las clases se superponen menos.

A pesar de ello, las comparaciones no suelen valer cuando *M* es grande, y la clasificación de "uno frente al resto" puede ser más rápida que la de "por parejas".

- *Error-Corrección de los Códigos de Salida*

Este método fue desarrollado primero para problemas de aprendizaje de clasificación multiclase en 1995 , y luego adaptado para SVM . Básicamente, consiste en generar un problema binario a partir de uno multiclase separando una clase del resto, generando así una serie de problemas binarios dividiendo el conjunto original de clases en dos subconjuntos.

Resulta obvio que si diseñamos correctamente un conjunto de clasificadores binarios $f^1,...,f^L$, las respuestas a los mismos determinarán completamente la clase de unas muestras de test.

Cada clase corresponde a un vector único en $\{\pm 1\}^L$; para *M* clases obtendremos así la llamada "matriz de decodificación" $M \in \{\pm 1\}^{M \times L}$.

Pero, ¿qué ocurre si las respuestas binarias son inconsistentes entre sí; si, por ejemplo, el problema tiene ruido, o la muestra de entrenamiento resulta demasiado pequeña para estimar con seguridad? Formalmente, tendríamos un vector de respuestas $f^1(x),...,f^L(x)$ sin ninguna ocurrencia en la matriz de decodificación. Para evitar estas situaciones, en [Crisp y Burges 2000] se proponía diseñar un conjunto de problemas de dos clases "inteligentes", que posean robustez frente a algunos errores. En ese caso, la correspondencia más cercana entre el vector de respuestas y las filas de *M* se determina con la distancia de Hamming (el nº de entradas en las que difieren dos vectores; es básicamente la distancia L_∞). Por ejemplo, si el código es tal que la distancia de Hamming mínima es tres, podremos "garantizar" que lograremos clasificar correctamente todos los ejemplos de test que conduzcan a al menos un error entre los clasificadores binarios.

Este método genera resultados muy buenos en las tareas multiclase, pero se ha observado que no hace uso de un dato fundamental para los clasificadores: el margen. Por ello, se ha desarrollado una versión [5] que sustituye la decodificación basada en la distancia de Hamming por un esquema más complicado que sí tiene en cuenta el margen.

- *Funciones Objetivo Multiclase*

A pesar de que los algoritmos anteriores son muy eficientes y elaborados, es indudable que lo más elegante, y también acorde con el principio de Vapnik de tratar siempre de abordar "directamente" el problema a resolver, sería modificar la función objetivo de la SVM para que permitiera simultáneamente el cómputo de un clasificador en múltiples clases. Por ejemplo [Weston y Watkins 1999], se puede variar (3.27) y utilizar el siguiente programa cuadrático:

Encontrar $\mathbf{w}_r \in \mathrm{H}$, $\xi^r \in \mathbb{R}^l$ y $b_r \in \mathbb{R}$ y $i = 1,\ldots,l$ que

$$\text{minimicen } \frac{1}{2}\sum_{r=1}^{M}\|\mathbf{w}_r\|^2 + \frac{C}{l}\sum_{i=1}^{l}\sum_{r\neq y_i}\xi_i^r \tag{3.32}$$

$$\text{sujeto a } \mathbf{w}_{y_i}{}^t \cdot \mathbf{x}_i + b_{y_i} \geq \mathbf{w}_r{}^t \cdot \mathbf{x}_i + b_r + 2 - \xi_i^r \quad \text{y} \quad \xi_i > 0 \quad \forall\, i = 1,\ldots,l \tag{3.33}$$

donde $l \in \{1,\ldots,M\}/\, y_i$ y $y_i \in \{1,\ldots,M\}$ .es la etiqueta multiclase del patrón $\mathbf{x}_i$

En términos de precisión, los resultados que se obtienen con este método son comparables a los conseguidos con el uso más amplio del clasificador "uno frente al resto". Lamentablemente, la dificultad del problema de optimización es considerable, ya que tiene que manejar *todos* los vectores soporte a la vez.

Esto no ocurría en los otros algoritmos, que lidiaban con conjuntos de vectores soporte más pequeños, con el consiguiente beneficio para el tiempo de entrenamiento.

3.8 Resultados empíricos

Agrupamos a continuación los resultados obtenidos atendiendo a la base de datos utilizada en cada caso, que se corresponde con cada uno de los dos años en estudio.

Datos de 2003

- *Clasificación con todos los factores de riesgo*

Veamos al fin cuáles son los resultados de clasificación de los asegurados en cuanto a si han tenido o no siniestros durante el año 2003.

El software utilizado en este caso es el LibSVM[23]. Se trata de un software integrado para la clasificación por vectores soporte, regresión, y estimación de la distribución. Es capaz de desarrollar también la clasificación en múltiples clases.

En él elegimos un kernel RBF (función de base radial[24]) y la validación cruzada en **5** "pliegues", para ambas pruebas.

El resultado principal es la *tasa de acierto* [(Nº de aciertos) / (Nº de casos)]. Se dan también la precisión ponderada por clase (media entre la tasa de acierto para todas las clases), y la matriz de confusión. Recuérdese que esta es la matriz que resulta de la clasificación:

	CLASE 1	**CLASE 2**
CLASE 1	Acierto 1	Error 12
CLASE 2	Error 21	Acierto 2

Donde, Acierto 1 son los casos de la clase 1 clasificados como clase 1 (es decir, correctamente); Acierto 2 son los casos de la clase 2 clasificados como clase 2; error 21 son los casos de la clase 2 clasificados como clase 1, y ; error 12 son los casos de la clase 1 clasificados como clase 2.

En el primer experimento (con el primer grupo de datos tal y como expusimos en el Capítulo 2), obtuvimos la siguiente *tasa de acierto*: 77.72%

En términos de la matriz de confusión, el resultado sería:

-1 -> 76.41% (21638/28318)
1 -> 79,13% (23675/29919)

Matriz de confusión $\begin{bmatrix} 21638 & 7320 \\ 6244 & 23675 \end{bmatrix}$

Obsérvese como el nº de clasificados correctamente entre los que han tenido siniestro es ligeramente superior al de los que no tuvieron. Esto es lógico, debido a que el nº de asegurados que presentaron siniestros es también un poco mayor que el de los que no lo hicieron.

[23] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[24] $K(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right), \gamma > 0$

- *Clasificación tras la selección con el Algoritmo Genético*

En el Capítulo siguiente veremos dos herramientas para seleccionar factores, cuyo interés de cara a la clasificación, como ya hemos comentado, es el de mejorar el funcionamiento (capacidad de generalización, tiempo de ejecución, etc.) del clasificador. Por ello, tras escoger los factores de riesgo más relevantes, se ejecutó de nuevo la SVM. Esto se realizó con los resultados obtenidos por el Algoritmo Genético (datos de 2003) y con los del Árbol de Clasificación (datos de 2005).

En el caso de los datos de 2003, el AG eligió 30 variables. El resultado de la clasificación con sólo esos 30 factores fue que la tasa de clasificación era de un 77.66%. Como se puede apreciar, la diferencia en la tasa de clasificación es del 0,12 % (77.72% con todas las variables frente al 77.66% utilizando sólo las 30 mejores). Se trata de un valor muy pequeño.
Es pues un resultado excelente de cara a la aseguradora, pues significa que utilizando menos factores de riesgo seremos capaces de clasificar a los nuevos asegurados prácticamente con la misma exactitud. Esto supone una gran ventaja para la Compañía, pues ahorrará en tiempo y dinero a la hora de recoger los factores de riesgo de sus clientes. El haber eliminado información redundante para el sistema hará, por otra parte, que mejore la estabilidad del clasificador, y que el coste computacional del proceso sea menor. Sí que hay que admitir, sin embargo, que se esperaba una mejor tasa de clasificación utilizando menos factores de riesgo; el motivo de que no sea así habría que buscarlo en la naturaleza de los datos, y en su número.

Datos de 2005

- *Clasificación con todos los factores*

Como ya se comentó en el Capítulo anterior, en las muestras que tomamos de la base de datos correspondiente a 2005, había una que incluía el nivel de Bonus Malus, y otra que no. Aunque el interés principal que tiene esta distinción es de cara a la selección de factores, lógicamente se realizaron también las clasificaciones con las dos muestras.

En este caso, se empleó una SVM usando validación cruzada de 10 pliegues para selección de parámetros y evaluación de resultados de modo que las muestras para evaluar las prestaciones nunca se usan para ajuste de parámetros.

El resultado en las dos muestras en niveles de clasificación fue el siguiente:

	SIN NIVEL BM	CON NIVEL BM
Tasa de clasificación	70,10%	72,23%

Obsérvese que se produce una mejoría significante en el momento de incluir los niveles de Bonus Malus como variables, de más del 2%. Eso indica lo que en el Capítulo siguiente tendremos ocasión de comprobar, que estos factores son realmente influyentes a la hora de estudiar la siniestralidad.

- *Clasificación tras la selección con el Árbol de Clasificación*

Al igual que hicimos en el caso de los datos 2003 y el experimento con el algoritmo genético, procedemos ahora a clasificar nuevamente los asegurados utilizando únicamente los factores seleccionados por el árbol de clasificación, cuyos resultados se observarán en el siguiente capítulo. Los datos sobre la SVM que se emplea para ello son los mismos que entonces, salvo que para validar se usó la validación cruzada en 10 pliegues, en vez de en 5.

Los resultados los podemos observar a continuación en la tabla:

	SIN NIVEL BM	**CON NIVEL BM**
Tasa de clasificación tras la selección	70,10%	72,23%

Como se puede apreciar, el hecho de incluir el nivel de Bonus Malus como factor en los datos, redunda de nuevo en una mayor tasa de clasificación. Comparándola con la de los datos sin nivel de BM, la tasa aumenta en un 2% cuando se utilizan todos los factores, y en un 3% cuando sólo se toman los factores seleccionados por el árbol. Asimismo, se observa que ahora sí se mejora la tasa de clasificación cuando se utilizan sólo las variables seleccionadas.

Resumiendo, si tenemos en cuenta las tasas de clasificación, en los dos experimentos (el del AG y el del árbol) se obtiene la misma conclusión: la diferencia en la tasa de clasificación con todas las variables y sólo con las 30 que escoge cada algoritmo es muy pequeña, con las consecuencias positivas que ya hemos comentado, tanto de cara a la Compañía, como al funcionamiento del clasificador.

4. ANÁLISIS DISCRIMINANTES. RESULTADOS

Dedicaremos el presente apartado a esta metodología, no porque sea más relevante o usada que las otras que ya hemos presentado en Capítulo anterior, sino por ser la que hemos seleccionado para comparar parte del estudio que hemos realizado en la aplicación con las técnicas de aprendizaje máquina.

4.1 Introducción

Con independencia del área de conocimiento en la que se está trabajando, es frecuente tener que enfrentarse con la necesidad de identificar las características que permiten diferenciar a dos o más grupos de sujetos. Como ocurría con las técnicas de aprendizaje, el motivo suele ser el poder clasificar nuevos casos como pertenecientes o uno u otro grupo. El análisis discriminante ayuda a identificar las características que diferencian (*discriminan*) a dos o más grupos y a crear una función capaz de distinguir con la mayor precisión posible a los miembros de uno u otro grupo.

Como es lógico, para llegar a conocer en qué se diferencian los grupos necesitamos disponer de la información (cuantificada en una serie de variables) en la que suponemos que se diferencian. El análisis discriminante es una técnica estadística capaz de decirnos qué variables permiten diferenciar a los grupos y cuántas de esas variables son necesarias para alcanzar la mejor clasificación posible. En este sentido, la pertenencia a los grupos, conocida de antemano, se utiliza como *variable dependiente* (una variable categórica con tantos valores discretos como grupos). Las variables en las que suponemos que se diferencian los grupos se utilizan como *variables independientes* o variables de clasificación (también conocidas como *variables discriminantes*). Éstas deberán ser variables cuantitativas continuas o, al menos, admitir un tratamiento numérico con significado.

El objetivo último del análisis discriminante es encontrar la combinación lineal de las variables independientes que mejor discrimina a los grupos. Una vez hallada esa combinación (la *función discriminante*), podrá ser utilizada para clasificar nuevos ejemplos. Se trata básicamente de una técnica descriptiva, aunque hoy en día se aplica cada vez más con fines decisionales. Dentro de las de análisis multivariante, es de las técnicas más profusamente utilizadas.

El análisis discriminante es aplicable en muy diversas áreas de conocimiento, desde la Medicina a los recursos humanos. En el ámbito económico se ha empleado para la predicción del riesgo a la hora de otorgar un crédito, y en el actuarial para *predecir la siniestralidad*, como ya hemos comentado. Entre sus aplicaciones más destacadas encontramos el Reconocimiento de cifras manuscritas, el sistema experto MENINGE y el Referéndum sobre el acta única europea.

El análisis discriminante presenta dos perspectivas distintas:

- El punto de vista *decisional*, que como decíamos, es el más corriente, para el cual se trata de construir una regla de asignación de los individuos a uno de los grupos que pueda ser aplicada en el futuro. Tal regla estará fundada en un conjunto de variables observadas denominadas también *predictores*. Una regla se considera “buena” si los errores de clasificación cometidos con las observaciones futuras son lo más pequeños posible.

- El punto de vista *explicativo*, que trata de descubrir las variables – o las asociaciones entre los niveles de varias de ellas- más pertinentes para describir las diferencias entre los grupos a priori, y determinar si estas diferencias son significativas.

4.2 Relación con las Máquinas (red) de Soportes Vectores

La manera más sencilla de clasificar en dos grupos los casos nuevos que se presenten por medio de un análisis discriminante consiste en calcular la distancia existente entre los centroides de ambos grupos y situar un punto de corte *d* equidistante de ambos centroides. A partir de ese momento, los casos cuyas puntuaciones discriminantes sean mayores que el punto de corte *d* serán asignados al grupo superior y los casos cuyas *puntuaciones discriminantes* sean menores que el punto de corte *d* serán asignados al grupo inferior.

Aunque esta tentativa de resolución del problema de la clasificación cuenta con serios inconvenientes (sólo vale para dos grupos, y no tiene en cuenta que los grupos pueden tener distinto tamaño), si nos sirve para apreciar una cierta analogía con el planteamiento que se hacía en las SVM.

La aproximación que hacíamos en el apartado anterior para la solucionar del problema era la de diseñar un algoritmo de análisis de patrones para patrones dados por funciones *lineales* en un espacio de características definido por funciones kernel. Para ello, primero debemos recodificar los datos en una aplicación particular de modo que los patrones puedan representarse por funciones lineales. Después, podremos aplicar uno de los algoritmos de análisis de patrones (SVM en nuestro caso) a los datos transformados. La clase de algoritmos que resulta se llama "Métodos kernel", y el análisis discriminante puede considerarse uno de ellos.

Un problema de clasificación supervisado, tal y como expusimos en el apartado anterior, se puede enunciar como:

"Dado el conjunto $S = \{(\mathbf{x}_1, y_1),...,(\mathbf{x}_l, y_l)\} \in \mathbb{R}^n \times \{\pm 1\}$ encontrar una función de predicción $g(\mathbf{x}) = sign(\langle \mathbf{w}, \mathbf{x} \rangle + b)$ tal que la esperanza de la diferencia entre $g(\mathbf{x})$ e y sea pequeña".

Al ser *g* una función lineal, la resolución de esta cuestión puede realizarse "aprendiendo" un hiperplano definido por la ecuación $\mathbf{w}^t \cdot \mathbf{x} + b = 0$ que separe los datos según sus etiquetas. Tanto estadísticos como investigadores de las redes Neuronales han empleado con frecuencia este tipo simple de clasificador, nombrándolo respectivamente *discriminantes lineales* y *perceptrones*. Fisher desarrolló la teoría de los discriminantes lineales en 1936, mientras que los investigadores en Redes Neuronales estudiaron los perceptrones a comienzos de 1960, principalmente en torno al trabajo de Rosenblatt, como ya sabemos.

La influencia de este campo en todo lo que se refiere a clasificación es enorme. Como muestra, el término *vector de pesos* para **w**, se toma prestado de la literatura de las RNA.

Existen muchos algoritmos para determinar **w**, de los cuales hemos visto detenidamente las SVM, y sólo enunciado el perceptrón. Aunque nosotros lo tratemos más desde el punto de vista estadístico como aplicación para comparar los resultados del problema actuarial que nos ocupa, sí nos gustaría señalar que el análisis discriminante también se puede ver como otro de estos algoritmos.

En este sentido, el discriminante de Fisher es una función de clasificación:

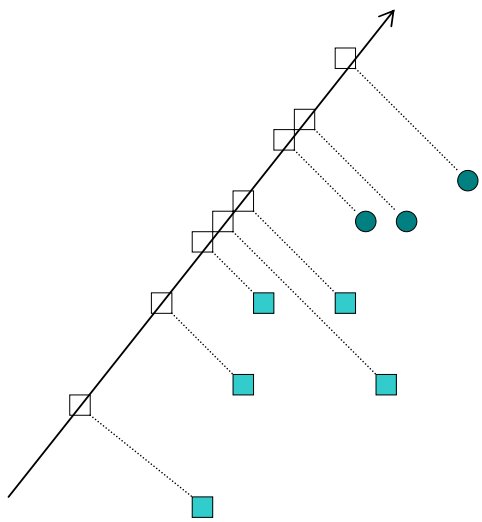
$$f(\mathbf{x}) = \mathbf{sgn}\left(\mathbf{w}^t \cdot \phi(\mathbf{x}) + b\right)$$

donde el vector de pesos se toma de manera que maximice el cociente:

$$J(\mathbf{w}) = \frac{\left(\mu_{\mathbf{w}}^{+} - \mu_{\mathbf{w}}^{-}\right)^2}{\left(\sigma_{\mathbf{w}}^{+}\right)^2 + \left(\sigma_{\mathbf{w}}^{-}\right)^2}$$

siendo $\mu_{\mathbf{w}}^{+}$ la media de la proyección de los ejemplos positivos sobre la dirección w,

$\mu_{\mathbf{w}}^{-}$ la media para los ejemplos negativos, y $\sigma_{\mathbf{w}}^{+}$, $\sigma_{\mathbf{w}}^{+}$ las correspondientes desviaciones estándar.



En la figura anterior se observa la proyección sobre una dirección particular de **w** que proporciona una buena separación de las medias con unas varianzas

pequeñas de los ejemplos positivos y negativos. Con el discriminante de Fisher se maximiza el radio entre esas cantidades. El motivo es que la dirección seleccionada maximiza la separación de las medias escaladas de acuerdo a las varianzas en tal dirección. Como estamos manejando espacios de características definidos por funciones kernel, tiene sentido el introducir una regularización del vector de pesos **w**, pasando a considerarse el *Discriminante de Fisher regularizado*, que toma **w** tal que resuelva el siguiente problema de optimización:

$$\max_{\mathbf{w}} J(\mathbf{w}) = \frac{\left(\mu_{\mathbf{w}}^{+} - \mu_{\mathbf{w}}^{-}\right)^2}{\left(\sigma_{\mathbf{w}}^{+}\right)^2 + \left(\sigma_{\mathbf{w}}^{-}\right)^2 + \lambda\|\mathbf{w}\|^2}$$

Se prueba que el problema anterior puede resolverse mediante el cálculo de un hiperplano en el espacio de características definido implícitamente por una función kernel.

De este modo, el análisis discriminante puede verse como un procedimiento de optimización de una medida de separación de las proyecciones de los datos sobre un espacio 1-dimensional, conectando así con el proceso básico de las SVM.

4.3 Planteamiento general del problema

Veamos ahora el método del Análisis Discriminante desde un punto de vista más "estadístico".

Según decíamos, el análisis discriminante permite analizar las relaciones que existen entre una variable a explicar de naturaleza cualitativa y un conjunto de variables explicativas de naturaleza cuantitativa. Los métodos discriminantes se aplican a poblaciones descritas por variables y dotadas de una partición, definida a priori, y que tiene un interés particular. Estos métodos tratan de *discriminar* (separar) lo mejor posible las clases de la partición con la ayuda de las variables explicativas.

Distinguiremos el problema planteado desde la perspectiva descriptiva o decisional:

1) *El análisis factorial discriminante o análisis discriminante con fines descriptivos*

Sean *n* individuos agrupados en *q* clases y sean un conjunto de *p* variables explicativas definidas sobre el conjunto de individuos.

El objeto de este tipo de análisis factorial es encontrar combinaciones lineales de las variables explicativas originales que nos permitan discriminar lo mejor posible entre los grupos.

El proceso comienza buscando los centros de gravedad de los grupos de individuos. A continuación, se calculan las varianzas dentro de cada grupo y las varianzas entre los grupos (esto es, la matriz de varianzas de los centros de gravedad de los grupos).

De este modo, sean:

V: matriz de varianza total
B: matriz de varianza entre los grupos
W: matriz de varianza dentro de los grupos

La variable que mejor discriminara sería aquella que tomara los mismos valores sobre todos los individuos del mismo grupo, y valores distintos sobre los distintos grupos. Esta variable poseería una varianza intra-grupos nula, mientras que la varianza entre-grupos sería máxima. En general, se tratará de maximizar la varianza entre los grupos, y de minimizar la varianza dentro de los grupos.

Como la suma de varianzas dentro de los grupos y entre los grupos es siempre constante e igual a la varianza total, el problema de análisis factorial discriminante se puede enunciar de la siguiente manera:

$$\text{Buscar } u \quad \text{tal que } \frac{u^t B u}{u^t V u} \text{ sea máximo} \tag{3.34}$$

Se demuestra que los vectores propios de $V^{-1}B$ denotados por $u^1,...,u^{q-1}$ ordenados según el orden decreciente de los valores propios positivos $\lambda_1,...,\lambda_{q-1}$ son las soluciones sucesivas del problema.

2) *El análisis discriminante con fines decisionales*

Supongamos que para un individuo sólo se conocen las variables explicativas. Se sabe que ese individuo pertenece a uno de los grupos definidos por la variable a explicar pero se ignora a cual. El análisis discriminante con fines decisionales permite descubrir a qué grupo pertenece el individuo.

Las técnicas para asignar un individuo suplementario a un grupo permiten además estudiar la "calidad" de la discriminación midiendo para cada clase *k* el porcentaje de individuos bien clasificados.

4.4 El análisis factorial discriminante

Consideremos un conjunto de *n* individuos sobre el cual se observa una variable cualitativa que toma *q* modalidades. Cada individuo se identifica con una sola modalidad de la variable, y se define por tanto una partición del

conjunto de los individuos en *q* clases disjuntas. Además, se miden sobre los mismos individuos *p* variables cuantitativas. Así, se plantea el siguiente problema: ¿*las q clases difieren sobre el conjunto de variables explicativas*? El objetivo del Análisis Factorial Discriminante (AFD) es responder a esta cuestión.

Al igual que en el Análisis de Componentes Principales, se determina una nueva variable, combinación lineal de las antiguas variables. Sin embargo, ya no se trata de obtener una variable de varianza máxima, sino de separar lo mejor posible los *q* grupos entre ellos. Más concretamente, se desea que esta nueva variable tome valores:

- lo más cercanos posible para los individuos que pertenezcan a un mismo grupo
- lo más distintos posible para los individuos que pertenezcan a grupos distintos

De éste modo, buscaremos aquella combinación lineal cuya varianza provenga en su mayor parte de las diferencias entre grupos, y en lo mínimo posible debido a la varianza dentro de los grupos. Una vez encontrada la primera combinación de variables con mayor poder discriminante buscaremos, entre las combinaciones no correladas con la primera, aquella que mejor discrimine, y así sucesivamente.
En el enfoque decisional nos plantearemos, como ya hemos comentado, la asignación de un individuo suplementario a una clase, conocidos los valores que toman las variables explicativas sobre dicho suplementario. Esto se hará asignando el individuo a la clase a la que esté más próximo.

Como decíamos en (3.34) *u* debe hacer máxima la cantidad

$$\lambda = \frac{u^t B u}{u^t V u} \quad (0 \le \lambda \le 1) \tag{3.35}$$

Utilizando la misma técnica que en el análisis de componentes principales (derivando e igualando a cero para obtener el máximo), se obtiene que

$$V^{-1} B u = \lambda u \tag{3.36}$$

De aquí se deduce que *u* de be ser el autovector de $V^{-1}B$. Su valor propio correspondiente, λ debe ser el más grande, ya que representa la cantidad a maximizar.

Sea $\mathbf{u^1}$ la solución. Entonces $\mathbf{u^1}$ es el *primer factor discriminante*, y λ_1 es su *poder discriminante*.
Una vez obtenida la primera variable discriminante $\mathbf{c^1} = X\mathbf{u^1}$ se busca $\mathbf{c}^2 = X\mathbf{u}^2$ incorrelada con $\mathbf{c^1}$ y

tal que $\frac{u^t B u}{u^t V u}$ sea máxima, y así sucesivamente.

La interpretación de los factores puede hacerse como en análisis de componentes principales calculando las correlaciones entre factores y variables observadas y representando el círculo de correlaciones. Es interesante representar también a los individuos (caracterizados por su grupo de pertenencia) sobre el plano factorial. El hacerlo así nos permitirá describir las especificidades de cada grupo en término de los factores y, en consecuencia, de las variables originales. Más concretamente, nos otorgará la posibilidad de detectar aquellos individuos que no sigan la tendencia general del grupo al que han sido asignados.

4.5 El análisis discriminante decisional

Se trata de clasificar a un individuo suplementario. Para ello, deben construirse reglas de decisión que produzcan un mínimo de errores cuando se apliquen en el futuro. Para responder correctamente a este objetivo, es necesario situarse en un marco probabilística preciso.

Planteémonos a continuación el siguiente problema: ¿el solo conocimiento de las variables explicativas permite asignar un individuo a su grupo de pertenencia definido por la variable a explicar? Supongamos que para un individuo sólo se conocen sus variables explicativas. Se sabe que ese individuo pertenece a uno de los grupos definidos por la variable a explicar, pero se ignora a cual. ¿Es posible asignarlo a uno de esos grupos y ello con un error mínimo?.
Las técnicas de resolución del problema dependen de la naturaleza de los datos y de las hipótesis de trabajo. Permiten además estudiar la calidad de la discriminación midiendo para cada clase *k* el porcentaje de individuos bien clasificados. Son muchos los métodos de discriminación desarrollados, y citaremos tan solo dos: el *método geométrico* , que consiste en asignar un individuo al grupo cuyo centro de gravedad esté más próximo (para medir la distancia se utiliza una métrica concreta, la “Métrica de Mahalanobis”, que en general es cualquier inversa de una matriz de varianzas-covarianzas), y el *método bayesiano*, en el que el individuo es asignado al grupo “más probable” (la variable aleatoria que define el grupo a priori está dotada de una ley de probabilidad).

4.6 El análisis discriminante “paso a paso”

Lo que se conoce como “métodos paso a paso” consisten en introducir en la función discriminante las variables de forma sucesiva según la importancia de su aporte condicional a las variables ya introducidas.

Si se consideran los valores propios asociados a los factores discriminantes, entonces se tiene que $\sum_i \lambda_i = Traza\left(V^{-1}B\right)$.

Como criterio de selección de variables se empleará el siguiente: se escoge en

primer lugar la variable que, considerada sola, haga máxima $Traza\left(V^{-1}B\right)$. A continuación, se busca, entre las otras variables aquella variable que añadida a la primera, haga máxima $Traza\left(V^{-1}B\right)$, etc. El algoritmo para cuando el valor de la traza disminuye. El procedimiento, sin embargo, no garantiza que en cada paso se obtengan las variables que proporcionen la mayor discriminación sobre los primeros ejes, ni tampoco que se logren las mejores variables discriminantes, pero sí es un modo de llegar en la práctica a un resultado aceptable, rápidamente y a bajo coste.

4.7 Resultados empíricos

Recordemos que se pretende hacer una clasificación de los siniestros a partir de las variables contenidas en la base de datos. Los datos que emplearemos serán los mismos que utilizamos en el primer experimento con la SVM, y que ya describimos entonces. Se realiza sólo la comparación con uno de los experimentos, a pesar de que hemos realizado varios, debido a que ese es el que da las mejores tasas de clasificación de entre todas las ejecuciones.

Como ya hemos visto, el análisis discriminante es una técnica que se aplica solamente a variables cuantitativas viniendo caracterizados los grupos a través de una variable categórica, en nuestro caso ésta será la variable siniestros que toma valores -1 y 1.

Nuestras variables son en su mayor parte de tipo cualitativo. De forma concreta, la base de datos está divida de la siguiente forma:

- *Variables cuantitativas* (o continuas): Antigüedad Carnet, Edad del Conductor, Antigüedad Vehículo, Potencia y Valor.
- Variables cualitativas (o nominales): el resto.

Es necesario por tanto, para aplicar un análisis discriminante, transformar previamente el conjunto de variables en variables cuantitativas/continuas.

Para ello se procederá de las siguiente forma:

1. Se realizará un Análisis Factorial Múltiple (AFM) sobre el conjunto completo de las variables.
2. Selección de aquellos factores resultantes del AFM que expliquen el 100% de la variabilidad total.
3. Realización del Análisis Discriminante (AD) con los factores anteriormente seleccionados.

Como estudiaremos en el siguiente Capítulo, el AFM es una técnica de análisis factorial que trata tablas en las cuales un conjunto de individuos viene descrito por varios grupos de variables. En el seno de un mismo grupo, las variables deben ser del mismo tipo (continuo o nominal) pero de un grupo a otro, las variables pueden ser de diferentes tipos. Nosotros llevaremos a cabo un AFM

sobre el conjunto de individuos descritos a través de dos tablas, la primera formada por las variables continuas y la segunda formada por las variables nominales. El método es en realidad un análisis factorial del conjunto de grupos (llamado análisis global). Para las variables continuas, el AFM se comporta como un Análisis en Componentes Principales, para las variables nominales como un Análisis de Correspondencias Múltiples. Es la introducción de pesos en las variables, que equilibran las inercias axiales máximas de los grupos, lo que hace posible la presencia simultánea de variables continuas y nominales. El objetivo final es obtener los principales factores de variabilidad de los individuos, estando estos descritos de forma equilibrada por varios grupos de variables. De los factores obtenidos con el AFM se seleccionaron los 59 primeros ya que estos explican el 100% de la variabilidad total[25].

Estos 59 factores fueron introducidos como variables discriminantes en un Análisis Discriminante realizado con el conocido software SPSS. El programa SPSS (Statistical Product and Service Solutions) es un conjunto de potentes herramientas de tratamiento de datos y análisis estadístico, muy popular tanto en el ámbito empresarial como en el de la investigación. Entre los métodos estadísticos que desarrolla están: creación de tablas de contingencia, frecuencias, descriptivos, Estadísticos Bivariantes, Pruebas no paramétricas, Regresión Lineal, Análisis Factorial, Análisis de Conglomerados, Análisis Discriminante. Los resultados de aplicar el SPSS se expresan, como ya hicimos antes con la SVM, en una matriz de confusión. Como aquella, el grupo al que de hecho pertenece el asegurado está en las filas, y el pronosticado se recoge en las columnas. La diferencia con la que contenía los resultados de la SVM es que aquí se pueden observar también los porcentajes de clasificados. Concretamente, la tabla ofrece las frecuencias absolutas, los porcentajes de fila y el porcentaje total de clasificaciones correctas, de modo que en la diagonal principal aparece el porcentaje de clasificaciones correctas. Así, la función consigue clasificar correctamente un 69,1% de los casos de los asegurados que no tuvieron siniestro, y un total de 71,8% de los que sí. De este modo, podemos decir que el resultado fue que los 59 factores clasificaban correctamente los siniestros en el 70,5% de los casos.

Resultados de la clasificación [a]

			Grupo de pertenencia pronosticado		
		SINIESTR	-1,00	1,00	Total
Original	Recuento	-1,00	19581	8737	28318
		1,00	8428	21491	29919
	%	-1,00	69,1	30,9	100,0
		1,00	28,2	71,8	100,0

a. Clasificados correctamente el 70,5% de los casos agrupados originales.

[25] Veremos este análisis con detenimiento en el Apéndice.

5. COMPARACIÓN DE LOS RESULTADOS: SVM *vs* ANÁLISIS DISCRIMINANTE (DATOS 2003)

Tras haber ejecutado el Análisis Discriminante, comparamos los resultados que se han obtenido mediante los dos procedimientos. Aunque en el caso de la SVM realizamos dos experimentos con dos bases de datos diferentes (el motivo era comparar luego con las soluciones alcanzadas después de seleccionar los factores con el Algoritmo Genético, por un lado, y el árbol de clasificación, por otro), ya hemos visto que el AD sólo lo hemos aplicado sobre una de éstas, la primera. La siguiente tabla resume la comparativa:

	SVM			ANÁLISIS DISCRIMINANTE		
Tasa de clasificación	77,72%			70,5%		
Matríz de confusión		-1	1		-1	1
	-1	21638 76,41%	7320	-1	19581 69,1%	8737
	1	6244	23675 79,13%	1	8428	21491 71,8%
Software	Libsvm Kemel $k(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\left\|\mathbf{x} - \mathbf{x}_i\right\|^2 / c\right)$ Valoración cruzada en 5 pliegues			SPSS SPAD (análisis factorial)		

Lo más destacable de la comparación entre ambas técnicas es el resultado final: mientras que la SVM alcanza un 77.72% de clasificados correctamente, el AD llega a un 70.5 % de los casos. La mejora, un 7.22 %, es importante. Y lo es aún más si no lo miramos en términos absolutos, es decir, si lo que valoramos es el hecho de que hemos conseguido superar una tasa de clasificación que ya era muy buena. Cuando esto ocurre, cuando se mejoran resultados que ya eran satisfactorios, cada tanto por ciento es un "paso de gigante", y supone un gran esfuerzo. Contrastamos la matriz de confusión obtenida con ambos procedimientos: El número de clasificados correctamente en las dos clases es mayor con la SVM que con el Análisis Discriminante, 21638 sin siniestro frente a los 19581, y 23675 clasificados en el grupo de los que tuvieron siniestro con los 21491 que incluyó en esa clase el AD. Es curioso observar que la proporción entre los clasificados correcta y erróneamente es muy parecida en los dos clasificadores. A la vista de los resultados, creemos que la aplicación de las Máquinas de Vectores Soporte a la clasificación de asegurados en función de la ocurrencia de siniestros es muy positiva, y merece la pena frente a los métodos "tradicionales" como el Análisis Discriminante.

CAPÍTULO 3
TÉCNICAS DE SELECCIÓN DE CARACTERÍSTICAS

En este Capítulo utilizaremos nuevas técnicas de aprendizaje para la resolución del segundo aspecto de nuestro problema, el de la selección de los factores de riesgo más relevantes. En nuestro caso, la selección de estos factores resulta de utilidad por dos motivos, como ya hemos comentado: el primero, común a los problemas de clasificación tal y como los hemos planteado aquí, para mejorar el funcionamiento del clasificador. El segundo, puramente actuarial, para cubrir esa etapa del proceso de tarificación. Sí nos gustaría añadir que en el caso particular de los datos de 2005, hemos realizado análisis complementarios a los anteriores, observando la influencia del nivel de Bonus Malus ejecutando experimentos sobre muestras de datos que incluían y que no incluían este interesante factor de riesgo.

Las técnicas de aprendizaje que hemos utilizado para seleccionar los factores son, indicábamos en la Introducción, los Algoritmos Genéticos y los Árboles de Clasificación. Son dos herramientas, y como contábamos con dos bases de datos (de dos años distintos, y con factores de riesgo diferentes), hemos dedicado una a cada técnica, de modo que la selección con AG se ejecuta con la base de 2003, y los AC trabajan con la de 2005.

Tras seleccionar las variables se procede a una nueva clasificación de los asegurados con la SVM, lo que nos permite comparar la tasa de clasificación obtenida con la que se logró utilizando todos los factores de riesgo que presentábamos en el capítulo anterior.

Del mismo modo que se hizo en el Capítulo anterior, comenzaremos describiendo las técnicas empleadas, para luego pasar a los resultados obtenidos con los experimentos.

Existen varias herramientas estadísticas que se pueden utiliza en problemas de selección de características. Nosotros hemos seleccionado una, de la que ya hablamos en el Capítulo anterior, el Análisis Factorial. Sí nos gustaría señalar que la utilidad que tiene para nosotros en este caso es como “preprocesador” para la ejecución del Análisis Discriminante que se llevó a cabo en el Capítulo anterior. Esto es, la utilidad que nos reporta es en el sentido de mejora del clasificador. No la emplearemos para escoger los factores relevantes, y luego comparar resultados, como hacíamos antes. Por ello, no la incluiremos en el presente Capítulo, sino en un Apéndice colocado al final del Trabajo.

1. INTRODUCCIÓN Y MOTIVACIÓN

El reconocimiento de patrones, o “el acto de tomar datos y ejecutar una acción basada en la “categoría” de la muestra” [Duda et al 2001], ha resultado crucial para nuestra supervivencia. A lo largo de los últimos diez millones de años, hemos desarrollado un sistema neuronal y cognitivo altamente sofisticado para llevar a cabo tareas aparentemente triviales de reconocimiento de patrones, pero detrás de las cuales subyacen procesos de elevada complejidad. Son actos tales como el de reconocer una cara, la comprensión oral, la lectura de caracteres escritos, o el saber si una fruta está madura por su olor. Es natural que el hombre haya tratado de construir máquinas capaces de llevar a cabo esos procesos.

Para que un problema de clasificación tenga éxito, a menudo es necesario que se reduzca la cantidad de información proporcionada, especialmente cuando en la mayoría de los casos, se maneja información que no es relevante para el proceso. Es por ello que se suele recurrir a lo que se conoce como *extracción de características*, procedimiento por el cual se seleccionan las características más importantes de la muestra, las que más información proporcionan para su posterior clasificación. Se trata, por tanto, de caracterizar un objeto para que sea reconocido por medidas cuyos valores son muy similares para objetos en la misma categoría, y muy diferentes para objetos de diferentes categorías. Esto nos lleva a la idea de buscar características “distintivas” que sean invariantes frente a transformaciones irrelevantes del dato de entrada.El siguiente esquema refleja el proceso habitual de Clasificación mediante sistemas de reconocimiento de patrones:

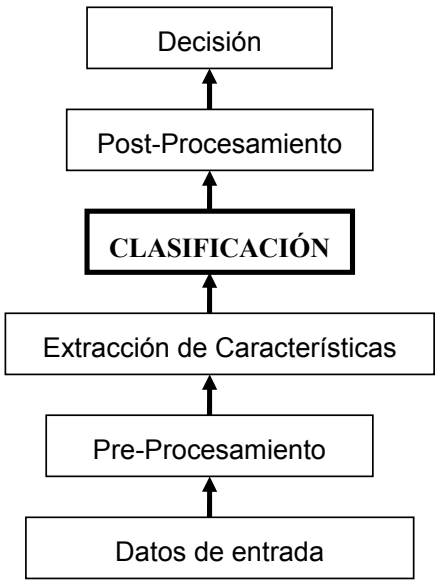


Figura 4.1

Normalmente, se precisa un Pre-Procesamiento de los datos de entrada, ya sea por que el tipo de datos no es el necesario para la alimentación del sistema, o, simplemente, porque se trata de entradas en forma de señales acústicas (reconocimiento de sonidos) o de imágenes (reconocimiento de caracteres escritos), en cuyo caso deberán pasar además por un sensor. A continuación se realiza un proceso de extracción de características en el que se analizan las propiedades de los objetos que resultan útiles para la clasificación. El clasificador emplea entonces esas características para asignar al objeto una clase u otra. El procedimiento suele terminar aquí, pero a menudo resulta útil dar un paso más e invocar la presencia de un Post-Procesador que tenga en cuenta otras consideraciones, como el contexto del problema, o el coste de los errores, para adoptar una decisión apropiada. En ese sentido, la medida más simple para evaluar el comportamiento del clasificador, es la tasa de error en la clasificación (el porcentaje de muestras nuevas que son asignadas a la clase equivocada), de modo que lo que se busca, que como hemos visto era el común denominador en los principales problemas de aprendizaje, es minimizar ese error. Lo que le estaremos pidiendo al clasificador es que proporcione una buena *generalización*, esto es, un buen comportamiento ante patrones desconocidos. Sin embargo, sería mucho mejor recomendar acciones que minimicen el coste total esperado, lo que se conoce como *riesgo*, tal y como lo definimos anteriormente.

Aunque un sistema relativamente complejo sea capaz de permitir una clasificación perfecta del conjunto de entrenamiento, puede que no se comporte bien ante las muestras nuevas. Esta situación, la de máquinas demasiado complejas que continúan teniendo errores en el conjunto de test, se conoce con el nombre de *sobrealimentación*. Una de las áreas de investigación más importante en la clasificación estadística de muestras es la de determinar cómo ajustar la complejidad del modelo de modo que no sea tan simple como para no poder explicar las diferencias entre las categorías, ni tan complejo como para dar una clasificación pobre ante patrones noveles.

Veremos a continuación los dos métodos que hemos seleccionado para resolver el problema de selección que teníamos planteado. Son dos de las herramientas más conocidas y utilizadas en este terreno, los Algoritmos Genéticos y los Árboles de Clasificación.

2. ALGORITMOS GENÉTICOS. RESULTADOS

La capacidad del ser humano para predecir el comportamiento de su entorno, se ha ido incrementando con el paso del tiempo. Ha comprendido que, si bien era capaz de controlar muchos aspectos de su vida, y su interacción con lo que le rodeaba, no lo era para otros tantos. La inteligencia artificial es responsable de muchos de esos logros. Los pioneros de esta ciencia estaban tan interesados en la electrónica, como en la biología, y por eso sus aplicaciones iban desde calcular trayectorias de misiles, a tratar de modelizar el cerebro, de imitar el proceso de aprendizaje humano, y de simular la evolución biológica.

Como ya hemos visto, los años 1980 marcan el florecimiento del interés de la comunidad científica por estos temas computacionales inspirados en la biología. Su desarrollo les llevará a cotas inimaginables, primero en el campo de las Redes Neuronales, luego en el del Aprendizaje, y por último en lo que ahora se conoce como "computación evolutiva", de la que los algoritmos genéticos constituyen su máximo exponente.

Por otro lado, la atracción hacia los métodos heurísticos de búsqueda para la *optimización*, ha crecido considerablemente. Uno de los desarrollos más importantes es en la aplicación de los conocidos como Algoritmos Genéticos. A diferencia de otras técnicas también populares como el Templado Simulado o la Búsqueda Tabú, el impacto de los Algoritmos Genéticos, incluso al nivel del gran público, ha sido espectacular [Goldberg 1989].

Inspirados en el proceso de la evolución biológica, los métodos evolutivos emplean la búsqueda estocástica en el diseño de un *clasificador* óptimo. Una ventaja "computacional" de estos métodos es que admiten una implementación natural mediante ordenadores paralelos. Básicamente, proceden del siguiente modo: primero, se crea una *población* (compuesta por varios clasificadores en nuestro caso), cada *individuo* de la misma levemente diferente de los otros; a continuación, se *puntúa* cada individuo en base a su comportamiento en la tarea que han de ejecutar, observando por ejemplo su precisión ante una serie de ejemplos ya etiquetados. En aras de preservar la analogía biológica, al dato resultante se le suene nombrar *capacidad*. A continuación se ordenan los elementos de la población atendiendo a su capacidad, seleccionando los mejores, que representarán una porción de la población total. Es lo que en la teoría de la evolución biológica se conocería como *supervivencia del más capacitado*. En ese momento, se alteran estocásticamente los individuos para crear la siguiente generación (los hijos o *descendencia*). Algunos de ellos tendrán una puntuación más alta que la de sus progenitores, otros la tendrán peor. Por ello, el proceso completo (que comprende los operadores de *reproducción*, *cruce* y *mutación*) se repite para esta la siguiente generación, y para las siguientes, hasta que se logre que un individuo posea una capacidad que supere un valor definido para el criterio que se desea alcanzar.

Este método se vale de variaciones estocásticas, que a su vez dependen de la representación fundamental de cada clasificador (individuo) de la población. Sea cual sea ésta, una propiedad fundamental es que en ocasiones se introducen grandes cambios en el clasificador. Es la presencia de esas alteraciones y de las variaciones aleatorias las que permiten que los métodos evolutivos puedan encontrar buenos clasificadores, incluso en espacios complejos tremendamente discontinuos o "paisajes de capacidad", a donde es difícil dirigirse mediante técnicas clásicas tales como el descenso del gradiente.

Pero, ¿por qué utilizar AG? Hay otras técnicas heurísticas que han cosechado grandes éxitos en el campo de la Optimización, pero que han calado mucho menos en el pensamiento popular que los AG. Seguramente, la causa la podamos encontrar en el poder de seducción del neo-Darwinismo. Sin

embargo, hay que evitar la peligrosa tentación de, iluminados por la luz de la evolución, justificar el uso de los AG sin una base teórica que lo sustente. Los argumentos del tipo "los AG equivalen a la evolución Darviniana, por lo que es claro su potencial para todo tipo de tarea", o, "si la evolución evidentemente funciona, imitarla (aunque se aproximadamente), también lo hará" pueden resultar nefastos. De modo que estas referencias "traídas a mano" a la evolución neo-Darvinista son insuficientes para acreditar la utilización de los AG como herramienta para la optimización. Una analogía más limitada, pero más adecuada, sería el uso de la selección natural en plantas y en experimentos de reproducción con animales, en los que contamos con siglos de experiencia que evidencian la capacidad de los organismos naturales a cambiar como respuesta a las presiones externas (o internas). A pesar de todo, incluso ésta, no hace más que demostrar el potencial de la idea de AG en la resolución de tales problemas. El que realmente los resuelvan está atestiguado por el éxito cosechado en los numerosos experimentos que se han llevado a cabo. [Reeves 2002]

Conocer la evolución histórica que han mantenido estos métodos ayudarán a comprenderlos. Por ello, a continuación, trazaremos el sendero histórico por el que ha discurrido la computación evolutiva, para luego adentrarnos en los entresijos de los algoritmos genéticos, analizando también los aspectos matemáticos de su representación y ejecución. Esto dará paso a los algoritmos genéticos basados en el aprendizaje y la clasificación, y a las aplicaciones más interesantes de la computación evolutiva.

2.1 Breve recorrido histórico por la computación evolutiva

El origen de lo que se conoce como computación evolutiva hay que buscarlo en su razón de ser: la idea de que los conocimientos sobre evolución se pueden aplicar en la resolución de problemas de optimización. Fue en las décadas de 1950 y 1960 cuando varios científicos, de modo independiente, comenzaron a estudiar los sistemas evolutivos, guiados por la intuición de que se podrían emplear como herramienta en problemas de optimización en ingeniería. La idea era "evolucionar" una población de candidatos a ser solución de un problema conocido, utilizando operadores inspirados en la selección natural y la variación genética natural [Mitchell 1998].

Es Rechenberg quien, en la década de 1960 (1965, 1973) introduce las "estrategias evolutivas", método que empleó para optimizar parámetros reales para ciertos dispositivos. La misma idea fue desarrollada posteriormente por Schwefel (1975, 1977). El campo de las estrategias evolutivas ha permanecido como un área de investigación activa, cuyo desarrollo se produce, en su mayor parte, de modo independiente al de los algoritmos genéticos (aunque recientemente se ha visto como las dos comunidades han comenzado ha colaborar). Fogel, Owens y Walsh (1966), fueron los creadores de la "programación evolutiva", una técnica en la cual las candidatas a soluciones a tareas determinadas, eran representadas por máquinas de estados finitos, cuyos diagramas de estados de transición se evolucionaban mediante

mutación aleatoria, seleccionándose el que mejor aproximara. Una formulación más amplia de la programación evolutiva, es un campo de investigación que también continúa en activo (ver, por ejemplo, a [Fogel 1999]). Estas tres áreas, estrategias evolutivas, algoritmos genéticos, y programación evolutiva, son las que forman la columna vertebral de la Computación Evolutiva, y de ellas parten los caminos hacia todos los campos de investigación inspirados en nuestros conocimientos sobre Evolución.

Muchos otros investigadores desarrollaron su trabajo en los algoritmos para la optimización y el aprendizaje inspirados en la evolución, empleando ordenadores digitales para desarrollar simulaciones de los sistemas genéticos. Cabe resaltar nombres como los de Box (1957), Friedman (1959), Bledsoe (1961), Bremermann (1962), y Reed, Toombs y Baricelli (1967). Sin embargo, su trabajo no ha tenido, ni con mucho, la atención que han recibido las estrategias evolutivas, programación evolutiva, y los algoritmos genéticos. Hay que recordar además a los biólogos evolucionistas que han utilizado el ordenador para simular la evolución para realizar experimentos controlados (Baricelli 1957, 1962; Fraser 1957 a,b; Martin y Coreman 1960). El trabajo de Fraser, por ejemplo, aunque dirigido a comprender el fenómeno natural, no era tan distante de la noción moderna de algoritmo genético. A pesar de ello, habría que esperar hasta que la computación electrónica se desarrollara, para poder apreciar la consolidación definitiva de la computación evolutiva [Goldberg 1989].

La primera mención del término “Algoritmos Genético”, y la primera publicación sobre una aplicación del mismo, se deben a Bagley (1967) [Reeves 2001]. En esa época se manifestaba un gran interés por los programas de desarrollo de juegos, y con ese espíritu, Bagley diseñó algoritmos genéticos para buscar conjuntos de parámetros en funciones de evaluación de juegos, y los comparó con los algoritmos de correlación, procedimientos de aprendizaje modelizados después de los algoritmos de pesos variantes de ese periodo (Friedberg, 1958; Samuel, 1959; Uhr y Vossler, 1961). El algoritmo genético de Bagley presentaba insensibilidad frente a la no linealidad del juego y se comportaba bien en un amplio rango de entornos.

El algoritmo genético de Bagley no era muy diferente de los algoritmos genéticos que utilizamos hoy en día. Él construyó operadores de reproducción, cruce y mutación muy similares a los que luego veremos se emplean en un algoritmo genético de tipo general; además, empleó operadores más avanzados, de dominancia e inversión, y usó representaciones en cadenas diploides. Por otro lado, Bagley se valió de alfabetos no binarios para codificar las cadenas. El área en que el trabajo de Bagley ensombrece las más modernas investigaciones es en el de la reproducción y la selección. Él percibía la necesidad de contar con unas tasas apropiadas de selección al comienzo y al término de la ejecución del Algoritmo Genético. Introdujo un mecanismo de gradación de la capacidad que realizaba dos tareas: reducía la selección en los comienzos de la ejecución, con lo que prevenía el dominio de la población por un “súper individuo”, y aumentaba la selección en los momentos finales de la

ejecución, con lo cual mantenía una competición apropiada entre las cadenas similares y con alta capacidad acerca de la convergencia de la población. Los investigadores actuales han adoptado similares procederes.

Bagley también introdujo la primera noción de algoritmo genético auto-regulado en lo que él dio en llamar "controles auto-contenidos". Sugirió la codificación de las probabilidades de cruce y mutación dentro de los propios cromosomas; sin embargo, no aportó ninguna simulación práctica mediante ordenadores con este mecanismo.

Contemporáneo de Bagley en su trabajo fue Rosenberg (1967), quien también investigó los algoritmos genéticos en su temprana tesis doctoral. Debido al énfasis en los aspectos biológicos y de simulación de sus investigaciones, su contribución a los algoritmos genéticos está un poco infravalorada. En sus estudios, Rosenberg simuló una población de organismos unicelulares con una bioquímica simple y todavía rigurosa, una membrana permeable, y una estructura genética simple (un gen, una enzima). A pesar de su interés en poner el acento en lo biológico, el trabajo de Rosenberg fue importante para el consiguiente desarrollo de los algoritmos genéticos en las aplicaciones artificiales, debido a su semejanza con la optimización y la búsqueda de raíces.

Pero es otro científico el considerado creador de los Algoritmos Genéticos [Goldberg 1989]: John Holland, que los desarrolló, junto a sus alumnos y colegas, durante las décadas de 1960 y 1970. En contraste con las estrategias evolutivas y la programación evolutiva, el propósito original de Holland no era diseñar algoritmos para resolver problemas concretos, sino estudiar, de un modo formal, el fenómeno de la adaptación tal y como ocurre en la naturaleza, y desarrollar vías de extrapolar esos mecanismos de adaptación natural a los sistemas computacionales. El libro que Holland escribiera en 1975 *Adaptación en Sistemas Naturales y Artificiales* presentaba el algoritmo genético como una abstracción de la evolución biológica, y proporcionaba el entramado teórico para la adaptación bajo el algoritmo genético. El Algoritmo Genético de Holland era un método para desplazarse, de una población de cromosomas (bits) a una nueva población, utilizando un sistema similar a la "selección natural" junto con los operadores de cruces, mutaciones e inversión inspirados en la genética. En este primitivo algoritmo, cada cromosoma consta de genes (bits), y cada uno de ellos es una muestra de un alelo particular (0 o 1). El operador de selección escoge, entre los cromosomas de la población, aquellos con capacidad de reproducción, y entre éstos, los que sean más "compatibles", producirán más descendencia que el resto. El de cruce extrae partes de dos cadenas, imitando la combinación[26] biológica de dos cromosomas aislados (gametos). La mutación se encarga de cambiar, de modo aleatorio, los valores del alelo en algunas localizaciones del cromosoma; y, por último, la inversión, invierte el orden de una sección contigua del cromosoma, recolocando por tanto el orden en el que se almacenan los genes.

[26] Aquí, como en la mayoría de la literatura sobre algoritmos genéticos, *combinación* equivale a *cruce*.

La mayor innovación de Holland fue la de introducir un algoritmo basado en poblaciones con cruces, mutaciones e inversiones[27]. Es más, Holland fue el primero en intentar colocar la computación evolutiva sobre una base teórica firme (Holland, 1975). Hasta hace poco, esta base teórica, fundamentada en la noción de ***esquema***, fue la estructura sobre la que se edificaron la mayoría de los trabajos teóricos sobre algoritmos genéticos en las décadas siguientes.

En estos últimos años se ha generado una amplia interacción entre los investigadores de varios métodos de computación evolutiva, rompiéndose las fronteras entre algoritmos genéticos, estrategias evolutivas y programación evolutiva. Como consecuencia, en la actualidad, el término "algoritmo genético" se utiliza para designar un concepto mucho más amplio del que concibió Holland.

2.2. Aspectos generales de los Algoritmos Genéticos

- *Concepto*

Los objetivos que perseguían John Holland y sus colegas de la Universidad de Michigan cuando concibieron los algoritmos genéticos, eran fundamentalmente dos: (1) abstraer y explicar rigurosamente el proceso adaptativo de los sistemas naturales, y (2) diseñar sistemas artificiales que retuvieran los mecanismos más importantes de los sistemas naturales. En este sentido, podemos decir que los algoritmos genéticos son

Algoritmos de búsqueda basados en los mecanismos de selección natural y genética natural. Combinan la supervivencia de los más compatibles entre las estructuras de cadenas, con una estructura de información ya aleatorizada, intercambiada para construir un algoritmo de búsqueda con algunas de las capacidades de innovación de la búsqueda humana. [Goldberg 1989].

Básicamente, el Algoritmo Genético funciona como sigue: en cada generación, se crea un conjunto nuevo de "criaturas artificiales" (cadenas) utilizando bits y partes más adecuadas del progenitor. Esto involucra un proceso aleatorio que no es, en absoluto, simple. La novedad que introducen los Algoritmos Genéticos es que explotan eficientemente la información histórica para especular sobre nuevos puntos de búsqueda, esperando un funcionamiento mejorado. El tema central en las investigaciones sobre algoritmos genéticos ha sido la robustez, el equilibrio necesario entre la eficiencia y la eficacia suficiente para la supervivencia en entornos diferentes. Las implicaciones que tiene la robustez en los sistemas artificiales son variadas. Si se puede conseguir que un sistema artificial sea más robusto, se podrán reducir, e incluso eliminar, los

[27] Las estrategias evolutivas de Rechenberg comenzaban con una población de dos individuos, un padre y un descendiente, siendo éste una versión mutada del padre; hasta más tarde no se incorporaron poblaciones de más individuos, ni cruces entre ellos. En cuanto a los programas evolutivos de Fogel, Owens y Walsh, sólo utilizaban mutaciones para producir variaciones.

costes por rediseños. Y si se es capaz de lograr niveles altos de adaptación, los sistemas podrán desarrollar sus funciones mejor y durante más tiempo. Sin embargo, ante la robustez, eficiencia y flexibilidad de los sistemas biológicos, sólo podemos sentarnos a contemplar, y maravillarnos; mentiríamos si dijéramos que somos capaces de igualarlos.

Pero, ¿por qué basarse en nuestros conocimientos sobre la evolución biológica? La respuesta la encontramos si observamos una constante que se repite en muchos problemas: la búsqueda de soluciones entre una cantidad ingente de candidatos. Tómese, por ejemplo, el cálculo de un conjunto de reglas (ecuaciones) capaz de regir las subidas y bajadas de un mercado financiero. El modo de llegar a la mejor solución en estas situaciones, pasa por ser capaz de obtener rendimiento de un uso eficaz del paralelismo, que permita explorar diferentes posibilidades de modo simultáneo. Para ello, se precisa, tanto paralelismo computacional (contar con varios procesadores computando al mismo tiempo), como una estrategia adecuada de búsqueda. Por otro lado, como ya hemos ido analizando a lo largo de este Trabajo, muchos problemas computacionales, precisan de un programa adaptativo, capaz de comportarse bien ante cambios en el entorno. Además, la mayoría de estos problemas tienen soluciones complejas, muy difíciles de programar a mano. Entre las técnicas que han surgido al tratar de resolver estas cuestiones, nos hemos familiarizado ya con el conexionismo, y nos encontramos ahora con la "computación evolutiva". Si en la primera las reglas pasan por umbrales neuronales, propagación de la activación, y refuerzo o no de las conexiones., en la segunda son la selección natural, con variaciones debidas a cruces y/o mutaciones, y su objetivo es el diseño de soluciones de alta calidad para problemas de elevado grado de complejidad, y la habilidad de adaptar esas soluciones de cara a cambios en el entorno.

La evolución, tal y como la conocemos, es básicamente un método de búsqueda entre un número enorme de posibles "soluciones". En biología las posibilidades están formadas por un conjunto de secuencias genéticas posibles, y las soluciones deseadas, por organismos capaces de sobrevivir y reproducirse en sus entornos. La evolución puede verse, asimismo, como un modo de "diseñar" soluciones a problemas complejos, con la capacidad de innovar. Estos son los motivos de que los mecanismos evolutivos sean una fuente de inspiración para los algoritmos de búsqueda. Por supuesto, el buen funcionamiento de un organismo biológico depende de muchos criterios, que además varían a medida que el organismo evoluciona, de modo que la evolución está "buscando" continuamente entre un conjunto cambiante de posibilidades. Por ello, podemos considerarla como un método de búsqueda masivamente paralelo, ya que evalúa y cambia millones de especies en paralelo. Para terminar, las reglas de la evolución, aunque de alto nivel, son simples: las especies evolucionan mediante variaciones aleatorias (vía mutaciones, recombinaciones, etc.) seguidas por la selección natural, donde el mejor tiende a sobrevivir y reproducirse, propagando así su material genético a posteriores generaciones.

- *La analogía biológica*

Todos los organismos que conocemos están compuestos por una o más células, cada una de las cuales contiene a su vez uno o más *cromosomas* (esto es, cadenas de ADN), que tienen la función de ser una especie de "anteproyecto" del organismo del que forman parte. Un cromosoma se puede dividir, conceptualmente, en *genes,* bloques funcionales de ADN que codifican una determinada proteína. Solemos pensar en los genes, aunque en una visión muy superficial, como los responsables de determinar los rasgos del individuo, tales como el color de los ojos, o del cabello. Las diferentes posibilidades de escoger un rasgo (ojos azules, marrones o verdes, por ejemplo) reciben el nombre de *alelos*. Cada gen está localizado en una determinada posición (lugar o *locus*) dentro del cromosoma que integra. En genética, la posición de un gene se identifica independientemente de la función de dicho gene. Podemos hablar así del gene "color de ojos", su posición, 10, y el valor de su alelo, azul.

Muchos organismos tienen varios cromosomas en cada célula. El *genoma* del organismo es la colección completa del material genético. Lo que se conoce como *genotipo*, es el conjunto de genes contenido en un genoma. El genotipo dará lugar, tras el desarrollo fetal, al *fenotipo* del organismo, esto es, a sus características físicas y mentales, tales como el color de ojos, la estatura, o la inteligencia.

La mayoría de las especies reproductoras sexualmente que habitan nuestro planeta, almacenan sus cromosomas por parejas (son *diploides*; se llaman haploides en caso contrario). En el caso del ser humano, cada célula somática (no germen) de su cuerpo contiene 23 pares de cromosomas. Durante la reproducción sexual se produce una recombinación o *cruce***:** en cada padre, se intercambian los genes entre cada par de cromosomas, para formar un gameto (un cromosoma único), y entonces, los gametos de los dos padres se emparejan para constituir un conjunto completo de cromosomas diploides. En el caso de la reproducción haploide, los genes se intercambian entre dos padres con una sola rama de cromosomas. La descendencia está sujeta a *mutaciones***,** donde se produce un cambio en algún nucleótido (bits elementales de ADN) de padre a hijo; esas modificaciones son resultado habitualmente de "errores de copia". La *capacidad* del organismo se define como la probabilidad de que el organismo viva para reproducirse (viabilidad) o como una función del número de descendencia que tenga ese organismo (fertilidad).

Traslademos estos conceptos a la "vida artificial": en los algoritmos genéticos, el término cromosoma se refiere a un candidato a solución del problema, que a menudo se codifica como una *cadena* de bits. Los genes son tanto un bit como bloques cortos de bits adyacentes que codifican un elemento particular del candidato a solución (por ejemplo, en el caso de la optimización de una función multiparamétrica, los bits que codifican un parámetro particular, se considera un gene). Un alelo en una cadena de bits será el valor que puede tomar, esto es, un 0 o un 1 (para alfabetos largos cada lugar puede tener más alelos).

Es importante observar que en el caso artificial no hemos distinguido entre un gene (un carácter particular) y su locus (posición); la posición de un bit en la cadena determina su significado (el modo en que se descodifica) de manera uniforme a lo largo de la población y del tiempo. Sin embargo, cabe señalar que existen estructuras más avanzadas que permiten tratar el gene y su posición de forma separada.

Por último, existe otra expresión genética que también tiene su traducción en los algoritmos genéticos. Se trata de la *epistaxis*, o sustitución de una expresión genética por otros genes. En un algoritmo genético se emplearan las no linealidades para simularla.

Detengámonos ahora brevemente los procesos genéticos básicos. La *reproducción o selección* se realizará en función de la capacidad de cada cadena, que vendrá medida por una función *f* que indicará la utilidad, bondad, o beneficio que se trata de maximizar en el proceso. Estudiaremos que este operador no es más que la versión artificial de la selección natural, la supervivencia Darviniana del mejor entre las criaturas de cadenas. El cruce consiste normalmente, como en su espejo biológico, en un intercambio de material genético entre dos cromosomas de dos padres haploides, esto es, dos cadenas de bits intercambiaran sus caracteres a partir de un lugar de la cadena seleccionado aleatoriamente, para dar lugar a dos nuevas cadenas. La mutación será una permutación en un bit en un lugar aleatorio (o, en el caso de alfabetos largos, remplazar el símbolo de un lugar escogido aleatoriamente con un símbolo nuevo escogido también aleatoriamente).

En la mayoría de las aplicaciones de los algoritmos genéticos nos encontraremos con individuos haploides, concretamente, de cromosomas únicos. El genotipo de un individuo en un algoritmo genético que emplea cadenas de bits es, simplemente, la configuración de bits del cromosoma de ese individuo. Si en los sistemas naturales el genotipo es el paquete genético completo, en los artificiales el paquete total de cadenas es la *estructura*. La noción de fenotipo no aparece en el contexto de los algoritmos genéticos, aunque sí se habla de que las estructuras se decodifican para formar un particular *conjunto de parámetros***,** *solución alternativa* **o** *punto* (en el espacio de soluciones). Avances recientes en la materia trabajan con algoritmos que poseen un nivel “genotípico” y uno “fenotípico” (por ejemplo, la cadena de bits que codifica una red neuronal, y la red en sí misma).

De este modo, es sencillo imaginar cada cromosoma de un algoritmo genético como un punto en el espacio de búsqueda de candidatos a soluciones. El algoritmo genético procesa poblaciones de cromosomas, remplazando sucesivamente cada población por otra. Como veíamos antes, para tratar la reproducción el algoritmo suele requerir una función de capacidad o potencial que asigna una puntuación (la capacidad) a cada cromosoma de la población actual. La capacidad o el potencial de un cromosoma depende de cómo

resuelva ese cromosoma el problema a tratar[28]. Un concepto relacionado es el del "paisaje del potencial". Definido originalmente por Sewell Wright (1931) en el contexto de la genética de poblaciones, el paisaje de un potencial es la representación del espacio de todos los posibles genotipos junto con sus capacidades.
De este modo, los operadores de cruce y mutación pueden tratarse como modos de mover una población en el paisaje definido por su función de capacidad, y un algoritmo genético como un método de búsqueda de ese paisaje para cadenas altamente cualificadas[29].

2.3 Elementos para diseñar un Algoritmo Genético

- *La Población*

Población inicial

Una de las cuestiones esenciales a la hora de diseñar un AG es determinar el tamaño de la población. Es este un aspecto que ha sido estudiado desde varios puntos de vista teóricos, aunque todos comparten la misma idea de fondo: el equilibrio entre la eficiencia y la efectividad. Intuitivamente, está claro que debe existir algún valor "óptimo" entre un tamaño demasiado grande, que no resultaría eficiente en tiempo de computación, y uno demasiado pequeño, que no nos permitiría explorar el espacio de búsqueda con efectividad. Goldberg fue probablemente el primero en tratar de dar una respuesta a esta cuestión, empleando la idea de esquema, que se desarrollará brevemente en el siguiente punto. Desgraciadamente, desde ese planteamiento, parece que el tamaño de la población crece exponencialmente a la longitud de las cadenas.

Afortunadamente, los resultados empíricos obtenidos por varios autores (ver [Grefenstette 1986] o [Schaffer et al 1989]) sugieren que 30 es un tamaño que funciona bien en la mayoría de los casos. Un análisis posterior desde una perspectiva diferente condujo a Goldberg y a sus colegas [Goldberg et al 1992] a la conclusión de que lo indicado era una dependencia lineal entre el tamaño de la población y la longitud de la cadena. Sin embargo, aunque sí es cierto que las poblaciones deben aumentar en relación a la longitud de las cadenas, incluso una tasa lineal es demasiado elevada para la mayoría de las poblaciones. También podríamos atacar la cuestión desde el punto de vista

[28] Como ejemplo, si se trata de maximizar la función $f(y) = y + |sen(32y)|, 0 \le y < \pi$ (Riolo, 1992), los candidatos a soluciones son valores de *y,* que se codifican como cadenas de bits que representan números reales. El cálculo de la capacidad traslada una cadena de bits dadas, *x,* en un n º real, *y,* y se evalúa entonces la función en ese valor. La capacidad de una cadena es, pues, el valor de la función en ese punto.

[29] Esta idea de evolución como movimientos de poblaciones a lo largo de paisajes invariables, no es realista biológicamente. Por ejemplo, a un organismo no se le puede asignar un valor potencial independientemente de otro organismo de su entorno; a medida que la población cambia, la capacidad de un genotipo particular cambia también.

contrario, determinando, o tratando de hacerlo, cuál ha de ser el número mínimo de individuos necesario para que se desarrolle una búsqueda que resulte significativa. En [Reeves (ED) 1993] se adopta el principio de que cada punto del espacio de búsqueda debe alcanzarse a partir de la población inicial utilizando únicamente el cruce. Esto sólo podría verificarse si existe al menos un ejemplo de cada alelo en cada locus de la población completa de cadenas. La probabilidad de que eso ocurra puede determinarse, y al hacerlo, aparece de nuevo 30 como un número de individuos adecuado para que esa probabilidad sea alta. También se llega a otra conclusión: que se precisa una población desproporcionadamente grande para alfabetos con cardinales altos.

Respecto a cómo elegir la población inicial, se asume que se realiza mediante un proceso aleatorio, lo que en la práctica equivale a utilizar secuencias pseudo-aleatorias. Sin embargo, los puntos así elegidos no recubren necesariamente el espacio de forma uniforme, lo que si se logra con métodos estadísticos sofisticados, que son además más adecuados para alfabetos no binarios. Una idea que funciona muy bien y es relativamente sencilla es la de insertar directamente los alelos que necesitemos, en vez de confiar en que alguna vez los generaremos.

En este sentido, cabe también contemplar la posibilidad de sembrar la población inicial con soluciones que sabemos que son buenas. Algunas investigaciones demuestran que incluir soluciones de alta calidad, localizadas mediante técnicas heurísticas, puede ayudar al AG a encontrar soluciones mejores bastante más rápidamente que comenzando aleatoriamente. Sin embargo, existe también la posibilidad de inducir una convergencia prematura hacia una solución pobre. En general, se demuestra que este "sembrado" artificial de soluciones tiende a reducir la calidad de la mejor solución encontrada.

Nueva Población

El algoritmo original de Holland asumía una estrategia de remplazamiento "generacional", esto es, se aplican mutación y cruce a una población de *N* cromosomas hasta que se genera un conjunto nuevo de otros *N* individuos, que será la nueva población. Desde el punto de vista de la optimización, no parece una táctica muy eficiente, el obtener una buena solución para luego correr el riesgo de perderla, impidiendo así que tome parte en una reproducción posterior. Es esta la razón que llevó a De Jong [De Jong 1975] a introducir los conceptos de *elitismo* **y** *superposición de población*. Se trata de ideas sencillas: una estrategia elitista asegura la supervivencia de los mejores individuos además de preservarlos y sustituir sólo los (*N*-1) miembros restantes de la población por nuevas cadenas. El solapamiento de poblaciones conlleva una etapa más en la que se remplaza sólo una fracción de la población en cada generación. Finalmente, llevándolo a su conclusión lógica, produce el conocido como estado estable, en el que sólo se crea un cromosomas (a veces un par) en cada etapa. Davis [Davis (Ed) 1991] proporciona una introducción general bastante buena de este tipo de AG.

Otro aspecto relevante de la generación de una nueva población es el del lugar de mantenimiento de la diversidad (esta noción se tratará en el punto siguiente). Aunque se trata de una cuestión que se suele considerar secundaria, puede colocarse en primer lugar empleando los conceptos de "aglomeración" y "niching" (no tiene traducción, viene de *niche* que en inglés significa nicho o hueco), que de nuevo están tomados de sus análogos biológicos: un nicho en el entorno natural es un conjunto de condiciones a las que un grupo específico de fenotipos (especies) está especialmente bien adaptado. En un AG se trata un nicho como un subconjunto de cromosomas que son similares en algún sentido. La idea sería que un cromosoma recién generado sustituiría a otro de su propio nicho, más que potencialmente a otro de la población en general. De Jong desarrolló un "facto aglomerante" para utilizar en estos casos, un entero que definía el tamaño de un subconjunto (escogido aleatoriamente) de la población existente, con el que se comparaba el nuevo cromosoma que habíamos generado. El más cercano era el que se seleccionaba para eliminar. Existen otras formas de generación de *nichos*, la más conocida de las cuales es el *intercambio* [Davis (Ed) 1991], en el que se define una función de intercambio sobre toda la población que se emplea para modificar la capacidad de cada cromosoma.

- *Convergencia*

Un punto importante a la hora de diseñar un AG es el de establecer un criterio de parada adecuado. A diferencia de los métodos de búsqueda por proximidad más sencillos que terminan cuando se alcanza un óptimo local, los AG son métodos estocásticos que, en principio, podrían ejecutarse para siempre. En la práctica, como decimos, se necesita un criterio de terminación. Lo normal es establecer un límite sobre el número de evaluaciones sobre la capacidad, o simplemente, sobre el tiempo de computación, o rastrear la diversidad de la población y parara cuando caiga por debajo de un umbral dado. A pesar de todo, el significado de la diversidad no siempre resulta obvio, y puede relacionarse tanto con los genotipos como con los fenotipos, o incluso, con las capacidades, pero en ningún caso necesitamos dotarlo de un significado estadístico. Por ejemplo, podríamos decidir terminar una ejecución si la proporción en cada locus de un alelo en particular supera el 90 %.

- Mantener la diversidad

Como el propio Holland decía: "…la verdadera esencia de un buen diseño de un AG es el retener la diversidad" *[Holland 2000].*

El efecto de la selección es reducir la diversidad, y algunos métodos pueden hacerlo además con mucha rapidez. Para mitigarlo, podemos considerar poblaciones grandes, o tasas de mutación mayores, o utilizando el método que mencionábamos en el punto anterior de "crear nichos y aglomerando". Todos ellos tienen una base" natural" en la que se fundamentan para su desarrollo.Otra aproximación bastante popular consiste en utilizar una política

de “no duplicar”, lo cual equivale a que no se permite la descendencia en la población si son clones de individuos ya existentes. Subyace, evidentemente, la necesidad de comparar cada individuo actual con el nuevo candidato, lo que añade un esfuerzo computacional, que puede ser muy significativo con poblaciones grandes.

También existe la posibilidad de reducir la ocurrencia de clones antes de que se genere la descendencia. Brooker [Brooker 1987] propuso que, antes de aplicar el cruce, se examinasen los progenitores seleccionados en busca de puntos de cruce adecuados (por ejemplo, 1101001 y 1100010 pueden engendrar clones si el punto de cruce fuera cualquiera de las tres primeras posiciones).

2.4 Operadores

Según hemos observado, podemos considerar que los algoritmos genéticos tienen, al menos, estos elementos en común: poblaciones de cromosomas, selección en base a su capacidad, cruces para producir descendencia nueva, y mutación aleatoria de la nueva descendencia. La inversión - el cuarto elemento de los algoritmos genéticos tal y como los concibió Holland - se usa raramente en las implementaciones actuales, y sus ventajas, si las tiene, no están del todo establecidas.

Veamos a continuación más detenidamente estos tres operadores básicos.

- *Selección o reproducción*

Este operador escoge cromosomas entre la población para efectuar la reproducción. Cuanto más capaz sea el cromosoma, más veces será seleccionado para reproducirse. La idea esencial en la reproducción, es que debe estar relacionada a la capacidad, y por ello, un método de selección muy común es el de la *capacidad-proporcionada*, en el que el número de veces esperado para que se reproduzca un individuo es igual a su capacidad dividida por la capacidad media de la población (es equivalente a lo que los biólogos llaman “selección viable”). El mecanismo habitual para implementarla es comúnmente conocido como método de la ruleta [Goldberg 1989a], que conceptualmente, es equivalente a asignar a cada individuo una porción de una ruleta circular de área igual a su capacidad individual. Se gira la ruleta, y se selecciona el individuo al que corresponde la región en la que “cae” la bola. De este modo, se emplea una distribución de probabilidad en la que la probabilidad de selección para una cadena dada es directamente proporcional a su capacidad. Se utiliza un número pseudo-aleatorio cada vez para escoger cadenas para la paternidad.

El problema de este método es que presenta una variabilidad estocástica alta, y que el número de veces que N_C el cromosoma C resulta seleccionado en cada generación puede ser diferente de su valor esperado $E[N_C]$

Ese es el motivo por el que debe utilizarse un muestreo sin remplazamiento , para asegurarnos así que al menos se alcanza la parte integral de $E[N_C]$

En la práctica se suele aplicar lo que se conoce como selección universal estocástica de Baker. En cada etapa, en vez de una selección única, suponemos que la ruleta tiene varios brazos que la permiten girar simultáneamente a intervalos regulares. Al girar la ruleta, se producen en el mismo instante los valores N_C para todos los cromosomas de la población. Desde el punto de vista de la teoría estadística de muestreo, se corresponde con un muestreo sistemático aleatorio.

El trabajo experimental realizado por Hancock [30] [Hancock 1996] demuestra rotundamente la superioridad de este otro enfoque, aunque son muchas las publicaciones sobre aplicaciones de los AG en los que todavía aparece el método de la ruleta como principal.

Por último, necesitamos determinar qué sucedería si intentamos emparejar inadvertidamente una cadena consigo misma. Para poblaciones y cadenas de longitudes razonablemente grandes, este no suele ser un problema en las primeras etapas. Sin embargo, a medida que la población converge, el seleccionar una cadena para crear descendencia consigo misma o con un clon, puede ser más normal de lo que parece. Podemos tomar este hecho como una señal de terminación, o adoptar una política de “no duplicar” para prevenirlo.

Otro problema asociado es el de encontrar una medida adecuada de la capacidad para los miembros de la población. Valernos de la función objetivo $f(x)$ únicamente no suele ser suficiente, puesto que la escala en que esté medida $f(x)$ es importante (por ejemplo, es más fácil distinguir valores de 10 y 20 que 1010 y 1020). Es más, si resulta que el objetivo que perseguimos es de minimizar en vez de maximizar, se requerirá una transformación.

A menudo se aplica por tanto algún tipo de escalado, y Goldberg facilita un algoritmo sencillo que permite manejar tanto maximización como minimización, pero resulta bastante pesado, y exige un re-escalado continuo según avanza la búsqueda. Hay otras alternativas que proporcionan soluciones más elegantes, como es el de establecer un ranking de los cromosomas según su capacidad. Aunque se pierde algo de información, no hay necesidad de re-escale y el algoritmo de selección es más simple y eficiente. Otra posibilidad a esta selección proporcional estrictamente a la capacidad es una selección “por torneo”, en la que se elige un conjunto de cromosomas y se compara, y el que resulte elegido como ganador se seleccionará como padre. Una ventaja potencial de este método es que, como sólo precisa un orden de preferencia entre pares o grupos de cadenas, puede afrontar situaciones en las que no

[30] En este artículo aparece un análisis experimental sobre varios métodos de selección.

exista una función objetivo formal. A pesar de ello, conviene señalar que tiene el mismo problema que la selección de la ruleta en cuanto a que está sujeta a efectos estocásticos arbitrarios.

- *Cruce o recombinación*

Una vez seleccionados los padres, necesitamos recombinarlos, y el proceso para realizarlo se conoce como cruce. Se trata de un operador cuya labor es elegir un lugar (punto de cruce, que se toma aleatoriamente), y cambiar las secuencias antes y después de esa posición entre los dos cromosomas padres, para crear nueva descendencia (por ejemplo, las cadenas 10010011 y 11111010 pueden cruzarse después del tercer lugar para producir la descendencia 10011010 y 11110011). Imita la recombinación biológica entre dos organismos haploides.

El descrito hasta ahora es el llamado cruce en 1 punto, pero de modo similar se puede definir el cruce en m puntos, con m>1. Eshelman et al. [Eshelman et al 1989]. Desarrollaron una investigación bastante temprana y profunda sobre este último tipo de recombinación, examinando el efecto del cruce tradicional en un punto, y considerando otras alternativas. Su argumento central consistía en afirmar que existen dos fuentes de sesgo a explotar en un AG: el posicional, y el de distribución. El cruce simple tiene un sesgo posicional considerable, ya que favorece subcadenas de bits contiguos, y no podemos estar seguros de si ese sesgo está actuando a favor de las soluciones buenas.

Por otro lado, el cruce en un punto carece del sesgo de distribución, ya que el punto de cruce se escoge aleatoriamente mediante una distribución uniforme. Sin embargo, la falta de este sesgo no es necesariamente algo bueno, puesto que limita el intercambio de información entre los progenitores. En la obra de Eshelman et al se examina la posibilidad de cambiar esos sesgos, en particular empleando la recombinación m-puntos, y los resultados empíricos demuestran que el cruce de 1 punto no es la mejor opción. Parece incluso probarse que el de 8-puntos es el que mejor comportamiento presenta, en cuanto al número de evaluaciones de la función precisos para llegar al óptimo global.

Otra alternativa bastante obvia, y que elimina cualquier sesgo, es realizar un proceso de cruce completamente aleatorio, el conocido como cruce uniforme. Esto es algo muy sencillo de ejecutar, especialmente si observamos que la manera más fácil de implementar el operador cruce es escribirlo como una cadena binaria, o máscara. Por ejemplo, la máscara 1100011 representa un cruce de dos puntos, que aplicado a las cadenas (1100110) y (0001101) generaría como descendiente a (1101110). Generando la muestra de 0s y 1s estocásticamente (usando una distribución de Bernouilli) obtendríamos el cruce uniforme, que podría proporcionar una máscara como (1010001).

Hay que observar que hasta ahora hemos tomado el operador como lineal, lo que tiene sentido si manejamos cadenas binarias. Pero, cuando la representación es no lineal, debe reinterpretarse la recombinación. Uno de los

problemas más habituales aparece cuando el espacio de soluciones es el de las Permutaciones Π_l, lo que ocurre por ejemplo en problemas de planificación, y en el famoso problema del viajante.

Esta situación de la necesidad de diferentes formas de cruces para representaciones distintas condujo a Radcliffe y Surry [Radcliffe y Surry 1995] a proponer una forma generalizada de cruce, que sirva para cualquier representación:

Elegir un ente aleatorio *n* € $[1,., l\text{-}1]$
Escoger *n* puntos de cruce
Genera una permutación aleatoria € (1..*n*+1) para el orden *n* segmentario
Designar un padre para copiar $k \leftarrow 1$

Repetir:
Copiar todos los alelos compatibles del segmento σ_k para el padre escogido
Intercambiar los padres designados: $k \leftarrow k+1$

Hasta $k = n+1$

Si el descendiente está incompleto, entonces insertar alelo(-s) legal en la posición necesaria utilizando el *voto de calidad* aleatorio, si es necesario. Este tipo de operador tiene el mérito de que construye un entramado general para la recombinación, más que confiar en las ideas *ad hoc*. Se puede generar un segundo descendiente revirtiendo el orden de los padres, y también se puede generalizar para el caso de más de dos padres.

Por último, conviene aclarar que el cruce no es un operador que haya que emplear siempre, una idea muy extendida, y comprensible por otra parte, dado el énfasis que Holland en su trabajo original puso sobre la recombinación. Por tanto, tan válida como la estrategia de cruce-AND-mutación para generar una nueva descendencia, es la de cruce-OR-mutación. Existen numerosos ejemplos de ambas en la literatura. La primera conlleva cruce (posiblemente con probabilidad menor que 1), y luego intentar efectuar mutación sobre la descendencia (una o las dos). Es incluso posible en este caso que no se aplique ni cruce ni mutación, de modo que la descendencia simplemente clone a sus padres. La implementación de la segunda estrategia prefiere hacer siempre algo, ya sea mutación o cruce, pero nunca las dos.

El mecanismo para implementar esas elecciones en términos probabilísticas es usualmente una regla aleatoria, donde la operación se lleva a cabo si una desviación uniforme pseudo-aleatoria cae por debajo de un cierto umbral. En el caso del cruce, es lo que a menudo se conoce con el nombre de tasa de cruce. En el de la mutación tenemos la posibilidad de elegir entre determinar el número de mutaciones por cadena o por gen., lo que proporcionará una tasa de mutación en cada caso.

En la estrategia -OR-existe además la posibilidad de modificar las proporciones relativas de cruce y mutación a medida que avanza la búsqueda. Davis argumenta en [Davis (Ed) 1991] que son apropiadas tasas diferentes a distintos tiempos: alta para el cruce al comienzo, alta para la mutación según converge la población. De hecho, sugiere que las proporciones de cada operador se pueden adaptar online, a su record de búsqueda de nuevos cromosomas de alta calidad.

Para terminar, existen otras consideraciones prácticas que influyen en la implementación del cruce. Por ejemplo, ¿con cuánta frecuencia lo aplicamos? Algunos lo aplican siempre, otros nunca, y otros lo hacen de un modo estocástico, utilizándolo con una probabilidad $\chi < 1$. Además, ¿generamos uno o dos descendientes? En muchos casos, se crean "gemelos" naturales como descendientes, aunque en los más sofisticados es posible que emerja un solo descendiente. Si escogemos sólo uno de los dos, ¿cómo lo hacemos?.

De acuerdo con la naturaleza estocástica de los AG, podemos elegir uno de los descendientes de modo aleatorio, o alternativamente, podemos sesgar la decisión usando alguna otra propiedad como la capacidad relativa de los nuevos individuos, o la pérdida en diversidad inherente a elegir a uno en vez de al otro.

- *Mutación*

Este operador (si se utiliza completamente, en otras palabras, si estamos en el caso de cruce-OR-mutación) produce variaciones de modo aleatorio en un cromosoma (por ejemplo, la cadena 00011100 puede mutar su segunda posición para dar lugar a la cadena 01011100). La mutación puede darse en cada posición de un bit en una cadena, con una probabilidad, normalmente muy pequeña (por ejemplo 0.001).

Como ocurría en el cruce, es posible representar este operador en forma de cadena, de manera que, por ejemplo, 0010010 (cadena generada por una distribución de Bernouilli, lo que refuerza la aseveración de que la probabilidad o tasa de mutación es pequeña), indicaría que se les asignaría a los genes 3º y 6º nuevo alelo.

A pesar de la simplicidad de la idea, hay diferentes modos de implementarla, que conllevan diferencias sustanciales en lo que al funcionamiento del AG se refiere. El más simple parece el establecer un nº aleatorio para todos los genes de la cadena, lo cual, evidentemente, es potencialmente caro computacionalmente hablando, especialmente si las cadenas son largas y la población grande. Una alternativa bastante más eficiente es extraer una variable aleatoria de una distribución de Poisson con parámetro λ, donde λ es el número medio de mutaciones por cromosomas. Como valor para λ se suele tomar 1, lo que significa que si la longitud de la cadena es *l*, la tasa de mutación por gen es $\mu = 1/l$.

Si hubiéramos fijado, por ejemplo, la cantidad de mutaciones en *m*, extraeríamos *m* números aleatorios (sin remplazamiento) uniformemente distribuidos entre 1 y *l* para determinar el *loci* donde tendrá lugar.

En el caso de las cadenas binarias, la mutación se traduce sencillamente en complementar los bits elegidos. Por ejemplo, si aplicamos la mutación a los genes 2 y 5 de la cadena 0110101 quedaría 0010001
Si tratamos con codificación *q*-área, el proceso es un poco más delicado. Al haber varios valores posibles de alelos para cada gen, la mutación se complica; si decidimos cambiar un alelo particular, debemos explicar cuál va a ser el nuevo valor, y por qué. Podría tratarse entonces de una elección estocástica, pero si hubiera algún tipo de relación ordinal entre los valores de los alelos, podría resultar más sensible restringir la elección a los alelos cercanos al valor actual, o al menos, sesgar la distribución de probabilidad en su favor.

Se ha sugerido a menudo que la mutación tiene una función secundaria, la de ayudar a preservar un nivel razonable de diversidad en la población (escapando así a las regiones sub-óptimas del espacio de soluciones), pero no todos los autores están de acuerdo.

Algunos autores sugieren una especie de mutación adaptativa: por ejemplo, Fogarty ([Fogarty 1989]) experimentó con tasas de mutación diferentes en loci distintos, y Reeves ([Reeves 1995]) varió la probabilidad de mutación de acuerdo con la diversidad de la población (medida en términos del coeficiente de la variación de la capacidad). Existen procedimientos más sofisticados, y la evidencia sugiere que los investigadores utilizan tasas de mutación variables como una política de conservación de la diversidad.

- *Inversión*

La presentación original de Holland incluía, además de los tres operadores anteriores, considerados como básicos en un AG general, un operador que el denominó inversión. El propósito del mismo era no introducir nuevo material, como en la mutación, ni recombinar material genético diferente, como e el cruce, sino realizar una reordenación, en una cadena individual, con la intención de alterar la proximidad de determinados genes antes del cruce. Para hacerlo posible, el locus de cada gen se debe codificar explícitamente con el alelo de cada locus. Lógicamente, esto implica que existen muchos modos de codificar una cadena en particular. Por ejemplo, las dos cadenas siguientes son en realidad la misma (la primera fila indica el locus, y la segunda el alelo):

1234567	2674153
1001001	0011100

Sin embargo, cuando se aplica el cruce de 1 punto a la segunda cadena, el grupo de 1s del centro es menos propicio a romperse que si estuvieran al comienzo. Existe una dificultad evidente, de todos modos: ¿qué ocurriría si el orden de los dos cromosomas progenitores fuera diferente?. Para afrontarla, deben alinearse los cromosomas de tal forma que el orden sea el mismo en

cada caso. Esta idea y este propósito están fuertemente ligados a la idea de *esquema* de Holland, que luego se verá.

2.5 Un ejemplo simple de Algoritmo Genético

Sea X el problema a resolver. Dada una representación de candidatas a soluciones en una cadena de bits, un algoritmo genético simple, tal y como se describe en [Mitchell M. 1998], trabajaría del siguiente modo:

1. Comenzar con una población P generada aleatoriamente compuesta de *n* cromosomas de l bit.

2. Calcular la capacidad *f*(*x*) para cada cromosoma *x* de P.

3. Repetir los siguientes pasos hasta que se hayan creado *n* descendientes.

4. Seleccionar un par de cromosomas padre de P, siendo la probabilidad de selección una función creciente de la capacidad. La selección se realiza "con remplazamiento", es decir, que el mismo cromosoma puede ser seleccionado en más de una ocasión para ser padre.

5. Con probabilidad P_C (probabilidad de cruce, o tasa de cruce), cruzar el par en un punto elegido aleatoriamente (con probabilidad uniforme) para formar dos descendientes. Si no tiene lugar ningún cruce, formar dos descendientes que sean copias exactas de sus respectivos padres. (Obsérvese que aquí la probabilidad de cruce se define como la probabilidad de que dos padres se crucen sobre un único punto. Hay otras versiones de algoritmos genéticos que son de "cruces en múltiples puntos", en los que la tasa de cruce para una pareja de padres es el n º de puntos en los que tiene lugar un cruce).

6. Mutar los dos descendientes en cada lugar con probabilidad P_M (probabilidad de mutación, o tasa de mutación), y colocar los cromosomas resultantes en la nueva población P'.

7. Si *n* es impar, se puede rechazar aleatoriamente a un miembro de la nueva población.

8. Remplazar la población actual P con la nueva P'

9. Volver al paso 2

Cada iteración del proceso recibe el nombre de *generación*. Lo usual es iterar el algoritmo de 50 a 500 o más veces. El conjunto completo de generaciones

se llama *serie*[31]. Al concluir una serie, a menudo se encuentran entre la población uno o más cromosomas con elevada capacidad. Como la aleatoriedad juega un importante papel en cada serie, dos series con diferentes números aleatorios en el origen darán lugar generalmente a comportamientos diferentes. Los investigadores en algoritmos genéticos acostumbran a reportar estadísticas (tales como el mejor potencial hallado en una serie, y la generación donde se encontró el individuo con la mejor capacidad, etc.) evaluadas sobre muchas series diferentes del algoritmo genético sobre el mismo problema.

Así, tras la reproducción, el cruce y la mutación de la población original, la nueva población está lista para ser evaluada. Para ello, basta con decodificar las cadenas nuevas creadas por este algoritmo genético sencillo, y calcular la función de capacidad sobre ellas. Ya con esta primera generación se puede observar cómo ha mejorado la capacidad de la población de manera asombrosa. Pero, ¿a qué es debido este buen funcionamiento?. Ciertamente, la aleatoriedad del proceso ayuda a la efectividad de la búsqueda de soluciones, pero no es la única. Para descubrir la respuesta, deberemos examinar los datos disponibles para cada procedimiento de búsqueda, y descubriremos que ésta es tanto más efectiva cuanto más explotemos las similitudes que yacen en la codificación que estemos empleando. Esto conduce a la noción de *esquema o plantilla de similitudes*, que a su vez lleva a la conocida como "hipótesis del bloque de construcción[32]".

La cuestión fundamental parece ser la siguiente: en un proceso de búsqueda en el que se proporcionan únicamente los valores sobre la capacidad, ¿qué información está contenida en la población de cadenas y en los valores de la función objetivo que ayuden a guiar una búsqueda dirigida que redunde en una mejora del proceso?. Nada hay a simple vista que nos indique el camino a seguir; pero, si contempláramos los datos y su capacidad, percibiríamos cierta similitud entre las cadenas. Si exploráramos esas semejanzas con más profundidad, veríamos que ciertas muestras de las cadenas parecen estar asociadas con el buen comportamiento. Por ejemplo, pudiera ser que las cadenas que empezaran por 1 tuvieran una capacidad más elevada que el resto. Estaríamos por tanto realizando dos labores en nuestro objetivo de mejorar la búsqueda: buscando semejanzas entre las cadenas de la población, y localizando relaciones casuales entre esas similitudes y la elevada capacidad. No estamos ya interesados en las cadenas como individuos aislados. Nos preguntamos cómo una cadena puede parecerse a sus cadenas parientes. Concretamente, queremos saber en qué sentido una cadena es representativa de otras clases de cadenas con semejanzas en ciertas posiciones de la cadena.

La noción de esquema proporcionará la herramienta para responder a estas cuestiones.

[31] En inglés la palabra es *run*, que en computación se asocia a *correr* un programa. En este caso se ha traducido por *serie* para indicar el conjunto de todas las iteraciones del programa.

[32] Se refiere a esos juegos de construcción para niños a base de bloques o cubos de madera.

Un *esquema* (Holland, 1968, 1975) es una plantilla de similitudes que describen un subconjunto de las cadenas que poseen semejanzas en ciertas posiciones de la cadena. Veámoslo con un ejemplo. Consideremos el alfabeto {0,1}, y extendámoslo añadiendo el símbolo *, que sirve de "comodín" (es decir, que representa tanto a un 0 como a un 1). Un esquema es una especie de dispositivo para encajar muestras. Por ejemplo, tomemos cadenas y esquemas de longitud 5. El esquema *0001 encaja con dos cadenas, 10001 y 00001. El esquema *101* describe otro subconjunto, el formado por {01010,11010, 01011, 11011}. Como se puede observar, un esquema es entonces un modo poderoso y compacto de hablar de las similitudes bien definidas entre cadenas de longitud finita sobre un alfabeto asimismo finito. Debe señalarse que el símbolo * es lo que se llama un *metasímbolo*, una notación que nos permite describir de manera simple todas las posibles similitudes entre cadenas de una particular longitud y alfabeto. En ningún caso será procesado explícitamente por el algoritmo genético.

Contabilizar el número de esquemas posibles resulta un ejercicio iluminador. En el ejemplo anterior, con longitud l = 5, hay exactamente 35 = 243 posibles plantillas, puesto que los tres elementos 0, 1, * pueden estar en 5 posiciones. En general, para un alfabeto con k caracteres, hay $(k + 1)^l$ posibles esquemas de longitud l. Obsérvese que puede parecer que los esquemas entorpecen más que ayudan en el proceso de búsqueda, puesto que el número de esquemas (243 en nuestro ejemplo) es superior al de cadenas posibles ($k^l = 25 = 32$). No es así: en realidad, los esquemas son útiles como potenciales fuentes de información en base al uso de las similitudes entre las cadenas, pero no todos los esquemas son igual de "útiles". Podemos acotar el número de esquemas en una población contando el número de esquemas en una cadena individual, y obteniendo luego una cota superior para el total de esquemas de la población. Para ello, tomemos el caso 11111. Esta cadena es miembro del conjunto de esquemas de tamaño 25, ya que cada posición puede tomar su valor actual, o el símbolo *. En general, una cadena particular contiene 2^l esquemas. Por tanto, una población de tamaño n contiene entre 2^l y $n2^l$ esquemas, dependiendo de la diversidad de la misma. Este hecho viene a verificar nuestra intuición anterior. La motivación original para considerar las similitudes importantes era obtener más información que guiara nuestra búsqueda. El argumento anterior de contabilizar demuestra que la riqueza de información contenida entre las semejanzas más relevantes está de hecho incluida en cualquier población, incluso en las de tamaño moderado.

Ahora, entre los 2^l y $n2^l$ esquemas contenidos en la población ¿cuántos son procesados de manera efectiva por el algoritmo genético? La respuesta la va a dar el efecto del proceso cruce sobre los esquemas importantes de generación en generación. El cruce deja un esquema indemne cuando no corta dicho esquema, pero puede trastornarlo cuando lo hace. Por ejemplo, tomemos los esquemas 1***0 y **11*. El primero tiene más posibilidades de interrumpirse mediante el cruce, mientras que el segundo parece que no quedará destruido. Éste pertenece al tipo de esquemas que nos van a interesar: con alta capacidad (cuanto mayor sea ésta, más posibilidades tendrá para ser

seleccionada para reproducirse, por lo que el número de muestras para las similitudes mejores observadas será mayor), y con longitud definida corta. Se transmitirán de generación en generación otorgando un número exponencialmente mayor de muestras a las mejores observadas.

Son lo que Goldberg [Goldberg 1989] llama *piezas o bloques de construcción*, que se combinarán para generar cadenas con un rendimiento esperado alto. La causa de que tal cosa ocurra es que las piezas de construcción se muestrean a tasas casi óptimas, y se recombinan vía el cruce. La mutación tendrá poco efecto en estos bloques; este es un mecanismo para prevenir la pérdida irrecuperable de material genético potencialmente importante. Desde luego, quedan otros muchos aspectos importantes en los que se debería profundizar, como cuál ha de ser el tamaño de la población, y cuáles las probabilidades de cruce (P_C) y de mutación (P_M).

De esos "detalles" dependerá, en gran parte, el éxito o fracaso del Algoritmo Genético que estemos aplicando. Asimismo, podemos encontrar otros Algoritmos Genéticos mucho más complejos (y eficaces) que el que se ha utilizado, como aquellos que trabajan sobre otras representaciones además de las cadenas de bits, o los que emplean otros operadores de cruce y mutación.

2.6. Los Algoritmos Genéticos y otros algoritmos de búsqueda

Como ya se ha expuesto, el AG es un método general para resolver problemas de "búsqueda de soluciones" (término más amplio que engloba los de Optimización). Otras aproximaciones son la búsqueda tabú, el templado simulado, o el de "ascenso de la colina".

En la Inteligencia Artificial, a estos métodos generales (los que se pueden aplicar a una gran variedad de problemas) se les conoce como "métodos débiles", para diferenciarlos de los "fuertes", que son aquellos especialmente diseñados para la resolución de un problema concreto. Los métodos de "búsqueda de soluciones" siguen habitualmente el siguiente esquema:

1. Generar un conjunto de candidatos de soluciones (en el AG es la población inicia, etc.).

2. Evaluar las soluciones candidatas de acuerdo a un criterio sobre la aptitud o capacidad de cada elemento del conjunto.

3. En base a esa evaluación, decidir qué candidatos se han de conservar, y cuáles descartar.

4. Generar más variantes empleando algún tipo de operador sobre los candidatos supervivientes.

La combinación particular de elementos que se da en los AG (búsqueda paralela de la población con la selección estocástica de muchos de los individuos, cruce y selección estocásticos), les distinguen de los otros métodos. Concretamente, difieren de los métodos tradicionales de búsqueda (y optimización), en cuatro cuestiones esenciales:

- Trabajan con un código del conjunto de parámetros, no con el conjunto mismo (necesitan que el conjunto de parámetros del problema de optimización esté codificado en cadenas finitas sobre un determinado alfabeto). Por trabajar a nivel de código, y no con las funciones y sus variables de control, como los otros métodos, son más difíciles de "engañar".

- Buscan una población de puntos, no un único punto. Manteniendo una población de puntos muestrales bien adaptados, se reduce la probabilidad de caer en una cima falsa.

- Emplean la función objetivo, no necesitan derivadas ni otra información complementaria, tan difícil a veces de conseguir. De este modo ganan en eficiencia y en generalidad.

- Se valen de reglas de transición estocásticas, no deterministas. Los Algoritmos Genéticos se valen de operadores aleatorios para guiar la búsqueda de los mejores puntos; puede parecer extraño, pero la Naturaleza está llena de precedentes al respecto.

2.7 Algunas aplicaciones de los Algoritmos Genéticos

Aunque, como se ha comentado, el Algoritmo que se utilizó en el apartado anterior es muy simple, ha servido para que los estudios realizados en torno a él, se hayan aplicado a diversos problemas y modelos en ingeniería, y en la ciencia en general[33]. Cabe destacar entre ellos:

- **Optimización:** Se trata de un campo especialmente abonado para el uso de los Algoritmos Genéticos, por las características intrínsecas de estos problemas. No en vano fueron la fuente de inspiración para los creadores estos algoritmos. Los Algoritmos Genéticos se han utilizado en numerosas tareas de optimización, incluyendo la optimización numérica, y los problemas de optimización combinatoria.

[33] En [Goldberg 1989] se puede encontrar un capítulo dedicado a las aplicaciones, tanto históricas como "actuales", de los Algoritmos Genéticos, junto con una tabla en la que se detalla una relación de las investigaciones, el campo al que pertenecen, el año, y los investigadores que las desarrollaron. El problema es que, aunque he consultado ediciones posteriores a 1989 de la obra (2002, 23 *rp*), el autor no ha incluido modificación alguna al citado capítulo.

- **Programación automática:** Los Algoritmos Genéticos se han empleado para desarrollar programas para tareas específicas, y para diseñar otras estructuras computacionales tales como el autómata celular, y las redes de clasificación.

- **Aprendizaje máquina:** Los Algoritmos Genéticos se han utilizado también en muchas de estas aplicaciones, tales como la predicción del tiempo o la estructura de una proteína. Han servido asimismo para desarrollar determinados aspectos de sistemas particulares de aprendizaje, como pueda ser el de los pesos en una red neuronal, las reglas para sistemas de clasificación de aprendizaje o sistemas de producción simbólica, y los sensores para robots.

- **Economía:** En este caso, se ha hecho uso de estos Algoritmos para modelizar procesos de innovación, el desarrollo estrategias de puja, y la aparición de mercados económicos.

- **Sistemas inmunes:** A la hora de modelizar varios aspectos de los sistemas inmunes naturales, incluyendo la mutación somática durante la vida de un individuo y el descubrimiento de familias de genes múltiples en tiempo evolutivo, ha resultado útil el empleo de esta técnica.

- **Ecología:** En la modelización de fenómenos ecológicos tales como las carreras de armamento biológico, la coevolución de parásito-huésped, la simbiosis, y el flujo de recursos.

- **Genética de poblaciones:** En el estudio de preguntas del tipo "¿Bajo qué condiciones será viable evolutivamente un gene para la recombinación?".

- **Evolución y aprendizaje:** Los Algoritmos Genéticos se han utilizado en el estudio de las relaciones entre el aprendizaje individual y la evolución de la especie.

- **Sistemas sociales:** En el estudio de aspectos evolutivos de los sistemas sociales, tales como la evolución del comportamiento social en colonias de insectos, y la evolución de la cooperación y la comunicación en sistemas multi-agentes.

Aunque esta lista no es, en modo alguno, exhaustiva, sí transmite la idea de la variedad de aplicaciones que tienen los Algoritmos Genéticos. Gracias al éxito en estas y otras áreas, los Algoritmos Genéticos han llegado a ser un campo puntero en la investigación actual.

2.8 Mejoras en el funcionamiento de los Algoritmos Genéticos

Una de las razones de ser de los fundamentos teóricos en el tipo de ámbito en el que nos estamos moviendo, debe consistir en ayudar a los profesionales a cuestionar sus intuiciones y asunciones. En el caso de los algoritmos

evolutivos, hablamos de algo muy complejo, lo que significa que lo que asumimos como una intuición cierta, a menudo no lo es. Por ejemplo, no hace mucho, se vendían los Algoritmos Genéticos como unos optimizadotes "de caja negra" que funcionaban mejor, de media, que la búsqueda aleatoria. De hecho, se llegaba a afirmar que el modo en que realizaban la búsqueda en un espacio ya era óptima en cierto sentido. Ahora sabemos que, lógicamente, esto es falso: no hay algoritmos de búsqueda mejores o peores de media. Sí es cierto que hay clases de problemas para los que algunos parecen especialmente adecuados, pero caracterizar tal clase es todavía un problema abierto. Podemos creer que se comportan mejor que la búsqueda aleatoria en el tratamiento de problemas del "mundo real", pero no podemos formalizar esa hipótesis a menos que caractericemos tanto los AG como las clases de problemas mejor de lo que lo hemos hecho hasta el momento.

Lo que piden los profesionales, los que tienen que ejecutar los AG para resolver problemas prácticos, de los investigadores que desarrollan los resultados teóricos, es que tales resultados les proporcionen la respuesta a cuestiones tales como cuál ha de ser el tamaño de la población, o si el algoritmo convergerá. Algunas preguntas son intrínsecamente difíciles de responder, y no existen fórmulas mágicas a las que acudir. Sin embargo, podemos enumerar algunas para las que sí se puede dar una solución:

1. Para garantizar que un algoritmo convergerá eventualmente (en algún sentido) hacia el óptimo, hay que usar un tipo de estrategia elitista, o algoritmo de creación continua. Alternativamente, la tasa de mutación se puede reducir suavemente, mientras se aumenta la fortaleza de la selección (de acuerdo con un esquema establecido).

2. Los Algoritmos Genéticos se han aplicado con éxito a una amplia variedad de problemas considerados difíciles. Sin embargo, existen casos en los que la aplicación del AG ha cosechado pobres resultados, como en problemas de asignación [Bäck et al 1998], o incluso un fallo complete, como en los ejemplos "engañosos" de Goldberg [Goldberg 1989 *b*]. Son lo que en [Goldberg 1989] se llaman *problemas AG-Difíciles*. Se han propuesto métodos para mejorar el comportamiento de los AG ante tales problemas: heurística para operadores genéticos, que introduce un conocimiento previo de las características del problema en el AG [Bäck 1994], [De Jong 1992], la ejecución en paralelo de AG anidados (en *nichos*), que consiste en correr múltiples poblaciones de tamaño pequeño al mismo tiempo, y permitir algunas migraciones de individuos entre las diferentes poblaciones [Cantu-Paz, E. 1999], [Sareni 1998]; y crear algoritmos híbridos con Redes Neuronales de Hopfield, Búsqueda Tabú o Templado Simulado, de manera que se combinen la búsqueda global y local; y las transformaciones *genotipo-genotipo* [Battle 1991], [Field, P. 1994], [Liepins 1990], [Salomon, R. 1996]. En [Salcedo-Sanz 2006] encontramos una discusión acerca de nuevos resultados respecto al efecto de las transformaciones lineales sobre la representación, el espectro y el funcionamiento de un AG estándar.

3. Para optimizar una función con parámetros reales mediante un AG, lo normal es utilizar una representación en código de Gray mejor que la codificación binaria directa, a no ser que la función de capacidad sea patológica.

4. Aumentar el tamaño de la población hará que el AG siga más de cerca la dinámica de la población infinita, lo que significa que la población pasará la mayoría de su tiempo en el entorno de los puntos fijos de las ecuaciones dinámicas que rigen su comportamiento. Eso puede o no ser bueno, en tanto repercute a la búsqueda.

5. Elevar la tasa de mutación empuja los puntos fijos, de modo que las poblaciones tenderán a ser una mezcla de un gran número de individuos. De nuevo, esto puede o no ser algo beneficioso.

6. El papel que juega el cruce es algo que no se comprende todavía con demasiada claridad. Se sabe que es un operador que puede ayudar a la población a saltar sobre algunos tipos de "huecos" en el espacio de búsqueda. Se tienen ejemplos de problemas en los que la recombinación proporciona un relevante aumento de la velocidad. El diseño de unos buenos operadores de recombinación es problemático, aunque sabemos cuáles han de ser las propiedades a preservar durante el proceso de búsqueda. En este caso, existen resultados que pueden ayudar, pero exigen tener un buen conocimiento de la naturaleza del espacio de búsqueda.

7. Es importante también tener en mente el efecto dual que tiene el cruce: la recombinación es un modo general de combinar elementos de dos (o más) padres, pero también implica la aplicación de un operador específico de búsqueda. Es necesario asegurarse de que estos dos efectos se dan en armonía.

8. En el caso de estructuras de tamaño variable (por ejemplo, en programación genética), la elección de los operadores de cruce y mutación puede sesgar el proceso de búsqueda hacia determinados tamaños. Aparentemente, los pequeños cambios en la representación del problema puede tener igualmente efectos dramáticos (es el caso de la modificación en el número de terminales posibles en un problema de programación genética).

9. En general, cuanto mayor sea el conocimiento del problema que podamos introducir en el algoritmo, mejor. En particular, podríamos intentar hacer la representación y los operadores tan "naturales" como sea posible para el problema.

Una de las tareas de los investigadores del ámbito teórico es la de criticar los modelos del comportamiento de los AG, con el objeto de validarlos y de encontrar su campo de aplicación. Debe tenerse especial cuidado con los

modelos de AG que se diseñan alrededor de una clase concreta de problemas. Los podemos describir como modelos "de ingeniería", cuyo ánimo principal es el de detallar la intuición sobre el comportamiento en el problema de modo que se pueda diseñar un algoritmo más eficiente. Sin embargo, aunque nada malo haya en ellos a priori, se debe ser muy cauteloso con la intuición, como ya hemos señalado. Además, debe evitarse la tentación de generalizar sin detenerse a comprobar a otras clases de problemas, que es lo que realmente le daría entidad de resultado a la intuición.

2.9 Resultados empíricos (base de datos 2003)

- *Selección de los factores de riesgo*

Para el primer experimento sobre la selección de factores empleamos un Algoritmo Genético que se programó (en C++) a tal efecto. Le proporcionamos como entrada 58238 clientes seleccionados aleatoriamente de la base de datos enlazada de la Cartera de Clientes 2003 y la de Siniestros 2003 que describimos en el Capítulo 2.

En nuestro problema de selección de los factores relevantes para la tarificación a priori, la población del AG está integrada por un número ξ de cadenas binarias $\sigma \in \{0,1\}^n$ a los que se aplica el procedimiento iterativo de los operadores genéticos. Una componente $\sigma_i = 1$ equivale a afirmar que el factor de riesgo correspondiente debe ser tenido en cuenta para la SVM, mientras que si $\sigma_i = 0$, se eliminará ese factor del conjunto de factores.

Debe observarse que cada individuo de la población del AG (un vector σ) permanece para un conjunto de factores diferente de la SVM. La función de capacidad asociada a cada individuo es el error de clasificación obtenido al clasificar *l* puntos de entrenamiento $(\mathbf{x} * \sigma, y)$, que será estimado por $R_{ent}(\mathbf{w}, b, \sigma)$.

Una última apreciación: como el AG maximiza la función de capacidad, y el objetivo en un problema de selección de características es minimizar la probabilidad de error, se introducirá una función de capacidad modificada:

$$F = 100(1 - R_{ent}(\mathbf{w}, b, \sigma))$$

En resumen, en nuestro Algoritmo, cada individuo es un conjunto de variables, que en nuestro caso representan los factores de riesgo. El número de esas variables puede ser fijo o variable. Nosotros escogimos que fuera fijo, de 30 variables[34] concretamente. El Objetivo del algoritmo es buscar las 30 mejores variables (las de mayor poder predictivo).

[34] Este número se fija por lo que se conoce como *búsqueda generacional*.

La inicialización del genético se hace con cuidado, de modo que todas ellas estén presentes en todos los individuos en la misma proporción. Todos los individuos se inicializan con el mismo número de variables. Luego se deja evolucionar el genético introduciendo ciertas mutaciones entre individuos, haciendo cruces, y se para cuando el error se estanca y/o se pierde la variabilidad en la población. El criterio para escoger los mejores individuos es la tasa de acierto en clasificación usando la técnica de aprendizaje que ya conocemos, la Máquina de Vectores Soporte (SVM). De esta manera, lo que llamamos función de capacidad (fitness) resulta ser la capacidad predictiva estimada con la SVM. El algoritmo utiliza elitismo, lo que significa, como ya vimos, que el mejor individuo de cada generación pasa a la siguiente (sin mutación). El operador de cruce es de 1 punto.

Las 30 variables que resultaron seleccionadas fueron: -antigüedad del carnet, -edad, -conductor, -antigüedad del vehículo, -Madrid, -Barcelona, -valor, -potencia, -Castilla y León, -Galicia, -PROF12 (sin código), PROF41 (funcionarios y administrativos: desplazamiento profesional habitual urbano), -casado, -hombres, -USO110 (turismo de uso particular), -Andalucía (Sevilla), -PROF56 (sin profesión),-Murcia, -Aragón, -soltero, -Sevilla, -PROF20 (industriales, comerciantes, profesiones liberales: sin desplazamiento profesional habitual), -Baleares, -PROF0 (sin código), -Cataluña (Bacelona), -Castilla-La Mancha, -Comunidad Valenciana (Valencia),-USO168 (todo terreno +5 hasta 9 plazas -), -Asturias y PROF55 (obreros manuales).

Aunque con un AG no se seleccionan variables individuales sino grupos de variables (conservando aquel que da mejores prestaciones de clasificación), y, por tanto, no se podría establecer una relación de orden en el grupo, en nuestro caso el modo de realizar el experimento sí lo ha permitido. Lo que se hizo fue probar con conjuntos de una variable, luego de dos, de tres, etc. De este modo, las variables seleccionadas en un paso se introducían en el paso siguiente, dotando así al conjunto final de una cierta relación de orden.Ya hemos comentado que el objetivo del trabajo no es analizar los resultados de la aplicación de las técnicas de aprendizaje a los datos desde un punto de vista actuarial. Pese a ello sí podemos observar con curiosidad los factores de riesgo que han resultado elegidos. Los dos primeros, además de no resultar sorprendente su selección, son siempre determinantes en cualquier proceso de tarificación, y, según nos aseguran en MAPFRE, son los que realmente utilizan, junto al sexo, en sus estudios. Precisamente llama la atención que la variable sexo se encuentre en la posición 13, por debajo de factores como conducir en Galicia o ser una persona casada.

3. ÁRBOLES DE CLASIFICACIÓN. RESULTADOS

3.1. Los Árboles de Clasificación

El Aprendizaje por Árboles de Decisión es una de las técnicas de inferencia inductiva más utilizadas y prácticas de las que abarca la Teoría del

Aprendizaje. Se han aplicado con mucho éxito a un amplio abanico de tareas, desde el diagnóstico médico, al estudio de riesgo en la concesión de créditos bancarios.

Aunque es un campo relativamente nuevo, y con menos historial de aplicaciones que las otras técnicas que hemos comentado hasta ahora, los orígenes se remontan a la década de 1960, con los trabajos en ciencias sociales de Morgan y Sonquist (1963), y Morgan y Messenger (1973). Una gran influencia tienen las publicaciones de Breimann et al. (1984), tanto en el sentido de despertar el interés de los estadísticos como en el de proponer nuevos algoritmos para la construcción de árboles.

Siguiendo a [Breimann 1984], sobre la misma época que los anteriores encontramos los primeros trabajos en el aprendizaje mediante árboles destacando el "Sistema de Aprendizaje de Concepto" de Hunt (Hunt et al., 1966) y el trabajo de Friedman y Breiman que desembocó en el sistema CART (Friedman 1977; Breiman et al. 1984). También conviene mencionar al autor del algoritmo ID3, Quinlan, en 1979 y 1983 ([Quinlan 1983]). Otros de los primeros estudios en el tema se deben a Kononenko et al. en 1984, y Cestnik et al. en 1987 con su algoritmo ASSISTANT. Clark y Pregibon (1992) describen los modelos basados en árboles y los implementan en lenguaje S, logrando que sean mucho más asequibles, y popularizándolos. Sus métodos son muy flexibles y orientados al análisis exploratorio de datos.

A pesar de todo, no hay una amplia literatura histórica sobre los árboles, estando dispersas la mayoría de las contribuciones estadísticas como la que nos interesa a nosotros en el presente trabajo. Su evolución histórica ha seguido las pautas del resto de técnicas que hemos analizado, ligada a los progresos tecnológicos y a los teóricos en cuanto al aprendizaje y la estadística.

Hablando de forma general, podemos considerar el aprendizaje mediante árboles como un procedimiento para aproximar funciones objetivo valoradas de forma discreta, en los que la función que se va a aprender se representa por medio de un árbol de decisión. Los árboles de aprendizaje permiten también una segunda representación como conjuntos de reglas if-then ("si-entonces") para mejorar la legibilidad humana.

Una de las novedades que presentan los árboles es que permiten el tratamiento de datos no numéricos de modo "directo". En los métodos que se habían tratado hasta ahora, se emplearon datos representados por valores numéricos, reales o discretos, generalmente en forma de vector. Pero, ¿cómo tratar un problema de clasificación en el que los valores que tenemos son *nominales* (por ejemplo, descripciones discretas)? La dificultad más importante que presenta este tipo de variables, es que, a diferencia de las numéricas, carecen de la noción de similaridad y, aún peor, no existe relación de orden entre ellas. Son lo que se llaman *variables categóricas*. En este apartado dirigiremos nuestra atención hacia las muestras que se presentan descritas por

listas de atributos, y no por vectores de números reales. Un modo bastante usual de tratar este tipo de datos es especificando los valores de un número fijo de propiedades mediante una *d-tupla de propiedades*, en las que se den los valores nominales que caractericen el objeto. Otra vía es describir la muestra utilizando una cadena de longitud variable con atributos nominales (por ejemplo, AGTTCACGATTCAT representaría un segmento de la cadena de ADN).

Entonces, ¿cómo podemos *aprender* eficientemente utilizando datos no métricos? El problema de clasificación es el mismo que llevamos planteando hasta ahora: dado un conjunto de medidas de cierto objeto, encontrar un método sistemático de predecir a qué clase de unas dadas pertenece. Sabemos que un modo natural e intuitivo de clasificar una muestra, venga como venga representada, es mediante una secuencia de preguntas, en las que la pregunta siguiente dependa de la respuesta de la pregunta anterior. Evidentemente, este sería un enfoque excelente para el tipo de datos al que ahora nos referimos, ya que las preguntas se plantean para que las respuestas sea "si/no", "verdadero/falso", o "valor(propiedad) $\in \{valores\}$", lo que no depende de métrica alguna.

La apariencia gráfica de esa secuencia de preguntas es la de un "árbol", en el que la primera pregunta se coloca en el primer *nodo* o *raíz*, que va uniendo sucesivamente las respuestas con los otros nodos mediante unos conectores que representan las *ramas* del árbol. Los nodos terminales, de los que no parten más ramas, serían las hojas del árbol, en los que se recogería la respuesta al problema de clasificación o decisión.

En este apartado, trataremos este método de clasificación, conocido como árbol de clasificación, dando en una primera instancia las definiciones y conceptos matemáticos necesarios para su descripción, así como unas breves notas históricas al respecto. Pasaremos luego a los procesos de construcción, desarrollo y podado del árbol. Finalmente, se tratarán las ventajas que la estructura de árbol tiene frente a otros métodos de clasificación.

3.2 Definiciones y conceptos matemáticos

- Elementos de un Árbol de Clasificación

Un *Árbol de Decisión* o de Clasificación, o, sencillamente, árbol, es básicamente un diagrama que representa un sistema de clasificación o un modelo predictivo como el arriba descrito [Breimann 1984]. El árbol se estructura como una secuencia de preguntas sencillas, cuyas respuestas trazan un camino que lleva hacia abajo en él árbol. El (o los) punto final alcanzado (*hojas*) determina la clasificación o predicción hecha por el modelo, que puede constituir una respuesta tanto cualitativa como cuantitativa.

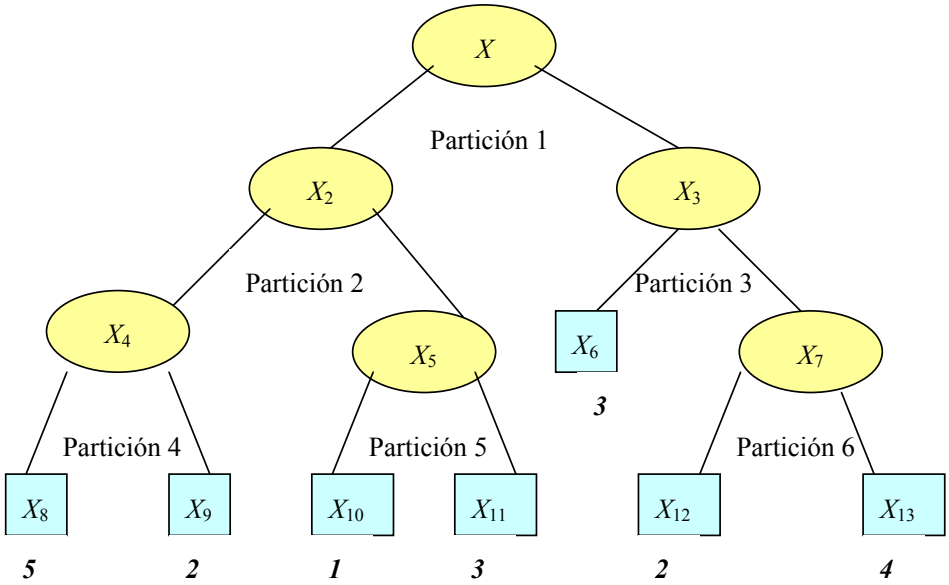
La construcción de árboles se puede ver como un tipo de selección de variables, en el que cuestiones como la interacción entre las variables o las transformaciones monótonas de éstas se manejan automáticamente. Como ya hemos apuntado, se utilizan más frecuentemente como método de clasificación, aunque es posible construir *Árboles de Regresión* en los que cada nodo final es un valor predicho. Como veremos más adelante, la diferencia entre ambos radica en el tipo de variable dependiente que se utiliza: mientras que en el árbol de clasificación es una variable cualitativa, en el de regresión es cuantitativa.

Notemos con *X* el espacio muestral de la variable que deseamos clasificar. El objetivo del árbol es determinar una partición del espacio. Los árboles dividen el espacio en rectángulos multidimensionales vía el uso de reglas lógicas *binarias*[35]. En realidad, un árbol corta el espacio *X* en hipercubos más y más pequeños, cada uno de ellos correspondiente a un nodo terminal del árbol.

Dos árboles se pueden comparar en el sentido de que su partición inducida se aproxime en mayor o menor medida a la regla de decisión correcta para el problema. En problemas de tipo lógico el modo más sencillo de comparar particiones es contar el número de errores que se cometen en la clasificación, o bien, si se dispone de una partición a priori de *X*, calculando la probabilidad de error.

Pero veamos más detalladamente cuáles son los elementos de un árbol, y cuál sería el proceso intuitivo para construirlo. El siguiente diagrama representaría un árbol binario de 5 clases [Breimann 1984]:

[35] La estructura de árboles clasificadores binarios, de los que de cada nodo parten dos únicas ramas, es la más habitual, pero no la única.



Cada división se corresponde con la respuesta a una pregunta formulada. Según ésta, se separan los elementos del conjunto en dos subconjuntos disjuntos[36], cuya unión es el conjunto del que habían partido. En este ejemplo, $X = X_2 \cup X_3$ y $X_2 \cap X_3 = \varnothing$. Del mismo modo, $X_2 = X_4 \cup X_5$ y $X_4 \cap X_5 = \varnothing$, y $X_3 = X_6 \cup X_7$ y $X_6 \cap X_7 = \varnothing$. Los subconjuntos sin división ulterior, en este caso $X_6, X_8, X_9, X_{10}, X_{11}, X_{12}, X_{13}$, son los subconjuntos o nodos terminales (las hojas del árbol, que decíamos antes). Todos ellos forman una partición del nodo raíz *X*.

Cada uno de los nodos terminales está señalado con una etiqueta que designa a una clase. Evidentemente, distintos de estos nodos pueden pertenecer a la misma clase, esto es, pueden tener la misma etiqueta. La partición correspondiente al clasificador es la obtenida colocando juntos todas las hojas que pertenecen a la misma clase. Así, en nuestro ejemplo tendríamos:

$$A_1 = X_{10} \qquad A_2 = X_9 \cup X_{12} \qquad A_3 = X_6 \cup X_{11}$$
$$A_4 = X_{13} \qquad A_5 = X_8$$

En problemas de tipo estadístico, la distribución de clases de *X* normalmente se solapa, por lo que no constituye realmente una partición que describa completamente tales clases. Para cada celda de la partición se tiene una

[36] Se puede demostrar que todo árbol puede representarse tomando únicamente decisiones binarias.

distribución de probabilidad sobre todas las clases, y la regla de decisión de Bayes escoge aquella clase con mayor probabilidad, que estará relacionada con la idea de minimizar la probabilidad conjunta de clasificar incorrectamente. Por supuesto, no se conoce en la práctica la estructura de la probabilidad completa, pero sí se dispondrá de un conjunto selecto de *n* casos clasificados (conjunto de test, como en los otros métodos de aprendizaje) que podemos asumir que constituyes una muestra aleatoria e independiente. De este modo podemos estimar la probabilidad de error de clasificación como la proporción de dichos errores en la muestra.

- El proceso de construcción de árboles (CART)

El desarrollo de la estructura de árboles tal y como la vamos a exponer en este apartado surgió como la mejor solución a aplicar para abordar el "problema de clasificación de barcos". Dicho proyecto (Hooper y Lucero, 1976) consistía en el reconocimiento de seis clases de barcos a través de los perfiles de sus señales de radar. Los datos los recogía un aeroplano que volaba en círculos por encima de los barcos de los seis distintos tipos. Una de las primeras dificultades con que se enfrentaron era la reducción de la dimensionalidad. Mucha de la información en los perfiles de las señales era redundante, estaba correlada para barcos cercanos, etc.

En un principio, se tomó la decisión de extraer de cada perfil el vector de localizaciones de los máximos locales. De este modo, cada perfil tenía asociado un vector que recogía en su primera componente la posición del primer máximo local, etc. Sin embargo, esta solución trajo consigo un nuevo problema: los datos tenían una dimensionalidad variable, en un rango de 1 a 15. Ninguno de los métodos de clasificación disponibles parecían adecuados para esta estructura de datos. La solución más satisfactoria la dieron las estructuras de árboles tal y como la vamos a contar a continuación.

Se utilizó la metodología de "un paso adelante", de la que se valen la mayoría de los métodos de construcción de árboles de clasificación, que básicamente consiste en tomar la partición siguiente a una dada de manera óptima pero sin optimizar el conjunto del árbol, lo que evita una explosión combinatoria en cuanto al número de las decisiones futuras a considerar. Eligiendo la medida justa de optimizar en cada corte, podemos facilitar las próximas divisiones.

El ejercicio de generación y "cultivo" (crecimiento) de un árbol parte de un conjunto de datos de entrenamiento "etiquetados" sobre el que se ha decidido el conjunto de propiedades que se pueden emplear para discriminar la muestra, aunque debe determinarse de qué modo se organizarán los test sobre el árbol. Lo ideal sería que cada subconjunto en el que se divida una muestra contenga datos con la misma etiqueta. En ese caso, diríamos que ese subconjunto sería "puro", y podría terminar esa porción del árbol. Sin embargo, lo habitual es que exista una mezcla de etiquetas en cada subconjunto, de manera que en cada rama, habrá de tomar la decisión de seguir dividiendo y hacer que el árbol crezca más, u optar por parar, y aceptar una decisión imperfecta.

Esta idea sugiere un proceso recursivo de crecimiento del árbol bastante obvio: dado un subconjunto de datos representado en cierto nodo, declararlo un nodo terminal (y establecer qué categoría asignarle), o dar otra propiedad para dividir el subconjunto en otros (dos). En realidad, ese es tan sólo un ejemplo de una metodología de crecimiento de árboles mucho más amplia conocida como CART ® (classification and regression trees), que es la que vamos a seguir en este Trabajo.

3.3 Elementos para la construcción de un árbol

El proceso general de construcción de un árbol gira en torno a tres elementos:

1. La selección de las reglas de decisión para separar cada nodo. Esto implica un procedimiento de búsqueda de la característica *j*-ésima que debería usarse en el *i*-ésimo nodo para realizar la partición de las muestras en los subgrupos.

2. Escoger el umbral x_k para la característica *j*-ésima que se debería asignar en el *i*-ésimo nodo. Un ejemplo de decisión sería $\left\{\mathbf{x} \in \Re : x_j \leq x_k\right\}$)

3. La decisión de cuándo declarar un nodo terminal o continuar dividiéndole. Posteriormente habrá que realizar la asignación de cada hoja a una clase.

La raíz del problema reside en encontrar buenas divisiones y en saber cuándo dejar de separar.

A continuación pasaremos a describir las reglas de separación, para luego comentar las reglas de parada para asignar los nodos terminales:

- Reglas de partición y de parada

Siguiendo a L. Breimann [Breimann 1984], las particiones se llevan a cabo de manera que la característica *j*-ésima y el umbral en cierto nodo sean tan "puros" como sea posible. El descenso en la impureza en cada nodo *t* está medido con una *función de impureza* $I(t)$. Una muy utilizada es la definida por el *Criterio de Gini*:

$$\begin{aligned} I(t) &= \sum_{i \neq j} p\left(w_i \middle| t\right) p\left(w_j \middle| t\right) \\ p\left(w_i \middle| t\right) &= N_j(t) / N(t) \end{aligned} \tag{3.1}$$

donde $I(t)$ es la impureza en el nodo *t*, $p\left(w_i \middle| t\right)$ es una estimación de la distribución de probabilidad sobre w_i, $N_j(t)$ es el número de muestras en la clase w_j que llevan al nodo *t* y *N*(*t*) es el número total de muestras en ese nodo.

La función de impureza $I(t)$ en el nodo *t* nos permite elegir la partición que minimice la impureza en las dos amas que conducen hacia los nodos t_I y t_D (izquierdo y derecho respectivamente). El descenso en la impureza de la función es,

$$\begin{aligned}\delta I(t) &\equiv \sum_{i\neq j} p(w_i|t)p(w_j|t) - \left(I(t_D)p_D + I(t_I)p_I\right) \\ &\equiv \sum_{i\neq j} p(w_i|t)p(w_j|t) - \left(\sum_{i\neq j} p(w_i|t_D)p(w_j|t_D)p_D + \sum_{i\neq j} p(w_i|t_I)p(w_j|t_I)p_I\right)\end{aligned} \tag{3.2}$$

Ahora podemos proceder a hacer crecer nuestro árbol mediante el "método de un paso adelante" seleccionando nodos que maximicen el descenso en la impureza sobre nuestras características. El procedimiento descrito arriba puede aplicarse hasta que cada nodo terminal contenga una sola observación, pero a menudo conduce a una sobrealimentación de los datos de entrenamiento y la clasificación no demuestra ser útil. Debe, por tanto, detenerse el proceso de división o poda***r*** el árbol hasta que se tengan miembros puros 8 en el sentido que se comentó antes de todas las muestras que lleguen a un nodo pertenezcan a la misma clase) a casi puros de clase. Veremos a continuación un algoritmo estándar para podar y determinar así qué nodos son terminales.

- La poda del árbol

El método más popular de poda es el propuesto por Breimann, que parte de una idea básica: los árboles demasiado grandes (con muchas hojas) conducen a una sobrealimentación del sistema.

Por otra parte, la tasa de mala clasificación $R(t)$ aumentará si el número de hojas es demasiado escaso, por lo que debe buscarse el equilibrio entre ambas premisas.

Antes de examinar el algoritmo, debemos dar algunas definiciones. Diremos que T_i es un *subárbol podado* de *T* si ambos tienen el mismo nodo raíz. Notaremos con $N(\widetilde{T}_i)$ el número de elementos de del conjunto $\widetilde{T}_i$ de nodos terminales del árbol T_i. Por último, sea α el coste de la complejidad (control de la penalización) de $N(\widetilde{T}_i)$. La tasa de la penalización por mala clasificación para el nodo *t* es $R_\alpha(t)$, y se puede calcular del siguiente modo:

$$\begin{aligned}R_\alpha(t) &= R(t) + \alpha \\ R_\alpha(T_i) &= \sum_{t\in\widetilde{T}_i} (R(t) + \alpha) = R(T_i) + \alpha N(\widetilde{T}_i)\end{aligned} \tag{3.3}$$

Cuando $R_\alpha(T_i) = R_\alpha(t)$ la contribución que hace al coste de la complejidad del el subárbol T_i es la misma que para el nodo *t*, y de este modo, α puede

expresarse como función del nodo *t*

$$\alpha(t) = \frac{R(t) - R(T_i)}{N(\widetilde{T}_i) - 1} \tag{3.4}$$

El numerador es la diferencia entre el error estimado por la clasificación errónea para el nodo *t* y la estimación del error por mala clasificación para el subárbol completo que tenga la misma raíz, y el denominador es el número de nodos terminales del subárbol. Si α = 0, no habrá poda, ni por tanto, penalización. Por el contrario, si escogemos $\alpha = +\infty$, se eliminarán todos los nodos menos el raíz. En la práctica, podemos calcular o estimar $R(T_i)$ estimando el tamaño del error por mala clasificación en el nodo *t*. Es la cuantía del error por clasificación errónea para el subárbol T_i. El único parámetro libre es $\alpha(t)$, reflejando de este modo el compromiso descrito anteriormente.

El proceso empleado para podar el árbol *T*, ahora que hemos definido todo lo necesario, es:

1. Buscar el nodo t_k (puede ser más de uno) con el menor valor de $\alpha(t) = \alpha_k$.
2. Convertirlo en un nodo terminal. El nuevo árbol si las sub-ramas desde el nodo t_k se etiqueta como T^k.
3. Calcular el nuevo $\alpha(t)$ para sus antecesores, siempre superior a α_k.

Este procedimiento se itera *k* veces hasta que el nodo terminal sea el raíz. Tendremos así una colección de árboles $T^k \quad (k \in \{1,...,k\})$ y el correspondiente α_k.

Ahora podemos hacer que nuestro árbol crezca y podarlo para que encaje con nuestro conjunto de entrenamiento, pero puede ser ineficiente para la clasificación especialmente cuando el conjunto de datos disponible contiene menos de mil casos [Breimann 1984] , por lo que se utilizará la validación cruzada para seleccionar el mejor árbol, permitiendo el pronóstico de nuestro clasificador.

- Selección del árbol óptimo

Tras iterar *k* veces la poda del árbol *T*, tendremos una serie de árboles $T^k \quad (k \in \{1,...,k\})$. Aplicando la validación cruzada de orden 10, tomaremos un 90% de la muestra, haremos crecer el árbol utilizando sólo es parte de la muestra, y podaremos una secuencia de subárboles, calculando la tasa de error para cada subárbol de la secuencia valiéndonos del 10% restante de la muestra como conjunto de test. Se repetirá este procedimiento durante 10 veces, empleando cada vez una parte diferente de la muestra como conjunto de test y de estimación.

El problema radica en que tenemos datos diferentes para aumentar el árbol y para podarlo, y, por tanto, secuencias diferentes de α_k.

El método descrito por Breimann consiste en primero hacer crecer y luego podar el árbol utilizando todos los datos, lo que nos proporciona la serie de α_k, para luego construir una nueva secuencia tomando la media geométrica $\hat{\alpha}_k = \sqrt{\alpha_k \alpha_{k+1}}$. Cuando la poda de los árboles crece con el 90% de la muestra, se escoge el mejor árbol del conjunto que minimice la estimación por la mala clasificación $R^{CV}(T^k)$.
Sea $T(\alpha)$ el subárbol podado de modo que todos sus nodos verifiquen que $g(t) > \alpha$ (esto es, $T(\alpha)$ = T^k en la etapa *k*). Ejecutando la validación cruzada "10--fold" para podar el árbol, el método quedaría del siguiente modo:

1. Emplear el proceso repoda para generar una secuencia de subárboles podados T^k de T.

2. Formar *V* subconjuntos de aproximadamente el mismo tamaño $L_v \in \{L_1, ...L_V\}$ y crear el árbol T_v ..

3. Utilizar el proceso de poda para dar lugar a una secuencia de subárboles poblados T^{vk} de T_v

4. Calcular la estimación de la validación cruzada por la mala clasificación

$$R^{CV}(T^k) = \frac{1}{V}\sum_{v=1}^{V} R_v\left(T_v(\hat{\alpha}_k)\right) \qquad (3.5)$$

donde R_v es la estimación por mala clasificación basada en el subconjunto L_v para el árbol $T_v(\hat{\alpha}_k)$ y $\hat{\alpha}_k$ es la media geométrica.

5. Seleccionar el árbol más pequeño $T_{\min} \in \{T^1, ..., T^k\}$ tal que

$$R^{CV}\left(T_{\min}\right) = \min_k R^{CV}\left(T^k\right) \qquad (3.6)$$

6. El árbol podado final es $T_{\min}$

3.4 Deficiencias (y sus soluciones) del procedimiento de crecimiento del árbol

A pesar del atractivo que tenían los clasificadores mediante árboles en un principio, las deficiencias en el proceso de crecimiento del árbol para el proyecto de clasificación de barcos comenzaron a revelarse pronto. Éstas, y las soluciones adoptadas para evitarlas, de las que se generan nuevos métodos de este tipo, son las que se exponen a continuación.

- *Crecimiento de árboles de tamaño adecuado*

La dificultad más significante fue que los resultados no eran fiables. Cuanto más se iba dividiendo, mejor se pensaba que se estaba haciendo. Muy pocos de los árboles que en un principio se hicieron crecer funcionaron cuando se ejecutó una muestra de test, y los árboles eran mayores de lo que garantizaba la información recogida en los datos.
El uso de reglas de parada más complejas no ayudó. La solución se presentó cuando, en vez de tratar de detener las divisiones en el conjunto adecuado de nodos terminales, se continuó con la partición hasta que los nodos terminales eran muy pequeños, lo que derivó en un árbol muy grande. A continuación, se le podó selectivamente, dando lugar a una secuencia decreciente de subárboles. Entonces se empleó la validación cruzada o las estimaciones de con muestras de test para escoger aquel subárbol con la tasa estimada de clasificación errónea más baja.

- *Reglas para las divisiones*

Son muchos los criterios que se pueden definir para seleccionar la mejor partición a realizar en cada nodo. Veamos dos ejemplos: Una es la que utiliza el *índice de* ***Gini*** *de la diversidad* como una medida de la impureza del nodo.

$$i(t) = \sum_{i \neq j} p(i/t)\, p(j/t)$$

Otro caso es la "*twoing rule*", que no está relacionada con ninguna medida de impureza: en el nodo *t*, cuando la regla *s* le divide en t_I y t_D, escoger aquella división *s* que maximice la expresión:

$$\frac{p_I p_D}{4}\left[\sum_j \left|p(j/t_I) - p(j/t_R)\right|\right]^2$$

Sin embargo, más que exponer reglas de división, la conclusión más importante que se alcanza en este terreno es que, sorprendentemente, el árbol final es bastante insensible a la regla que se haya elegido para realizar las particiones. Mucha más importancia tiene el criterio de poda.

- *Combinaciones variables*

Otra deficiencia que se origina en los árboles que emplean la estructura estándar es que todas las particiones se realizan sobre variables individuales. Esto representa un problema cuando la estructura de las clases depende de variables que están combinadas.

En situaciones en las que se espera una estructura lineal, se puede extender el conjunto de divisiones posibles para incluir todas las combinaciones lineales de las divisiones de la forma :

$$\text{¿Es } \sum_m a_m x_m \leq c?$$

Existe un algoritmo que realiza una búsqueda entre tales particiones en un esfuerzo para encontrar aquella que maximice la bondad del criterio de división.

Al incorporar las combinaciones de divisiones, los árboles con la nueva estructura obtienen resultados competitivos e incluso mejores que aquellos que se logran con el análisis discriminante.

En otro tipo de problemas de clasificación, lo que se observan son variables en combinaciones Booleanas. Esto suele darse en las diagnosis médicas, donde las preguntas son del tipo "¿Presenta el paciente el (síntoma A y el síntoma B) o el síntoma C? En estos casos, resulta útil extender las divisiones posibles para incluir las combinaciones Booleanas entre ellas. Sin embargo, hay que tener en cuenta que, dado que el número de particiones se incrementará rápidamente con la longitud de las expresiones Booleanas, habrá que desarrollar un método de "paso a paso" para hacer la búsqueda computacionalmente factible.

- *Valores perdidos*

En este tema se consideran dos aspectos diferentes: primero, algunas de las clases pueden tener algunas medidas de los valores perdidas (suponemos que siempre contamos con la etiqueta de las clases), y segundo, puede que queramos el árbol completo para predecir la etiqueta de la clase a la que pertenecerá un vector de medidas con algunos de sus valores perdidos.

Existe un algoritmo que sirve para tratar ambos problemas, y que se vale de lo que llaman *divisiones sustitutas*. La idea básica en que se fundamenta es la siguiente: se define una medida de similaridad entre dos divisiones *s* y *s'* en el nodo *t*. Si la mejor separación en *t* es *s* sobre la variable x_m, hallar la división *s'* sobre el resto de las variables que sea más similar a *s*, y nombremos *s'* a la mejor sustituta de *s*. Del mismo se definen la segunda mejor sustituta, la tercera, etc.

Si un caso tiene perdido a x_m en esta medida, se decide si irá a t_I o t_D utilizando la mejor división sustituta. Si la variable perdida es la que contenía la mejor sustituta, se emplea la segunda mejor, etc.

- *Interpretación del árbol*

Otra dificultad a añadir a las que ya hemos enunciado, es que la estructura final del árbol puede resultar engañosa, y conducir a una mala interpretación. Por ejemplo, si una variable nunca se ha dividido al llegar al final del árbol, podría pensarse que es debido a que su asociación con los miembros de la clase es muy pequeña.

Las estructuras de árbol pueden ser muy inestables. Si una variable cubre estrechamente a otra, entonces un pequeño cambio en la muestra de aprendizaje puede mover la división de una variable a otra. Dos particiones posibles en un nodo pueden no ser similares pero sí tener la misma bondad. Los cambios pequeños pueden en un caso favorecer a una, y en otros a otra.

Para resolver esta situación hay un método que consiste en hacer un ranking de variables en términos de su efecto potencial sobre la clasificación. Incluso aunque una variable no aparezca en una división en el árbol final, su ranking puede ser muy alto, indicando si está enmascarando a otra.
Muy a menudo, interesa explorar un rango de parámetros, y/o el efecto de añadir o eliminar variables. Como es posible hacer crecer un número de árboles y utilizarlos para comparar, puede no ser necesario tener que llegar a una validación cruzada, y utilizar un método de crecimiento de árboles exploratorios.

- *Eficiencia Computacional*

El proceso de crecimiento de un árbol mediante la validación cruzada de 10 puede consumir computacionalmente mucho tiempo con un conjunto de datos grande.

Si el tamaño de la muestra en un nodo *t* es mayor que un máximo fijado, se puede utilizar una "submuestra" de cada una de las clases representadas en *t* para determinar la mejor división. A pesar de ello, el conjunto completo de datos que había en *t* desciende en la división. Por tanto, hacer esta selección de las muestras provoca que el tiempo de computación sea menor en los nodos grandes sin deque afecte al tamaño de la muestra que desciende por el árbol.

3.5 Otros algoritmos de generación de árboles

En principio, todas las técnicas de clasificación basadas en árboles pueden incorporar los elementos que se han descrito arriba. Sin embargo, todo lo que se ha planteado gira en torno a las ideas centrales que se utilizaban en las primeras presentaciones de CART. A continuación se comenta otro algoritmo de construcción de árboles bastante popular, dando así una conclusión más global al apartado.

El que se considera como el algoritmo más sencillo y del que parten todas las variaciones/ampliaciones es el que el que emplea una búsqueda exhaustiva de arriba abajo sobre el espacio de posibles árboles de decisión. Éste tipo de aproximación se ejemplifica por medio del algoritmo ID3[37] (Quinlan 1989) y su sucesor C4.5 (Quinlan 1993). Estaba desarrollado para utilizarse sólo con entradas nominales (sin orden). Si en el problema aparecían variables reales, primero se codificaban en binario en intervalos, cada uno de los cuales se consideraba como un atributo nominal sin orden.

Cada división tiene un factor de ramificación B_j, donde B_j es el número de atributos discretos binarios de la variable *j* elegida para realizar la partición.

Este tipo de árboles el mismo número de hojas que de variables de entrada. El algoritmo continúa hasta que todos los nodos son puros y no hay más variables sobre las que dividir.

El algoritmo básico, ID3 "aprende" árboles de decisión construyéndoles de arriba abajo, comenzando por hacer la pregunta "¿qué atributo debería testarse en la raíz de árbol? Para responderla, se toman ejemplos de cada atributo y se evalúa su capacidad para clasificar por sí solos las muestras de entrenamiento. Se selecciona así el mejor atributo y se utiliza como test en la raíz del árbol.

Para cada valor de este atributo se crea un descendiente de la raíz, y para cada uno se clasifica el conjunto de entrenamiento. El proceso completo se repite empleando los ejemplos de entrenamiento asociados con cada nodo descendiente para seleccionar el mejor atributo para testar en ese punto del árbol. Esto constituye una búsqueda exhaustiva para un árbol de decisión aceptable, en la que el algoritmo nunca retrocede para reconsiderar las elecciones pasadas.

Como se puede apreciar, una pieza básica del algoritmo es la determinación del mejor argumento para localizarlo en la raíz y comenzar a realizar las particiones del conjunto. Eso requiere la definición de una medida que cuantifique de algún modo la "utilidad" de ese atributo a la hora de clasificar los ejemplos.

En ese sentido se define una propiedad estadística llamada *ganancia de información*, que mide qué tal separa el atributo dado los ejemplos que se le presenten de acuerdo a su clasificación objetivo. Sin especificar mucho más, se la puede definir como la reducción esperada en la *entropía* (medida de la impureza en una colección de ejemplos de entrenamiento), causada por la partición de los ejemplos de acuerdo al atributo. Concretamente, el ID3 emplea esta medida para escoger entre los atributos candidatos en cada paso mientas hace crecer el árbol.

[37] El nombre viene de que era el tercero en una serie de procedimientos para "dicotomizar de modo interactivo".

3.6 Ventajas de la estructura de árboles

Como ya se ha establecido en las secciones anteriores, la clasificación mediante la estructura de árboles es un procedimiento iterativo y recursivo[38] que requiere la determinación de sólo unos pocos elementos:

1. El conjunto de preguntas.
2. Una regla para seleccionar la mejor partición en cada nodo.
3. Un criterio para escoger el tamaño más adecuado para el árbol.

Tiene el potencial para ser, y de hecho lo es, una herramienta de clasificación que combina la flexibilidad con la potencia. En particular:

1. Puede aplicarse a cualquier estructura de datos vía la formulación apropiada del conjunto de preguntas. En una estructura estándar puede manejar variables tanto categóricas como ordenadas de un modo sencillo y natural.

2. La clasificación final se presenta de forma simple, y puede almacenarse de modo compacto. Además, y esto es muy importante, clasifica eficientemente datos nuevos.

3. Logra un uso poderoso de la información condicional en el manejo de las relaciones no homogéneas.

4. La selección “paso a paso” de las variables y la reducción de complejidad las realiza de forma automática.

5. Proporciona, sin esfuerzo adicional, no sólo la clasificación, sino también una estimación de la probabilidad de una clasificación errónea del objeto.

6. En una estructura de datos estándar es invariante bajo todas las transformaciones monótonas de las variables ordenadas individuales.

7. Es extremadamente robusta frente a los “outliers” y los puntos mal clasificados.

8. La salida del árbol proporciona una información fácil de comprender y de interpretar respecto a la estructura predictiva de los datos.

Es por todo ello que, como veremos en el siguiente punto, los métodos de estructura de árboles se han empleado en una variedad de aplicaciones en colaboración con químicos sin orientación estadística, médicos, meteorólogos, físicos, etc. Su reacción casi unánime fue que el árbol clasificador nos da una vía natural e iluminativa para comprender la estructura del problema.

[38] Por supuesto, nos referimos al algoritmo general de árboles de clasificación.

3.7 Problemas apropiados para el aprendizaje mediante árboles

A pesar de que se han desarrollado una variedad de métodos de aprendizaje mediante árboles con una diversidad de requisitos y aptitudes, existen problemas para los que este tipo de aprendizaje está especialmente capacitado. Una muestra de ellos es la siguiente:

- *Los datos están representados por pares de atributos-valores*: Cuando los ejemplos que se le presentan al clasificador están descritos por unos atributos (por ejemplo, *Temperatura*) y sus valores (*Caliente*), los árboles destacan por su comportamiento. La situación más sencilla será, lógicamente, aquella en la que los atributos tomen un número pequeño de valores disjuntos (*caliente*, *templado*, *frío*). Existen extensiones del algoritmo básico que permiten manejar también atributos con valores reales. (por ejemplo, representar la *Temperatura* en grados centígrados).

- La función objetivo tiene valores discretos de salida: Es habitual encontrar árboles que asignan una clasificación de tipo booleano a cada ejemplo. Por supuesto, los árboles se pueden extender para aplicarlos a funciones de aprendizaje con más de dos valores de salida. Una extensión más sustancial permitiría tratar también funciones de aprendizaje con output reales, si bien es cierto que la aplicación de los árboles de decisión a estos casos es menos frecuente.

- Se precisan descripciones disyuntivas: Como ya se ha comentado, los árboles representan de modo natural expresiones dadas como disyuntivas.

- Los datos de entrenamiento pueden contener errores: Estos métodos de aprendizaje son robustos frente a los errores, tanto de clasificación en los ejemplos de entrenamiento, como en los valores de los atributos que describen tales ejemplos.

- Los datos de entrenamiento presentan atributos cuyo valor se ha perdido: Lo árboles de decisión pueden emplearse cuando algunos ejemplos de entrenamiento tienen valores desconocidos (por ejemplo, que la *Humedad* del día se conozca sólo para algunas fechas).

Son muchos los problemas de la práctica que encajan con estas características. Como muestra, tenemos el de la clasificación de pacientes en Medicina según los síntomas de sus enfermedades, el de la clasificación de equipos que funciona mal según la causa de los errores, y la de la agrupación de clientes para la concesión de créditos según su comportamiento a la hora de incumplir pagos.

Así, en la última conferencia sobre *CART® Data Mining* organizada por Salford Systems celebrada del 22 al 24 de Mayo de 2004 en San Francisco, encontramos artículos de “interés general” (*Avoiding the 10 Top Mistakes in Data Mining*), de “biomedicina” (*Using CART to Develop a Diagnostic Tool for*

Erectile Dysfunction, *CART for Outcome Predictions in Clinical Settings: Emergency Department Triage*, *Survival Prediction and Prediction of Neurologic Survival*, *Improved Predictions in Structure-Based Drug Design Using CART and Bayesian Models*), de "servicios financieros y marketing" (*Insurance Premium Increase Optimization: Case Study, CART/ MARS*[39] *Risk Assessment of Automobile Loans and Leases*, *Insurance Fraud Detection: MARS vs. Neural Networks*, *Committee of Decision Trees Solution for Personal Bankruptcy Prediction*), y de "medio ambiente" (*Application of CART for Air Quality Forecasting*). En el Congreso sobre Data Mining que se celebrará este año 2006 podemos hallar ya aplicaciones más especializadas a la Ciencia Actuarial (*Experiences Applying Insurance Tasks to the New CART 6 Software: Customer Retention and Data Analytics*) y a la Logística (*Using Classification Trees to Explain Failure to Deliver On-Time*).

3.8 Resultados empíricos. Base de datos 2005

- Selección de factores de riesgo

La selección de factores mediante árboles de clasificación la hemos realizado con los datos correspondientes a la Cartera 2005, como ya advertimos en la Introducción del trabajo. Conviene recordar que existe un interés añadido en escoger los factores con estos datos, y es el de observar la influencia de la variable "nivel de Bonus Malus". Por ese motivo, se ejecutaron dos procesos de selección, uno con una muestra que no incluía éste factor, y otro con una que sí lo hacía.

Las variables más importantes se seleccionaron usando un "Random Forests"[40]. Se trata de una herramienta bastante nueva para el análisis de datos, que realiza una combinación aleatoria de clasificadores débiles (esto es, se escogen muestras y variables aleatoriamente para entrenar cada clasificador). Esos clasificadores suelen ser *árboles de decisión*, que no tienen influencia unos sobre otros en el momento de ser construidos. La suma de todas las decisiones hechas por los árboles determina la predicción completa del "bosque". Al escoger variables y muestras de modo aleatorio, es posible obtener la eficacia de cada variable a la hora de separar las clases. Este hecho promediado para todos los árboles, donde las variables se seleccionan, como digo, de manera aleatoria, permite obtener rápidamente una medida de la utilidad de la variable.

La aleatoriedad se presenta de dos modos en el RandomForests; una a nivel de los árboles, y otra al de los nodos. En el caso de los árboles, la

[39] MARS son las siglas de otro método de aprendizaje, *Multivariate Adaptive Regression Splines.*

[40] Aunque más que un Árbol de Clasificación es una colección de ellos (un bosque), siempre que nos refiramos a esta parte del experimento, lo haremos como a Árbol.

aleatorización tiene lugar vía las observaciones. A nivel de los nodos, sin embargo, ocurre empleando un subconjunto de predictores seleccionados aleatoriamente. Se permite crecer a cada árbol hasta un tamaño máximo, sin ejecutar ninguna poda. Se repite este proceso hasta que se crean los árboles que el usuario haya determinado, y a ese conjunto es al que llamamos "bosque aleatorio". Una vez que lo tenemos, se utilizan las predicciones para cada árbol en un proceso de "votación". La predicción total se determina votando por clasificación y haciendo una media por regresión. De este modo, para clasificar un objeto nuevo desde un vector de entrada, hacemos "descender" el vector por cada uno de los árboles del bosque. Cada árbol da, como decimos, una clasificación, "vota" por esa clase en concreto. El bosque asignará al objeto la clase más votada.

Por último, en el RF no se precisa del método de validación cruzada que utilizábamos hasta ahora para obtener una estimación insesgada del error del conjunto de Tes. En este caso, se estima internamente, durante la ejecución.

El software utilizado para los árboles es el llamado CART (Classification and Regresion Trees). Como ya sabemos, existen varios modos de hacer crecer los árboles de decisión. El que emplea CART es binario, dividiendo cada nodo padre en dos descendientes, colocando preguntas con respuestas *si / no* en cada nodo de decisión. Además, busca esas preguntas de modo que dividan los nodos en descendientes que sean relativamente homogéneos. Incorpora además un método automático de testeo y validación del árbol.

En nuestro problema, la idea es seleccionar los factores de riesgo que son más discriminativos, esto es que separan mejor los datos en los nodos de los árboles. Para medir la importancia de una variable se permutan sus valores y se mira cuánto error más se comete por no usar los valores "buenos". Si el error al colocar los valores erróneos en un factor es grande, ese factor será seleccionado. De este modo, se van tomando decisiones en el árbol de modo que en cada nodo del árbol se decide con una variable.

Los resultados que obtuvimos son los que se muestran a continuación. Junto al factor seleccionado aparece un valor numérico que indica la importancia relativa que asigna el RF a esa variable.

Sin nivel de Bonus Malus	
Barcelona	*45.8380*
ANTEVÍ	*31.2240*
ámbito8	*30.8260*
ámbito9	*28.2190*
Carnet	*27.5340*
uso1	*27.5140*
TIPO9	*23.8380*
Edad	*22.1510*
Tara	*21.4760*
ámbito10	*20.0310*
Aragón	*18.5590*
Andalucía	*16.1720*
tipo13	*15.9550*
CV	*13.9670*
Diesel	*13.9080*
Castilla y León	*13.5800*
Plazas	*10.9120*
Privado	*10.0080*
Cataluña	*9.2500*
uso19	*8.9560*
uso7	*8.7900*
Galicia	*8.3710*
tipo14	*7.1480*
Asturias	*4.4950*
Comunidad Valenciana	*1.9270*

Con nivel de Bonus Malus

B 109	107 .8240
Carnet	65.4550
Barcelona	48.2820
Edad	48.1710
ámbito 8	47.1090
ANTEVÍ	43.1090
B 116	42.5970
TARA	42.3360
ámbito 9	41.3570
uso 1	38.4670
TIPO 9	36.8170
ámbito 10	29.3960
B 110	28.5630
tipo 13	27.9960
Andalucía	25.7540
Plazas	25.2830
B 119	23.8490
B 121	23.1880
CV	22.9840
Galicia	21.2720
Aragón	20.8040
Cataluña	20.4460
Diesel	19.7060
B 36	17.7030
B 120	17.0260
B 118	16.5940
B 56	16.2080
uso 19	15.8140
tipo 14	14.7140
B 113	11.4270

Es interesante observar la importancia de los factores de riesgo representados por los niveles de BM. En concreto, el primer factor seleccionado (B109) se corresponde con el nivel de BM de máxima bonificación, en el que se encuentran, probablemente, los mejores conductores. La importancia relativa asignada por el árbol a dicho factor (107.8240) se encuentra, además, a gran distancia de las asignadas a los restantes factores (el siguiente factor en importancia, la antigüedad de carnet, tiene asociado el valor 65.4550). Es interesante observar también que gran parte de los restantes niveles de BM han resultados seleccionados asimismo como factores relevantes del riesgo.

Estos hechos nos permiten resaltar la importancia del nivel de BM como factor de riesgo en la tarificación a priori y sugieren interesantes relaciones entre ambos tipos de tarificación a priori y a posteriori.

4. FUTURAS LÍNEAS DE INVESTIGACIÓN

Este nuevo campo de actuación, el actuarial, ofrece muchas expectativas para la aplicación de las técnicas de aprendizaje, así que las líneas de investigación que se abren ante nuestros ojos son numerosas.

En el problema concreto de clasificación de asegurados, quizás la más atractiva sea la de dar el paso a la clasificación multiclase. Si bien no podemos esperar grandes resultados frente a los ya obtenidos, lo natural es que ahora nos planteemos clasificar los asegurados en clases según su riesgo para así poder ser capaces de calcular la prima a pagar en cada una de ellas. Para ello nos resultará muy útil la información que recogimos para los datos de 2005 en cuanto al número de siniestros por conductor, aunque estamos estudiando otras vías en cuanto al modo de i9ncluirla en el modelo.

Otra cuestión muy interesante, en la que de hecho estamos trabajando al término de este trabajo, es una clasificación de los asegurados atendiendo al riesgo de presentar o no siniestro según el factor EDAD. Como ya hemos comentado, este es uno de los factores de riesgo que más interesa a la compañía aseguradora. Hemos adoptado el método de “uno frente al resto”, de manera que vamos dividiendo las edades por tramos y construyendo clasificadores binarios en cada uno de ellos. También parece atrayente, dentro del ámbito del seguro, el extender el estudio a otras coberturas.

Además, está la cuestión de interpretar los factores que se obtienen con el Análisis factorial, y poder así comparar con los que se consiguen con el AG y El AC. Por supuesto, en el tema de los factores, está el interesantísimo tema de la influencia de los niveles de Bonus Malus, de los que se podrían sacar mechas conclusiones.

En cuanto a la comparación de los resultados con las técnicas estadísticas, estamos investigando en aplicar una herramienta más eficiente (*logit*) para contrastar con lo obtenido por la SVM. En el ámbito de las herramientas de aprendizaje, creemos que sería bueno probar con otros kernel en la SVM, y dar un paso más, empleando nuevas técnicas como el *filter method*, o la *programación genética con árboles*.

Para terminar, y ya volviendo a las bases de datos utilizada en esta investigación, los nuevos experimentos deberían dirigirse a alterar el número de datos utilizados en los experimentos, así como a la distribución de los mismos en cada clase.

APÉNDICE
EL ANÁLISIS FACTORIAL

1. INTRODUCCIÓN

Siguiendo a la Dra. De Vicente en [De Vicente et al 2000][41], el método de análisis factorial es un procedimiento matemático mediante el cual se pretende reducir la dimensión de un conjunto de p variables obteniendo un nuevo conjunto de variables más reducido pero capaz de explicar la variabilidad común encontrada en un grupo de individuos sobre los cuales se han observado las p variables originales.

El análisis factorial tiene como objeto simplificar las numerosas y complejas relaciones que se puedan encontrar en un conjunto de variables cuantitativas observadas. Para ello trata de encontrar dimensiones o factores que ponen en relación a las aparentemente no relacionadas variables.Se trata por tanto de encontrar las variables fundamentales que intervienen en la explicación de ciertos fenómenos. Según este planteamiento, podría pensarse que se trata de un método equivalente al del Análisis en Componentes Principales, pero veremos que no es así. La diferencia principal estriba en que en el caso del Análisis Factorial la búsqueda de variables fundamentales o factores comunes está encaminada a un objetivo final que es el de encontrar relaciones matemáticas que nos permitan expresar las variables originales a través de los factores comunes más los factores específicos de cada variable observada. En este sentido veremos como el Análisis en Componentes Principales se introduce, junto con otros posibles métodos, en una de las etapas del análisis factorial: la de la obtención de los factores comunes.

Una cuestión fundamental será la interpretación y el significado de los factores. Los resultados de un análisis factorial simplemente ponen de manifiesto un conjunto de factores comunes, el significado de estos factores se deduce de las cargas factoriales. Sin embargo, debe remarcarse que toda interpretación de los factores basada en las cargas deberá ser validada por algún criterio externo. Esta será una de las líneas abiertas de investigación que plantearemos para nuestro problema, ya que recordemos que nosotros no empleamos el análisis de características para seleccionar, sino para pode completar el análisis discriminante que se efectuó en el Capítulo anterior.

[41] Nos basaremos en las investigaciones de la Dra. De Vicente durante todo el Apéndice.

En la aplicación del análisis factorial, podemos diferenciar entre lo que se conoce como Análisis Factorial Exploratorio y el llamado Análisis Factorial Confirmatorio.

En Análisis Factorial Exploratorio el objetivo es explorar los datos para descubrir las dimensiones fundamentales. Este fue el propósito de Spearman cuando lo desarrolló en el año 1904. De este modo, en sus orígenes, el análisis factorial fue simplemente un método exploratorio. Sin embargo, se ha hecho posible contrastar hipótesis usando el análisis factorial a través del método desarrollado por Joreskog en 1973, y que no es otro que el mencionado Análisis Factorial Confirmatorio. En este caso, se hacen hipótesis sobre las cargas factoriales basándose en estudios previos o en una determinada teoría. El análisis factorial confirmatorio procede entonces a ajustar esas cargas tanto como sea posible. Además se puede medir la calidad del ajuste.

Nosotros describiremos sólo el análisis factorial exploratorio.

2. PLANTEAMIENTO DEL PROBLEMA DE ANÁLISIS FACTORIAL

En el planteamiento del problema de AF es importante el estudio, por un lado, del modelo matemático de AF y, por otro, de los métodos de obtención de factores. Comenzaremos viendo como establecer el modelo matemático de AF.

2.1 El Modelo Matemático del Análisis Factorial

Consideramos:

$$X_1, X_2, ..., X_p \equiv \textbf{Variables observadas}$$

$$F_1, F_2, ..., F_m \equiv \textbf{Factores comunes}$$

$$e_1, e_2, ..., e_p \equiv \textbf{Factores específicos}$$

Se supone que las variables observadas están tipificadas.

El modelo de análisis factorial se escribe de la siguiente forma:

$$X_1 = l_{11}F_1 + ... + l_{1m}F_m + e_1$$

$$X_2 = l_{21}F_1 + ... + l_{2m}F_m + e_2$$

… … …

$$X_p = l_{p1}F_1 + ... + l_{pm}F_m + e_p$$

donde l_{hj} es el peso del factor *h* en la variable *j*. Estos coeficientes se llaman *cargas factoriales*.

Las variables observables se escriben como combinación lineal de los factores comunes y de los factores específicos. Debe señalarse que tanto los factores comunes como los factores específicos son no observables.

En forma matricial el modelo quedaría de la siguiente forma:

$$\begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_p \end{bmatrix} = \begin{bmatrix} l_{11} \dots\dots\dots\dots l_{1m} \\ l_{21} \dots\dots\dots\dots l_{2m} \\ \dots\dots\dots\dots\dots\dots \\ l_{p1} \dots\dots\dots\dots l_{pm} \end{bmatrix} \begin{bmatrix} F_1 \\ \\ \dots \\ F_m \end{bmatrix} + \begin{bmatrix} e_1 \\ \\ \dots \\ e_p \end{bmatrix} \quad \Leftrightarrow \quad \mathrm{X} = \mathrm{Lf} + \mathrm{e}$$

Sobre este modelo deben hacerse las siguientes hipótesis:

A. Hipótesis sobre los factores comunes

$$\mathrm{E}[\mathrm{f}] = 0$$

La matriz de varianzas-covarianzas es la identidad: $E[ff^t] = I$
Los factores son variables tipificadas e incorreladas entre sí

B. Hipótesis sobre los factores específicos

$$\mathrm{E}[\mathrm{e}] = 0$$

Matriz de varianzas-covarianzas: $\mathrm{E}[\mathrm{ee}^{\mathrm{t}}] = \Omega$.
Donde Ω es una matriz diagonal.
Los factores específicos son incorrelados.

C. Hipótesis sobre la relación entre factores comunes y específicos

La matriz de varianzas-covarianzas entre los factores comunes y específicos es:

$$\mathrm{E}[\mathrm{fe}^{\mathrm{t}}] = 0$$

Los factores comunes y los específicos están incorrelados entre sí.

Teniendo en cuenta todo lo anterior, el modelo de análisis factorial presenta las siguientes propiedades:

1. La matriz de varianzas-covarianzas coincide con la matriz de correlaciones por estar éstas variables tipificadas.

$$E\left[XX^{t}\right] = Rp = \begin{pmatrix} 1 & \rho_{12} & \ldots\ldots & \rho_{1p} \\ \rho_{21} & 1 & \ldots\ldots & \rho_{2p} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \\ \rho_{p1} & \ldots\ldots & \ldots\ldots & 1 \end{pmatrix}$$

2. La matriz de correlación poblacional puede descomponerse en dos partes, una debida a los factores comunes y otra que coincide con la matriz de varianzas-covarianzas de los factores únicos. En concreto,

$$Rp = LL^{t} + \Omega \tag{1.1}$$

En efecto:

$$R_{\rho} = E\left[XX^{t}\right] = E\left[(Lf + e)(Lf + e)^{t}\right] = LE\left[ff^{t}\right]L^{t} + E\left[ee^{t}\right] + LE\left[fe^{t}\right] + E\left[ef^{t}\right]L^{t} =$$

$$= LIL^{t} + \Omega = LL^{t} + \Omega$$

3. La varianza de la variable poblacional X_j se descompone de la siguiente forma:

$$1 = h_j^2 + \omega_j^2$$

donde:

h_j^2, es la *comunalidad*, que se define como la parte de la varianza que es debida a los factores comunes.

$\omega_J^{\,2}$ es la parte de la varianza que es debida a los factores específicos.

Veámoslo:
Desarrollando 1.1 se tiene:

$$\begin{pmatrix} 1 & \rho_{12} & \ldots\ldots & \rho_{1p} \\ \rho_{21} & 1 & \ldots\ldots & \rho_{2p} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \\ \rho_{p1} & \ldots\ldots & \ldots\ldots & 1 \end{pmatrix} = \begin{pmatrix} l_{11} & \ldots\ldots & \ldots\ldots & l_{1m} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \\ l_{p1} & \ldots\ldots & \ldots\ldots & l_{pm} \end{pmatrix} \begin{pmatrix} l_{11} & \ldots\ldots & \ldots\ldots & l_{p1} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \\ l_{1m} & \ldots\ldots & \ldots\ldots & l_{pm} \end{pmatrix} + \begin{pmatrix} \omega_1^2 & 0 & \ldots\ldots & 0 \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \\ 0 & \ldots\ldots & \ldots\ldots & \omega_p^2 \end{pmatrix} \tag{1.2}$$

Es decir,

$$1 = l_{11}^2 + \ldots\ldots\ldots\ldots\ldots\ldots + l_{1m}^2 + \omega_1^2$$

$$1 = l_{j1}^2 + \ldots\ldots\ldots\ldots\ldots\ldots + l_{jm}^2 + \omega_j^2$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$1 = l_{p1}^2 + \ldots\ldots\ldots\ldots\ldots\ldots + l_{pm}^2 + \omega_p^2$$

Y si en $1 = l_{j1}^2 + \ldots\ldots\ldots\ldots\ldots\ldots + l_{jm}^2 + \omega_j^2$

Hacemos: $h_j^2 = l_{j1}^2 + \ldots\ldots\ldots\ldots\ldots\ldots + l_{jm}^2$

tenemos: $1 = h_j^2 + \omega_j^2$ (1.3)

4. Se puede reproducir la correlación entre las variables observadas a partir de las cargas factoriales a partir de la siguiente expresión:

$$\rho_{rs} = l_{r1} l_{s1} + \ldots\ldots\ldots\ldots\ldots\ldots + l_{rm} l_{sm} = \sum_{k=1}^{m} l_{rk} l_{sk}$$ [42]

Esta propiedad es de gran interés pues un buen test de la adecuación del análisis es reproducir las correlaciones y entonces restarlas de sus valores originales. El resultado es lo que se denomina ***matriz residual***. Cuanto más precisa sea la reproducción –por tanto cuanto menor sea la matriz residual- mejor será el análisis.

Resumiendo, el problema que plantea el análisis factorial es la estimación de los coeficientes l_{jk} .

A los coeficientes estimados se les llama *cargas factoriales estimas*, aunque se suele prescindir del adjetivo estimadas. Una vez obtenidas las cargas factoriales los factores comunes quedan determinados.

2.2. Los Métodos de Obtención de Factores

Cuando Spearman introdujo la técnica del análisis factorial tenían que usarse métodos muy simples de cálculo. Sin embargo, como ocurrió con las técnicas que hemos estado tratando hasta ahora, con el paso del tiempo y el avance de los ordenadores se han ido desarrollando diferentes métodos de obtención de factores.

[42] Esto se deduce directamente de 1.2

El método de *Análisis en Componentes Principales* es uno de los métodos de obtención de factores del análisis factorial. Los factores comunes obtenidos a través del método de análisis en componentes principales son, precisamente, las componentes principales.

El análisis en componentes principales es una técnica de reducción de la dimensionalidad. Su objetivo es explicar la mayor parte de la variabilidad total de un conjunto de variables cuantitativas con el menor número de componentes o factores comunes posible. Se trata de un método geométrico de carácter descriptivo. El objetivo es descubrir la estructura subyacente en un conjunto de n individuos estudiados bajo una serie de p variables cuantitativas.

Esta técnica permite transformar un conjunto de variables originales en otro conjunto de variables llamado ***conjunto de Componentes Principales***. Estas componentes principales son combinación lineal de las variables originales y se caracterizan por estar incorreladas entre sí.

En principio se generarán p componentes principales, es decir, tantas como variables originales, aunque veremos que normalmente no se necesitan todas para el problema que se esté resolviendo.

Si se toman las variables originales y se calcula su matriz de correlaciones se puede observar que, habitualmente, existe un alto grado de correlación entre algunas de las variables. Esto nos lleva a pensar que quizás podríamos trabajar sobre un conjunto de variables incorreladas de menor dimensión. Así, si las variables originales están muy correladas entre sí, es esperable que su información se pueda expresar a través de unas pocas componentes principales.

Si por el contrario las variables originales están muy poco correladas entre sí, el número de componentes principales será similar al de variables originales. De esta forma, la varianza será una "medida de la información" que contiene cada variable.

Veamos el planteamiento del problema: Sea Y la matriz de variables originales tipificadas[43]. Entonces *la primera componente principal es la combinación lineal de variables originales de varianza máxima.*

Por tanto, buscamos v_1 de norma uno tal que la varianza de la primera componente principal c_1 sea máxima. Se demuestra que la solución es $c_1 = Yv_1$ donde v_1 es el vector propio de la matriz de varianzas-covarianzas (la notaremos V_y) con mayor valor propio asociado.

[43] En los problemas de Análisis en Componentes Principales se suele trabajar con las variables originales tipificadas para evitar problemas de escalas

Una vez obtenida la primera componente principal el resto de componentes principales se obtienen de la siguiente forma: La segunda componente principal sería aquella combinación lineal de variables originales de varianza máxima y ortogonal a c_1. Al ser toda matriz de varianzas-covarianzas simétrica y semi-definida positiva, tendrá p vectores propios ortogonales dos a dos y sus valores propios asociados serán todos positivos o nulos.

Los vectores propios de la matriz V_y asociados a los valores propios escritos en forma decreciente serán los vectores buscados (los factores principales). Estos vectores nos permiten calcular las componentes principales a través de la expresión $c = Yv$. La varianza de cada las componentes principales viene dada por los valores propios.

El número de valores propios no nulos nos proporcionará la dimensión del espacio de las componentes principales.

En cuanto a la cuestión de cuántas componentes tomar, parece lógico por un lado escoger aquellas componentes cuyo valor propio asociado sea mayor que uno, es decir aquellas componentes que aportan más información que cualquiera de las variables originales tipificadas. Sin embargo hay que tener en cuenta algunas otras consideraciones tales como que las componentes que agrupan una gran cantidad de variables originales tienen una gran importancia. La experiencia y un buen conocimiento del contexto en el que se enmarca el estudio son casi siempre determinantes a la hora de la interpretación de los ejes.

Veamos finalmente la interpretación de las componentes principales. Es ésta una de las fases más complejas del Análisis en Componentes Principales.

En general, la interpretación de los ejes factoriales (de las componentes principales) se hace a través del estudio de las correlaciones entre las componentes principales y las variables originales.

Se demuestra que estos coeficientes de correlación se obtienen a través de la siguiente expresión:

$$r_{kj}^{*} = c_{kj}\sqrt{\lambda_k}$$

donde r_{kj}^{*} es el coeficiente de correlación lineal entre la k-ésima componente principal (c_k) y la j-ésima variable inicial tipificada (Y^j), y c_{kj} es la j-ésima coordenada de la k-ésima componente principal.

Se trata de observar por tanto el círculo de correlaciones. Aquellas variables originales cuya correlación con una componente principal dada esté muy próxima a 1 en valor absoluto, serán las que más contribuyan a la explicación de dicha componente principal.

Puede también ser de interés en algunas ocasiones representar el conjunto de individuos sobre el plano factorial y estudiar los que más contribuyen a la explicación de un determinado factor. Esto último no tiene sentido más que cuando los individuos tienen un interés en si mismos.

Además, una vez calculados los coeficientes c_{jk}, se pueden obtener las *puntuaciones factoriales*, es decir, los valores de las componentes correspondientes a cada observación:

$$C_{hi} = c_{h1}X_{1i} + c_{h2}X_{2i} + ... + c_{hp}X_{pi} \qquad h = 1,2,.....p \qquad i = 1,.2,.......n$$

Si una componente se divide por su desviación típica se obtiene una componente tipificada. De esta forma las puntuaciones factoriales tipificadas de la componente h-ésima serán:

$$\frac{c_{hi}}{\sqrt{\lambda_h}}$$

Pero volvamos al Análisis Factorial propiamente dicho. Hemos establecido que las componentes principales se expresan en función de las variables originales del modo:

$$C_1 = c_{11}X_1 + c_{12}X_2 + ... + c_{1p}X_p$$

$$C_2 = c_{21}X_1 + c_{22}X_2 + ... + c_{2p}X_p$$

$$.......... ....$$

$$C_p = c_{p1}X_1 + c_{p2}X_2 + ... + c_{pp}X_p \qquad (1.4)$$

Se puede demostrar que este sistema es reversible. En consecuencia se tendrán las siguientes expresiones de las variables originales en función de las componentes:

$$X_1 = c_{11}C_1 + c_{21}C_2 + ... + c_{p1}C_p$$

$$X_2 = c_{12}C_1 + c_{22}C_2 + ... + c_{p2}C_p$$

$$.......... ....$$

$$X_p = c_{1p}C_1 + c_{2p}C_2 + ... + c_{pp}C_p \qquad (1.5)$$

Ahora bien, la utilización del sistema anterior como base para el cálculo de los factores plantea el problema de que las componentes *C* no están tipificadas (recordemos que esta era una de las hipótesis que se planteaban en el modelo factorial). Esta dificultad puede solventarse usando las componentes

tipificadas, que como se ha visto tenían la siguiente forma:

$$Z_h = \frac{C_h}{\sqrt{\lambda_h}} \qquad h = 1,2,.....p \tag{1.6}$$

De estas expresiones, y desarrollando los oportunos cálculos, se sigue que:

$$X_j = r_{1j}Z_1 + r_{2j}Z_2 + ... + r_{mj}Z_m + (r_{m+1,j}Z_{m+1} + ... + r_{pj}Z_p) \tag{1.7}$$

siendo r_{ij} el coeficiente de correlación entre la variable j-ésima y la componente h-ésima.

Pero recordemos que la ecuación de la variable j-ésima en el modelo factorial se escribía del siguiente modo:

$$X_j = l_{j1}F_1 + l_{j2}F_2 + ... + l_{jm}F_m + e_j$$

Comparando estas dos expresiones, podemos ver que los m factores F_h se estiman mediante las *m* primeras componentes principales tipificadas Z_h y además se tienen las estimaciones de los coeficientes l_{jh} :

$$l^*_{j1} = r_{1j}$$

$$l^*_{j2} = r_{2j}$$

.....

$$l^*_{jm} = r_{mj}$$

Y gracias a las estimaciones anteriores se obtiene la siguiente estimación de la comunalidad:

$$h_j^{2*} = l_{j1}^{2\,*} + l_{j2}^{2\,*} + ... + l_{jm}^{2\,*}$$

y del factor único e_j : $e_j* = r_{m+1,1}Z_{m+1} + ... + r_{p1}Z_p$

Finalmente, la especificidad, es decir la parte de la varianza debida al factor único, será:

$$\omega_j^2* = 1 - h_j^{2\,*}$$

Para terminar, y aunque sea una cuestión de matiz, pensamos que es

interesante hacer una distinción entre componentes y factores (aunque como indica Harman (1976) esta distinción es trivial cuando se trata de grandes conjuntos de datos). Las componentes son factores reales porque se derivan directamente de la matriz de correlación. Los factores comunes del análisis factorial son hipotéticos porque son estimados a partir de los datos.

Visto el método de análisis en componentes principales pasemos a ver algunos otros métodos de obtención de factores comunes. No haremos un repaso exhaustivo de los métodos sino que simplemente mencionaremos algunos de los que nos han parecido más interesantes.

- *Máxima verosimilitud*

El método de máxima verosimilitud obtiene a través de factorizaciones sucesivas un conjunto de factores cada uno de los cuales explica a su vez tanta varianza como sea posible en la matriz de correlaciones de la población, como estimaciones de la matriz de correlación muestral.

Esta es una diferencia fundamental con el análisis en componentes principales que explica la mayor varianza posible en la matriz observada o muestral. Por esta razón el método de máxima verosimilitud es considerado como un método estadístico porque, como en el caso de la estadística inferencial, las inferencias se hacen de la muestra a la población. Obviamente, se requieren muestras grandes y es preciso hacer la hipótesis de que las variables proceden de una normal multivariante. Cuando los test de fiabilidad y las comunalidades son altas la diferencia entre el análisis factorial por máxima verosimilitud y el análisis por componentes principales es trivial, como de hecho es el caso entre el análisis en componentes principales y el análisis por ejes principales.

El mayor argumento en favor del uso del análisis por máxima verosimilitud reside en el hecho de que tienen test estadísticos para la significación de cada factor extraído.

- *Ejes principales*

Es parecido al de componentes principales, pero supone un procedimiento iterativo. La diferencia es que en lugar de encontrar unos en la diagonal de la matriz de correlación las comunalidades son estimadas. Para ello, se obtiene una primera estimación de las comunalidades calculando la regresión de cada variable mediante el coeficiente de determinación. A continuación se sustituye en la matriz de correlaciones muestral cada 1 de la diagonal principal por la estimación de la comunalidad correspondiente a cada variable. A esta matriz se le denomina matriz de correlación reducida. Trabajando sobre la matriz de correlación reducida, se calculan los autovalores, los autovectores y se determina el número de factores a retener.
Las comunalidades son ahora calculadas como la suma de los cuadrados de

las cargas factoriales. La estimación de las varianzas debidas al factor único se obtendrá entonces como la diferencia entre 1 y la comunalidad de la variable.

El proceso se itera siempre que la varianza debida al factor único sea positiva o, más concretamente, mayor que un valor positivo suficientemente pequeño establecido a priori.

- *Método alfa*

Se considera que las variables del estudio constituyen una muestra de un universo de potenciales variables, observadas sobre una población dada de sujetos. De este modo, la elección de las variables es aleatoria mientras que los individuos, en tanto que población, dan lugar a los valores fijos de los parámetros que los caracterizan. Este método, según sus autores, se basa en el principio de que los pesos factoriales se fijan de tal modo que los factores comunes extraídos tienen correlaciones máximas con el universo de los factores comunes existentes. Los factores específicos serían errores introducidos por el muestreo aleatorio. Por consiguiente, las comunalidades estimadas se consideran como "fiabilidades" en un contexto de medición.

- *Mínimos cuadrados no ponderados*

El criterio que se aplica en este método es la minimización de las sumas de las diferencias al cuadrado entre los elementos de las matrices de correlación observada y reproducida, aunque sin tener en cuenta la diagonal principal.

- *Mínimos cuadrados generalizados*

Este método aplica el mismo criterio que el método anterior, pero ponderando las correlaciones con la inversa de la especificidad de las variables.

2.3. La Rotación de Factores

Una vez completado el análisis factorial, por cualquiera de los métodos vistos, los factores deben ser interpretados e identificados. Para esto las soluciones deben ser rotadas ya que, por un lado, factores con muchas cargas factoriales son difíciles de interpretar y, por otro, no existe una solución perfecta o ideal en análisis factorial. Toda rotación ortogonal de una solución es también solución:

Sea H una matriz ortogonal y L una solución del análisis factorial

Entonces B=LH también es una solución: $BB^t = LHH^tL^t = LL^t$,

Ya que por ser H ortogonal $HH^t = I$.

Básicamente existen dos tipos de rotaciones, las rotaciones ortogonales y las rotaciones oblicuas.

En las rotaciones ortogonales los factores se rotan de tal modo que los ángulos entre ellos sean siempre ángulos rectos. La rotación ortogonal de factores hace variar las cargas factoriales y por tanto el significado de los factores, pero las diferentes soluciones factoriales analíticas son matemáticamente equivalentes ya que explican la misma cantidad de varianza en cada variable y por tanto en el conjunto de la matriz.
Además los factores rotados reproducen las correlaciones de forma precisa al igual que las soluciones no rotadas.

Dentro de los métodos de rotación ortogonal comentaremos las rotaciones Varimax, Quartimax y Equimax.

- *Método Varimax*

El objetivo es minimizar el número de variables que tienen cargas altas en un factor, y para ello se maximiza la suma de las varianzas de las cargas factoriales dentro de ese factor dejando por columna cantidades próximas bien a 1, bien a 0.

- *Método Quartimax*

En este caso el objeto de la rotación es conseguir que una variable tenga una carga alta en un factor y baja en los demás, de esta manera se obtendrá un factor general con cargas altas en todas las variables. En este caso una medida numérica de la simplicidad es la variabilidad aportada por los cuadrados de los pesos factoriales para cada variable. Puesto que la varianza es el promedio de las desviaciones al cuadrado respecto de la media, la varianza mayor (para un número fijo de factores y una comunalidad fija) será para la variable cuyo peso al cuadrado iguale a la comunalidad, con ceros para todos los demás factores columnas.

- *Método Equimax*

Es una solución intermedia entre las anteriores, se pueden combinar ambos métodos y asignar pesos a cada criterio.

Las rotaciones ortogonales son las adecuadas cuando el interés principal es reducir la información y sobre todo si se quiere que los factores estén incorrelados. Obviamente, debido a que existen una infinidad de soluciones factoriales matemáticamente equivalentes, es preciso tener algún criterio para la elección de la solución factorial. Este criterio es la “Ley de Parsimonia”. La ley de parsimonia establece que se deberá tomar la explicación más simple de

aquellas que sean coherentes con los datos, esto es lo que se conoce como la rotación a una estructura simple.

Thurstone (1947) propuso cinco criterios para decidir sobre la estructura simple. Estos cinco criterios son los siguientes:

1. Todas las filas de la matriz factorial deben tener al menos un cero.

2. Si existiesen k factores comunes, todas las columnas de la matriz factorial deberían tener al menos k ceros.
3. Para cada par de columnas de la matriz factorial deben acumularse muchas variables en una columna pero no en la otra.

4. Para cada par de columnas de la matriz factorial deben acumularse una gran proporción de variables en las dos columnas cuando haya cuatro o más factores.

5. Para cada par de columnas de la matriz debe haber un pequeño número de variables que se anulen en las dos columnas.

Cuando Thurstone propone la idea de una estructura simple supone que ésta se adaptará exclusivamente a sistemas de ejes ortogonales, es decir, que los factores resultantes no estarán correlacionados entre sí. Sin embargo, no siempre puede probarse la existencia de correlaciones nulas entre los factores y resulta más fácil buscar una estructura simple cuando no se impone la condición de ortogonalidad de los factores.

En las rotaciones oblicuas los ejes factoriales pueden tomar cualquier posición en el espacio factorial. El coseno del ángulo entre los ejes factoriales indica la correlación entre ellos. Las rotaciones oblicuas permiten más libertad en la selección de las posiciones de los factores en el espacio factorial al no existir la restricción de ortogonalidad.

Es importante comentar sin embargo algunos de los problemas que pueden surgir con las rotaciones oblicuas:

1. La suma de los cuadrados de las cargas factoriales en una cualquiera de las filas no suele ser igual a la comunalidad. En consecuencia, las cargas oblicuas no indican claramente la proporción de varianza para cada variable explicada por los factores.

2. Las correlaciones originales entre variables no pueden ser reproducidas a partir de productos de las cargas factoriales.

En general se identifican dos tipos de enfoques para la rotación oblicua: uno que utiliza ejes de referencia y otro que emplea el modelo de matriz primaria (modelo factorial de los factores primarios).

- *Rotaciones oblicuas basadas en los ejes de referencia*. se apoyan en la ida de la existencia de grupos definidos de variables que representan dimensiones separadas y correctamente identificadas por factores primarios. Cada grupo de variables tendría, por tanto, una proyección cercana a cero en todos los ejes de referencia menos uno. Este criterio se conoce como criterio *quartimin.*

- *Rotaciones Oblicuas basadas en el modelo de los factores primarios: Oblimín directo.* Se apoyan en la simplificación de los pesos factoriales sobre los factores primarios y no sobre los ejes de referencia.

3. RESULTADOS DE LA SELECCIÓN CON ANÁLISIS FACTORIAL

Veamos ahora detenidamente el resultado de la aplicación del Análisis Factorial en nuestro problema. Para ello, tomamos la misma muestra de datos que se escogió para el experimento con el Algoritmo Genético, esto es, los 58238 clientes seleccionados aleatoriamente de la base de datos enlazada de la Cartera de Clientes 2003 y la de Siniestros 2003 que se describió en el Capítulo 2.

Por otra parte, la herramienta estadística que elegimos para comparar los resultados de la Clasificación, el Análisis Discriminante, es, como ya hemos visto, una técnica que se aplica solamente a variables cuantitativas viniendo caracterizados los grupos a través de una variable categórica. En nuestro caso ésta será la variable siniestros que toma valores -1 y 1.

Nuestras variables son en su mayor parte de tipo cualitativo. De forma concreta, la base de datos está divida de la siguiente forma:

- *Variables cuantitativas (o continuas):* Antigüedad Carnet, Edad del Conductor, Antigüedad Vehículo, Potencia y Valor.

- *Variables cualitativas (o nominales):* el resto.

Es necesario por tanto, para aplicar un análisis discriminante, transformar previamente el conjunto de variables en variables cuantitativas/continuas, para lo que también nos será útil la aplicación del Análisis Factorial.

Para ello se procederá de la siguiente forma:

1. Se realizará un Análisis Factorial Múltiple (AFM) sobre el conjunto completo de las variables.

2. Selección de aquellos factores resultantes del AFM que expliquen el 100% de la variabilidad total.

3. Realización del Análisis Discriminante (AD) con los factores anteriormente seleccionados[44].

El AFM es una técnica de análisis factorial que trata tablas en las cuales un conjunto de individuos viene descrito por varios grupos de variables. En el seno de un mismo grupo, las variables deben ser del mismo tipo (continuo o nominal) pero de un grupo a otro, las variables pueden ser de diferentes tipos. Nosotros llevaremos a cabo un AFM sobre el conjunto de individuos descritos a través de dos tablas, la primera formada por las variables continuas y la segunda formada por las variables nominales.

El método es en realidad un análisis factorial del conjunto de grupos (llamado análisis global). Para las variables continuas, el AFM se comporta como un Análisis en Componentes Principales, para las variables nominales como un Análisis de Correspondencias Múltiples. Es la introducción de pesos en las variables, que equilibran las inercias axiales máximas de los grupos, lo que hace posible la presencia simultánea de variables continuas y nominales.

El objetivo final es obtener los principales factores de variabilidad de los individuos, estando estos descritos de forma equilibrada por varios grupos de variables.

Para la realización del AFM se utilizó el software SPAD[45]. Se trata de una herramienta muy conocida que se emplea para el Análisis de Datos, Data Minino, y la gestión de Calidad de Datos.

De los factores obtenidos con el AFM se seleccionaron los 59 primeros ya que estos explican el 100% de la variabilidad total, como se observa en la siguiente tabla:

[44] Los correspondientes resultados aparecen en el Capítulo dedicado a Clasificación

[45] Más información en http://www.spad.eu/

ANÁLISIS GLOBAL

VALORES PROPIOS
Valores de precisión de los cálculos: traza antes de la diagonalización 28.7461
Suma de valores propios: 28.7461
Histograma de los primeros 138 valores propios

Nº	Valor propio	%	% acumulado	Varianza
1	1.3546	4.71	4.71	**********
2	0.9915	3.45	8.16	**********
3	0.9536	3.32	11.48	**********
4	0.6647	2.31	13.79	**********
5	0.6335	2.20	16.00	**********
7	0.5826	2.03	2.03	**********
8	0.5690	1.98	22.14	**********
9	0.5565	1.94	24.08	**********
10	0.5449	1.90	25.97	**********
11	0.5417	1.88	27.86	**********
12	0.5324	1.85	29.71	**********
13	0.5261	1.83	31.54	**********
14	0.5217	1.81	33.35	**********
15	0.5136	1.79	35.14	**********
16	0.5126	1.78	36.92	**********
17	0.5098	1.77	38.70	**********
18	0.5057	1.76	40.45	**********
19	0.5045	1.76	42.21	**********
20	0.5033	1.75	43.96	**********
21	0.4993	1.74	45.70	**********
22	0.4993	1.71	47.41	**********
23	0.4895	1.70	49.11	**********
24	0.4883	1.70	50.81	**********
25	0.4865	1.69	52.50	**********
26	0.4820	1.68	54.18	**********
27	0.4800	1.67	55.85	**********
28	0.4796	1.67	57.52	**********
29	0.4788	1.67	59.18	**********
30	0.4780	1.66	60.84	**********
31	0.4774	1.66	62.50	**********
32	0.4769	1.66	64.16	**********
33	0.4758	1.66	65.82	**********
34	0.4752	1.65	67.47	**********
35	0.4740	1.65	69.12	**********
36	0.4729	1.64	70.77	**********
37	0.4718	1.64	72.41	**********
38	0.4709	1.64	74.05	**********
39	0.4691	1.63	75.68	**********
40	0.4630	1.61	77.29	**********
Nº	Valor		%	Varianza

	propio	%	acumulado	
41	0.4616	1.61	78.89	**********
42	0.4601	1.60	80.49	**********
43	0.4565	1.59	82.08	**********
44	0.4544	1.58	83.66	**********
45	0.4496	1.56	85.23	**********
46	0.4455	1.55	86.78	**********
47	0.4378	1.52	88.30	**********
48	0.4348	1.51	89.81	**********
49	0.4272	1.49	91.30	**********
50	0.4155	1.45	92.74	**********
51	0.3960	1.38	94.12	**********
52	0.3765	1.31	95.43	**********
53	0.3527	1.23	96.66	**********
54	0.3389	1.18	97.84	**********
55	0.3031	1.05	98.89	**********
56	0.1600	0.56	99.45	**********
57	0.1548	0.54	99.99	**********
58	0.0020	0.01	99.99	*
59	0.0013	0.00	100.00	*

En ella aparecen los siguientes datos:

Nº: número de orden del factor 1,2,3,……
Valor propio : el valor propio de dicho factor
% : porcentaje, tanto por ciento de la varianza total que explica el factor
% acumulado: tanto por ciento de la varianza total explicada por ese factor y los anteriores (tanto por ciento acumulado)
**********: representación gráfica de la varianza explicada por cada factor

Así, observando la última línea tenemos lo siguiente:

Nº: 59 (factor nº 59)
Valores propios: 0.0013
%: 0.00 (con una precisión de dos dígitos ese factor no explica ya prácticamente nada)
% acumulado: 100.00 (el tanto por ciento total de la varianza explicada por los 59 factores es del 100 %)

BIBLIOGRAFÍA

[Altman et al. 1994] Altman E. I., Marco, G. and Varetto, F. (1994) *Corporate distress diagnosis: comparisons using discriminant analysis and neural networks (the italian experience)*, Journal of Banking and Finance, 18:505-529.

[Andenberg 1973] Andenberg , M. R. (1973). *Cluster Analysis for Applications* Academic Press. New York.

[Ambrose et al. 1994] Ambrose, J. M., Carol, A. M. (1994) *Using best ratings in life insurer insolvency prediction*, Journal of Risk and Insurance, 61:317-327.

[Bäck 1994] Bäck, T. (1994). *Selective pressure in evolutionary algorithms: A characterization of selection mechanisms*. In Proceedings of the 1st IEEE Conf. on Evolutionary Computation, pages 57–62, IEEE Press, New York, NY, USA.

[Bäck et al 1998] Bäck, T., Eiben A. E., and Vink, M. E. (1998). *A superior evolutionary algorithm for 3-SAT.* J. Complex Systems, 1(1):39–66.

[Bannister (1997)] Bannister, J. (1997) *Insurance solvency analysis*, LLP limited, second edition.

[Barniv(1990)] Barniv, R. (1990). *Accounting procedures, Market data, cash-flow figures and insolvency classification: the case of the insurance industry*, The Accounting Review, 65(3):578-604.

[Battle 1991] Battle D. L. and Vose, M. D. (1991). *Isomorphisms of genetic algorithms.* In J.E. Rowling, editor, Foundations of Genetic Algorithms 1, pages 242–251, Morgan Kaufmann, San Mateo, CA, USA.

[Bishop (1995)] Bishop, C. M. (1995) *Neural Networks for pattern recognition*, Oxford University press.

[Boj (2003)] Boj del Val, E. (2003) *Análisis multivariante aplicado a la selección de factores de riesgo en la Tarificación* Tesis Doctoral defendida en el Departament de Matemàtica Econòmica, Financera i Actuarial de la Universitat de Barcelona.

[Boj et al 2004] Boj del Val E., Claramunt Bielsa M.M., Fortiana Gregori J. (2004) Análisis multivariante aplicado a la selección de factores de riesgo en la tarificación Cuadernos de la Fundación MAPFRE Estudios, Nº. 88

[Boser et al., 1992] Boser, B.E., Guyon I.M., y Vapnik, V. (1992). *A training algorithm for optimal margin classifiers*. En D. Haussler Ed. Proceedings of the 5th Annual Workshop on Computational Learning Theory, 144-152. Pittsburg, PA. ACM Press.

[Bredensteiner y Bennett 1999] Bredensteiner, E.J. y Bennett, K.P. (1999). *Multicategory Classification by Support Vector Machines.* Computational Optimization and Applications **12** 35--46

[Breimann et al 1984] Breiman L., Friedman, J.H., Olshen, R.A., Stone, C.J. (1984). *Classification and regression trees*

[Brockett et al 1993] Brockett P.L., Cooper W.W., Golden L.L., Pitaktong U. *A Neural Network Method for Obtaining an Early Warning for Insurer Insolvency*. The Jornal of Risk and Insurance, Vol. 61, **3** pp 402-424.

[Brockett et al 1998] Brockett, P.L., Xiaohua Xia and Derring, R. A. (1998). *Using Kohonen's Self-Organizing Feature Map to Uncover Automobile Bodily Injury Claims Fraud*. Journal of Risk and Insurance, Vol. 65, No 2, 245.

[Brooker 1987] *Improving search in GA*. (1987). In L.Davis (Ed.) Genetic algorithms and Simulated Annealing. Morgan Kaufmann, Los Altos, CA, 61-73.

[Burges (1998)] Burges, C. J. (1998): *A tutorial on Support Vector Machines for pattern recognition*, Knowledge Discovery and Data Mining, 2(2):121-167.

[Cantu-Paz, E. 1999] Cantu-Paz, E. (1999). *Migration policies, selection pressure and parallel evolutionary algorithms*. IlliGAL Report No. 99015, University of Illinois, Urbana, Illinois.

[Chen (1996)] Chen SH, Yeh CH. (1996) *Using genetic programming to model volatility in .nancial time series*. In: Koza J, et al., editors. Proceedings of the Second Annual Conference on Genetic Programming pp. 58–63.

[Cherkassky y Mulier 1998] Cherkassky V. Mulier F. (1998) *Learning from data. Concepts, Theory and Methods.*. John Wiley &Sons.

[Cortes y Vapnick, 1995] Cortes C., Vapnik V. (1995) *Support Vector Networks*. Machina Learning, 20. 273-297, 1995.

[Crisp y Burges 2000] Crisp D.J., Burges C.J.C.(2000) *A geometric interpretation of u-SVM classifiers.* In S.A Solla, T.K. Leen, K.R. Müller ed. Advances in Neural Information Processing Systems 12 MIT Press.

[Cristianini y Shawe-Taylor 2000] Cristianini N, Shawe Taylor J. (2000) *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press.

[Davis (Ed) 1991] Davis L.(Ed) (1991) *Handbook of GA* Van Nostrand Reinhold, New York.

[De Jong 1975] De Jong K.A. (1975) *An analysis of the behavior of a class of genetic adaptive systems*. Doctoral dissertation, University of Michigan, Ann Arbor, Michigan.

[De Jong 1992] De Jong, K. A. and Spears, W. M. (1992). *A formal analysis of the role of multi-point crossover in genetic algorithms*. Annals of Mathematics and Artificial Intelligence, 5(1):1–26.

[De Vicente et al 2000]De Vicente y Oliva, M., Manera Bssa, J., Blanco Jiménez, F. J. (2000) *Análisis Multivariante para las Ciencias Sociales*. Ed.: Dykinson. Madrid.

[de Wit 1986] de Wit, G. W. (1986). *Risk Theory, a Tool for Management.* M Goovaerts et al. eds., Insurance and Risk Theory, Reidel, Dordrecht-Boston, MA, pp 7-17.

[Dimitras et al. (1996)] Dimitras, A. I., Zanakis, S. H., and Zopounidis, C. (1996): *A survey of busines failures with an emphasis on prediction methods and industrial applications*. European Journal of Operational Research, 90(3):487-513.

[Dimitras et al. (1998)] Dimitras, A. I., Slowinski, R., Susmaga, R. and Zopounidis, C. (1998): *Business failure using rough set.* European Journal of Operacional Research, 114(2):263-280.

[Duan y Keerthi 2005] Kai-Bo Duan, S. Sathiya Keerthi.(2005) *Which Is the Best Multiclass SVM Method? An Empirical Study.* Multiple Classifier Systems. pp 278-285.

[Duda et al 2001] Duda, R.O., Hart, P.E. Stork, D.G. (2001) *Pattern Classification*. John Wiley & Sons

[Eshelman et al. 1989] Eshelman L.J., Caruana R.A. Schaffer, J.D. (1989). *Biases in the crossover landscape*. En J.D. Schaffer Ed. Proceedings of the 3rd International Conference on GA, Morgan Kaufmann, Los Altos, CA, 10-19.

[Field, P. 1994] Field, P. (1994). *Nonbinary transforms for genetic algorithms problems*. InFogarty, T. C., editor, Proceedings of Artificial Intelligence and Simulation of Behaviour Workshop, pages 38–50, Springer Verlag, Berlin, Germany.

Fisher 1952] Fisher R. A.(1952) *Contributions to Mathematical Statistics.*. Wiley

Fogarty 1989] Fogarty T.C. (1989) *Varying the probability of mutation in the genetic algorithm*. En J.D. Schaffer Ed. Proceedings of the 3rd International Conference on GA, Morgan Kaufmann. Los Altos, CA, 104-109

[Fogel 1998] Fogel, D.B. *Evlutionary Computation: The Fossil Record*, (1998). IEEE Press, Piscataway, NJ.

[Forrest 1993] Forrest, S. and Mitchell, H. (1993). *What makes a problem hard for a genetic algorithm? Some anomalous results and their explanation*. Machine Learning, 13(2-3):285-319.

[Garrido y Comas, 1987] Garrido y Comas (1987).*Teoría general y derecho español en los seguros privados.* En Tratado general de seguros: teoría y práctica de los seguros privados .I:II Editado por el Consejo General de los Colegios de Agentes y Corredores de Seguros de España.

[Goldberg 1989] Goldberg, D. E. (1989): *Genetic algorithms in search, optimization and machine learning*, Reading, MA: Addison-Wesley.

[Goldberg 1989 *b*] Goldberg. D. E. (1989). *Genetic algorithms and Walsh functions: Part II, deception and its analysis.* Complex Systems , 3:153–171.

[Goldberg et al 1992] Goldberg, D.E., Deb K., Clark J.H. (1992) *GA, noise and the sizing of populations*. Complex Systems, **6**, 333-362.

[González et al. 2002] González, J., Rojas, I., Pomares, H., Salmerón, M. and Merelo, J. (2002). *Web newspaper layout optimization using Simulated Annealing,* IEEE Trans. Systems, Man and Cybernetics, 32(5):686-691.

[Grefenstette 1986] Grefenstette J. J. (1986) *Optimization of control parameters for GA*. IEEE-SMC**, SMC-16**, 122-128.

[Hair et al 1999] Hair, Anderson, Tatham, Black, (1999). *Análisis Multivariante.* Ed.: Prentice Hall.

[Hancock 1996] Hancock P.J.B. (1995) *Selection methods for evolutionary algorithms*. En L. Chambers (ed.). Practical Handbook of GA: New frontiers, Vol II, CRC Press, Boca Raton, FL, 67-92.

[Haykin 1999] Haykin,S. *Neural networks comprehensive foundation.* (1999). (International Edition) Prentice Hall.

[Holland 2000] Holland J.H.(2000) *Building blocks, cohort genetic algorithms, and hyperplane-defined functions.* Evolutionary Computation, **8**, 373-391.

[Hossack 1999] Hossack I. B., Pollard J.H., Zehnwirth B. *Introducción a la estadística con aplicaciones a los seguros generales*. Editado por Fundación MAPFRE Estudios en 2001.

[Johnson 1972] Jonson, P. D. Hey G.B. (1972) *Statistical review of a motor insurance portfolio.* ASTIN Bulletin **6:3**, 222-232.

[Kirpatrick (1983)] Kirpatrick, S., Gerlatt, C. D. and Vecchi, M. P. (1983). *Optimization by simulated annealing*, Science, 220:671-680.

[Koza 1992] Koza J. *Genetic programming*. Cambridge, MA: MIT Press; 1992

[Koza 1994] Koza J. *Genetic programming for economic modelling*. Statistics and Computing 1994;4(2):187–97.
[Lanteli 1962] Lanteli, G. (1962), *Novelties i Swedish Automobile insurance rating.* ASTIN Bulletin **2:1**, 96-101.

[Lemaire 1995] Lemaire, J (1995) *Bonus-malus system in automobile insurance. Kluwer-Nijhof publishing* . Boston M.A..

[Liepins 1990] Liepins, G. E. and M. D. Vose, M. D. (1990). *Representational issues in genetic optimization*. J. Experimental and Theoretical Artificial Intelligence, 2(1):101-115.

[Martín et al1999] Martín Martín, Q., Cabero Morán, M.T., Ardanuy Albajar, R. (1999). *Paquetes Estadísticos Spss 8.0*. Ed. Hespérides.

[Martín y Sanz 2001] Martín del Brío B., Sanz Molina A. *Redes Neuronales y Sistemas Borrosos* (2001) Ra-Ma. Madrid.

[Matías et al. 2006] Matías J. M., Rivas T., Martín J.E., Taboada J. *Analysis of workplace accidents using machine learning techniques*, Proceedings of the 2006 conference on computational and mathematical methods on science and engineering. Regino Criado (editor) vol II pp. 470-482.

[Michalewicz 1996] Michalewicz Z. (1996) *Genetic algorithms + data structures = evolution programs*. Berlin: Springer.

[Mitchell 1997] Mitchell T.M. (1997) *Machine Learning*. McGraw Hill

[Mitchell 1998] Mitchell M. (1998) *An introduction to genetic algorithms*, MIT Press USA (7ª edición 2001)

[Nieto y Vegas 1993] Nieto, U., Vegas J. (1993) *Matemática actuarial*. Editorial MAPFRE S.A.

[O'leary 1998] O'Leary, D. E. (1998): *Using neural networks to predict corporate failure*. International Jounal of Intelligent Systems in Accounting Finance and Management, 7:187-197.

[Pérez-Cruz et al. 2005] Pérez-Cruz, F. Bousoño-Calzón, C., Artés-Rodríguez A. (2005). *Convergence of the IRWLS Procedure to the Support Vector Machine Solution*. Neural Computation 17, 7-18 (2005) MIT Press.

[Patuwo et al. 1993] Patuwo, E., Wu, M. Y., and Hung, M. S. (1993): *Two group classification using neural networks*. Decision Sciences, 23:899-916.

[Popper 1959] Popper, K. *The logic of Scientific Discovery*, 1932. Springer (1ª edición inglesa por Hutchinson, 1959)

[Quinlan 1983] Quinlan, J.R. *Learning Efficient Classification Procedures And Their Application To Chess End Games*.

[Radcliffe y Surry 1995] Radcliffe N.J., Surry P.D. (1995). *Format and the variance of fitness* En D. Whitley D. Y M.Vose (Eds.) Foundations of GA 3, Morgan Kaufmann, San Mateo, CA 51-72.

[Reeves (Ed) 1993] Reeves C. (Ed) (1993) *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, Oxford, UK (reeditado por McGraw-Hill, London UK 1995).

[Reeves 1994] Reeves, C. and Wright, A. (1994). *An experimental design perspective on genetic algorithms*. In Foundations of genetic algorithms, pages 7–22.

[Reeves 1995] Reeves C.R. (1995). *A genetic algorithm for flowshop sequencing.* Computers& Operations Research, **22**, 5-13.

[Reeves 2002] Reeves, Colin R., Rowe, Jonathan E.(2002) *Genetic Algorithms-Principles and perspectives: A guide to GA Theory*. Secaucus, NJ, USA: Kluwer Academic Publishers.

[Refenes 1995] Refenes, A-P (ed.). (1995) *Neural Networks in the Capital Markets*. New York: John Wiley & Sons

[Salcedo-Sanz et al 2002] Salcedo-Sanz S.,De Prado-Cumplido M., Pérez-Cruz F., Bousoño-Calzón C. *Feature Selection via Genetic Optimization.* ICANN 2002: 547-552

[Salcedo-Sanz et al. et al. 2005] Salcedo-Sanz, S., Fernández-Villacañas J. L., Segovia-Vargas, M. J. and Bousoño-Calzón, C. (2005). *Genetic programming for the prediction of insolvency in non-life insurance companies*, Computers & OR 32: 749-765

[Salcedo-Sanz 2006] Salcedo-Sanz, S., Bousoño-Calzón, C. (2006). *On the Application of Linear Transformations for Genetic Algorithms Optimization* International Journal of Knowledge-base and Ingelligent Engineering Systems, IOS press.

[Salcedo-Sanz et al 2003] Salcedo-Sanz, S., Bousoño-Calzón, C. Figueiras-Vidal, A. (2003). *A mixed neural-genetic algorithm for the broadcast scheduling problem* IEEE Trans. Wireless Commun., **2**(2):277–283.

[Salomon, R. 1996] Salomon, R. (1996). *Reevaluating genetic algorithm performance undercoordinate rotation of benchmark functions.* BioSystems, 39(3):263–278, Sept.1996.

[Samuel 1959] Samuel M. *Some studies in machine learning using the game of checkers.* IBM Journal of Research and Development, 29: 210-229, 1959

[Sanchis et al. (2003)] Sanchis, A., Gil, J. A., and Heras, A. (2003): *El análisis discriminante en la previsión de la insolvencia en la empresa de seguros no-vida*, Revista Española de Financiación y Contabilidad 115.

[Sareni 1998] Sareni, B. and Krähenbühl, L. (1998). *Fitness sharing and niching methods revisited.* IEEE Trans. Evol. Comput., 2(3):97–106.

[Scholkopf et al. 1999] Scholkopf B, Burges C.J.C., Smola A. (1999) *Advances in Kernels Methods. Support vector Learning*. Cambridge, MA: MIT Press.

[Scholkopf y Smola 2002] Scholkopf B, Smola A. (2002) *Learning with kernels*. Cambridge, MA: MIT Press.

[Segovia 2002] Segovia MJ, Gil JA, Heras A, Vilar JL, Sanchis A. *Using Rough Sets to predict insolvency of Spanish non-life insurance companies.* Proceedings of Sixth International Congress on Insurance: Mathematics and Economics, Lisboa, 2002.

[Segovia-Vargas et al 2003] Segovia-Vargas MJ, Salcedo-Sanz S, Bousoño-Calzón C. *Prediction of insolvency in non-life insurance companies using Support Vector Machines and genetic algorithms.* In: Proceedings of X SIGEF Congress in Emergent Solutions for the Information and Knowledge Economy, León, Spain, 2003.

[Serrano (1996)] Serrano-Cinca, C. (1996). *Self organizing neural networks for financial diagnosis, Decision Support Systems*, 17:227-238.

[Shapiro 2001] Shapiro, *A. Soft Computing Applications in Actuarial Science.* ARCH. 2001. http://www.soa.org:80/library/arch/2000-09/arch01v113.pdf

[Shapiro 2002] Shapiro, A. *The merging of Neural Networks, fuzzi logic and genetic algorithms* Insurance: Mathematics and Economics 31 (2002) 115-131

[Shawe-Taylor 2004] Shawe-Taylor, J., Cristianini, N. *Kernel Methods for pattern analysis*. (2004) Cambridge University Press .

[Schölkopf y Smola 2002] Schölkopf B., Smola A. 2002 *Learning with kernels : support vector machines, regularization, optimization and beyond* MIT Press Cambridge, Massachusetts.

[Tam 1991] Tam, K. Y.: *Neural network models and the prediction of bankruptcy*, (1991). Omega, 19(5):429-445.

[Tam et al. 1992] Tam, K. Y., and Kiang, M. Y. (1992): *Managerial applications of neural networks: the case of bank failure predictions*, Management Science, 38(7):926-947.

[Theodossiou 1996] Theodossiou P, Kahya E, Saidi R, Philippatos G. *Financial distress and corporate acquisitions: further empirical evidence*. Journal of Business Finance 1996;23(5–6):699–719.
[Tolmos 1998] Tolmos, P. *Las Redes Neuronales y los mercados de capitales (I)*. Studia Carande 98.

[Tolmos et al. 1999] Tolmos, P., Aparicio, A., Ibarra, A., Garrido, G. *Redes Neuronales y su aplicación predictiva en la Bolsa de Valores Española*. Actas de las VII Jornadas de ASEPUMA. Valencia 1999.

[Tolmos et al. 2000]. Tolmos, P. Molero J.J., Sevillano F. J. *The use of a Neural Network in the prediction of the IBEX-35 stock index.* Proceedings of the 5th International Meeting on Artificial Intelligence And Emerging Technologies in Accounting, Finance and Taxation. Huelva, 2000.

[Tolmos 2002] *Introducción a los Algoritmos Genéticos y sus aplicaciones*. Working Papers 2003/02. Servicio de publicaciones de la Universidad Rey Juan Carlos

[Tolmos 2002] *La red neuronal de Oja. Aspectos matemáticos.* Studia Carande 2002. Servicio de publicaciones de la Universidad Rey Juan Carlos.

[Turing 1950] Turing A. (1950), *Computing Machines and Intelligence.* Mind, 29:233-260.

[Vafaie y De Jong (1992)] Vafaie, H.,De Jong, K. A. *Genetic Algorithms as a Tool for Features Selection in Machine Learning.* (1992) Proc. of the 4th Intl. Confon Tools with Artificial Systems, IEEE computer society press, Arlintong, VA, 200-204.

[Vapnik y Chervonenkis 1974] Vapnik V., Chervonenkis A. (1974). *Theory of Pattern Recognition* (*in Russian*) Nauka, Moscú.

[Vapnik 1995] Vapnik, V., Cortes, C. *Support-Vector Networks.* 1995. Machine Learning, 20, 273-297.

[Vapnik 1998] Vapnik V. *Statistical learning theory.* New York: Wiley; 1998

[Vapnik 1999] Vapnik V. *An overview of Statistical Learning Theory.* IEEE Transactions on Neural Networks, vol. 10, **5**, Sept 1999.

[Vapnik 1999] Vapnik V. *The Nature of Statistical learning theory*. (1999). Springer.

[Vaughn et al. 1997] Vaughn, M. L., E. Ong and S.J. Cavill, *Interpretation and Knowledge Discovery from a Multilayer Perceptron Network that Performs Whole Life Assurance Risk Assessment*, (1997). Neural Computing and Applications, 6:201-213

[Vegas , A. 1992a] Vegas A. *Fundamentos técnicos del Sistema Bonus-Malus*. VIII Jornadas comunitarias del Seguro del Automóvil., 1992.

[Vegas , A. 1992b] Vegas A. *Fundamentos técnicos del Sistema Bonus-Malus*. Previsión y Seguro, Enero 1993. 22: 141-167.

[Vegas , A. 1993] Vegas A. *Aplicación del sistema Bonus-Malus a la tarificación de los jóvenes conductores en el seguro del automóvil*. Previsión y Seguro, Septiembre 1993. 29: 55-78.

[Weston y Watkins 1999] Weston J. Watkins C. *Multi-class SVM*. 1999. En M. Verleisen (ed) Proceedings of the ESANN. Bruselas. D Facto.

[Weston et al. (2000)] Weston, H., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2000): *Feature Selection for SVMs*, Advances in NIPS 12, MIT Press, 526-532.

[Wilson et al. (1994)] Wilson, R. L., and Sharda, R. (1994): *Bankruptcy prediction using neural networks*, Decision Support Systems, 11:545-557.

[Zopounidis et al. (1998)] Zopounidis. C., and Dimitras, A. (1998): *Multicriteria decision aid methods for the prediction of business failure*, Kluwer.

[Zopounidis (1999)] Zopounidis, C. (1999): *Multicriteria decision aid in financial management*, European Journal of Operational Research, 119(2):404-415.

COLECCIÓN "CUADERNOS DE LA FUNDACIÓN"
Instituto de Ciencias del Seguro

Para cualquier información o para adquirir nuestras publicaciones puede encontrarnos en:

Instituto de Ciencias del Seguro
Publicaciones
Monte del Pilar, s/n – 28023 El Plantío, Madrid – (España)
Telf.: + 34 915 818 768
Fax: +34 913 076 641
publicaciones.ics@mapfre.com
www.fundacionmapfre.com/cienciasdelseguro

122. Factores de riesgo y cálculo de primas mediante técnicas de aprendizaje. 2008

121. La solicitud de seguro en la Ley 50/1980, de 8 de octubre, de Contrato de Seguro. 2008

120. Propuestas para un sistema de cobertura de enfermedades catastróficas en Argentina. 2008

119. Análisis del riesgo en seguros en el marco de Solvencia II: Técnicas estadísticas avanzadas: Monte Carlo y Bootstrapping. 2008

118. Los planes de pensiones y los planes de previsión asegurados: su inclusión en el caudal hereditario. 2007

117. Evolução de resultados tecnicos e financieros no mercado segurador iberoamericano. 2007

116. Análisis de la Ley 26/2006 de Mediación de Seguros y Reaseguros Privados. 2007

115. Sistemas de cofinanciación de la dependencia: seguro privado frente a hipoteca inversa. 2007

114. El sector asegurador ante el cambio climático: riesgos y oportunidades. 2007

113. Responsabilidade social empresarial no mercado de seguros brasileiro influências culturais e implicações relacionais. 2007

112. Contabilidad y análisis de cuentas anuales de entidades aseguradoras. 2007

111. Fundamentos actuariales de primas y reservas de fianzas. 2007

110. El *Fair Value* de las provisiones técnicas de los seguros de Vida. 2007

109. El Seguro como instrumento de gestión de los M.E.R. (Materiales Específicados de Riesgo). 2006

108. Mercados de absorción de riesgos. 2006

107. La exteriorización de los compromisos por pensiones en la negociación colectiva. 2006

106. La utilización de datos médicos y genéticos en el ámbito de las compañías aseguradoras. 2006

105. Los seguros contra incendios forestales y su aplicación en Galicia. 2006

104. Fiscalidad del seguro en América Latina. 2006

103. Las NIIF y su relación con el Plan Contable de Entidades Aseguradoras. 2006

102. Naturaleza jurídica del Seguro de Asistencia en Viaje. 2006

101. El Seguro de Automóviles en Iberoamérica. 2006

100. El nuevo perfil productivo y los seguros agropecuarios en Argentina. 2006

99. Modelos alternativos de transferencia y financiación de riesgos “ART”: situación actual y perspectivas futuras. 2005

98 Disciplina de mercado en la industria de seguros en América Latina. 2005

97 Aplicación de métodos de inteligencia artificial para el análisis de la solvencia en entidades aseguradoras. 2005

96. El Sistema ABC-ABM: su aplicación en las entidades aseguradoras. 2005

95. Papel del docente universitario: ¿enseñar o ayudar a aprender?. 2005

94. La renovación del Pacto de Toledo y la reforma del sistema de pensiones: ¿es suficiente el pacto político?. 2005

92. Medición de la esperanza de vida residual según niveles de dependencia en España y costes de cuidados de larga duración. 2005

91. Problemática de la reforma de la Ley de Contrato de Seguro. 2005

90. Centros de atención telefónica del sector asegurador. 2005

89. Mercados aseguradores en el área mediterránea y cooperación para su desarrollo. 2005

88. Análisis multivariante aplicado a la selección de factores de riesgo en la tarificación. 2004

87. Dependencia en el modelo individual, aplicación al riesgo de crédito. 2004

86 El margen de solvencia de las entidades aseguradoras en Iberoamérica. 2004

85. La matriz valor-fidelidad en el análisis de los asegurados en el ramo del automóvil. 2004

84. Estudio de la estructura de una cartera de pólizas y de la eficacia de un Bonus-Malus. 2004

83. La teoría del valor extremo: fundamentos y aplicación al seguro, ramo de responsabilidad civil autos. 2004

81. El Seguro de Dependencia: una visión general. 2004

80. Los planes y fondos de pensiones en el contexto europeo: la necesidad de una armonización. 2004

79. La actividad de las compañías aseguradoras de vida en el marco de la gestión integral de activos y pasivos. 2003

78. Nuevas perspectivas de la educación universitaria a distancia. 2003

77. El coste de los riesgos en la empresa española: 2001.

76. La incorporación de los sistemas privados de pensiones en las pequeñas y medianas empresas. 2003

75. Incidencia de la nueva Ley de Enjuiciamiento Civil en los procesos de responsabilidad civil derivada del uso de vehículos a motor. 2002

74. Estructuras de propiedad, organización y canales de distribución de las empresas aseguradoras en el mercado español. 2002

73. Financiación del capital-riesgo mediante el seguro. 2002

72. Análisis del proceso de exteriorización de los compromisos por pensiones. 2002

71. Gestión de activos y pasivos en la cartera de un fondo de pensiones. 2002

70. El cuadro de mando integral para las entidades aseguradoras. 2002

69. Provisiones para prestaciones a la luz del Reglamento de Ordenación y Supervisión de los Seguros Privados; métodos estadísticos de cálculo. 2002

68. Los seguros de crédito y de caución en Iberoamérica. 2001

67. Gestión directiva en la internacionalización de la empresa. 2001

65. Ética empresarial y globalización. 2001

64. Fundamentos técnicos de la regulación del margen de solvencia. 2001

63. Análisis de la repercusión fiscal del seguro de vida y los planes de pensiones. Instrumentos de previsión social individual y empresarial. 2001

62. Seguridad Social: temas generales y régimen de clases pasivas del Estado. 2001

61. Sistemas Bonus-Malus generalizados con inclusión de los costes de los siniestros. 2001

60. Análisis técnico y económico del conjunto de las empresas aseguradoras de la Unión Europea. 2001

59. Estudio sobre el euro y el seguro. 2000

58. Problemática contable de las operaciones de reaseguro. 2000

56. Análisis económico y estadístico de los factores determinantes de la demanda de los seguros privados en España. 2000

54. El corredor de reaseguros y su legislación específica en América y Europa. 2000

53. Habilidades directivas: estudio de sesgo de género en instrumentos de evaluación. 2000

52. La estructura financiera de las entidades de seguros, S.A. 2000

50. Mixturas de distribuciones: aplicación a las variables más relevantes que modelan la siniestralidad en la empresa aseguradora. 1999

49. Solvencia y estabilidad financiera en la empresa de seguros: metodología y evaluación empírica mediante análisis multivariante. 1999

48. Matemática Actuarial no vida con MapleV. 1999

47. El fraude en el Seguro de Automóvil: cómo detectarlo. 1999

46. Evolución y predicción de las tablas de mortalidad dinámicas para la población española. 1999

45. Los Impuestos en una economía global. 1999

42. La Responsabilidad Civil por contaminación del entorno y su aseguramiento. 1998

41. De Maastricht a Amsterdam: un paso más en la integración europea. 1998

Nº Especial Informe sobre el Mercado Español de Seguros 1997
Fundación MAPFRE Estudios

39. Perspectiva histórica de los documentos estadístico-contables del órgano de control: aspectos jurídicos, formalización y explotación. 1997

38. Legislación y estadísticas del mercado de seguros en la comunidad iberoamericana. 1997

37. La responsabilidad civil por accidente de circulación. Puntual comparación de los derechos francés y español. 1997

36. Cláusulas limitativas de los derechos de los asegurados y cláusulas delimitadoras del riesgo cubierto: las cláusulas de limitación temporal de la cobertura en el Seguro de Responsabilidad Civil. 1997

35. El control de riesgos en fraudes informáticos. 1997

34. El coste de los riesgos en la empresa española: 1995

33. La función del derecho en la economía. 1997

Nº Especial Informe sobre el Mercado Español de Seguros 1996
Fundación MAPFRE Estudios

32. Decisiones racionales en reaseguro. 1996

31. Tipos estratégicos, orientación al mercado y resultados económicos: análisis empírico del sector asegurador español. 1996

30. El tiempo del directivo. 1996

29. Ruina y Seguro de Responsabilidad Civil Decenal. 1996

Nº Especial Informe sobre el Mercado Español de Seguros 1995
Fundación MAPFRE Estudios

28. La naturaleza jurídica del Seguro de Responsabilidad Civil. 1995

27 La calidad total como factor para elevar la cuota de mercado en empresas de seguros. 1995

26. El coste de los riesgos en la empresa española: 1993

25. El reaseguro financiero. 1995

24. El seguro: expresión de solidaridad desde la perspectiva del derecho. 1995

23. Análisis de la demanda del seguro sanitario privado. 1993

Nº Especial Informe sobre el Mercado Español de Seguros 1994
Fundación MAPFRE Estudios

22. Rentabilidad y productividad de entidades aseguradoras. 1994

21. La nueva regulación de las provisiones técnicas en la Directiva de Cuentas de la C.E.E. 1994

20. El Reaseguro en los procesos de integración económica. 1994

19. Una teoría de la educación. 1994

18. El Seguro de Crédito a la exportación en los países de la OCDE (evaluación de los resultados de los aseguradores públicos). 1994

Nº Especial Informe sobre el mercado español de seguros 1993
FUNDACION MAPFRE ESTUDIOS

16. La legislación española de seguros y su adaptación a la normativa comunitaria. 1993

15. El coste de los riesgos en la empresa española: 1991

14. El Reaseguro de exceso de pérdidas 1993

12. Los seguros de salud y la sanidad privada. 1993

10. Desarrollo directivo: una inversión estratégica. 1992

9. Técnicas de trabajo intelectual. 1992

8. La implantación de un sistema de *controlling* estratégico en la empresa. 1992

7. Los seguros de responsabilidad civil y su obligatoriedad de aseguramiento. 1992

6. Elementos de dirección estratégica de la empresa. 1992

5. La distribución comercial del seguro: sus estrategias y riesgos. 1991

4. Los seguros en una Europa cambiante: 1990-95. 1991

2. Resultados de la encuesta sobre la formación superior para los profesionales de entidades aseguradoras (A.P.S.). 1991

1. Filosofía empresarial: selección de artículos y ejemplos prácticos. 1991