UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS

TESIS DOCTORAL

SELECCIÓN DE ATRIBUTOS EN APRENDIZAJE AUTOMÁTICO BASADA EN TEORÍA DE LA INFORMACIÓN

JOSÉ JAVIER LORENZO NAVARRO

Las Palmas de Gran Canaria, Mayo de 2001

59/2000-01

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
UNIDAD DE TERCER CICLO Y POSTGRADO

Reunido el día de la fecha, el Tribunal nombrado por el Excmo. Sr. Rector Magfco. de esta Universidad, el/a aspirante expuso esta TESIS DOCTORAL.
Terminada la lectura y contestadas por el/a Doctorando/a las objeciones formuladas por los señores miembros del Tribunal, éste calificó dicho trabajo con la nota de Sobresaliente cum laude

Las Palmas de Gran Canaria, a 10 de julio de 2001.

El/a Presidente/a: Dr. D. Antonio Falcón Martel,

El/a Secretario/a: Dr. D. Juan A. Méndez Rodríguez,

El/a Vocal: Dr. D. Antonio Bahamonde Rionda,

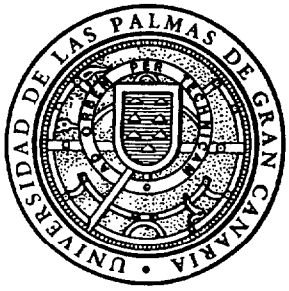
El/a Vocal: Dr.D. José Andrés Moreno Pérez,

El/a Vocal: Dr.D. Casiano Rodríguez León,

El Doctorando: D. José Javier Lorenzo Navarro,

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Departamento de Informática y Sistemas



TESIS DOCTORAL

SELECCIÓN DE ATRIBUTOS EN APRENDIZAJE AUTOMÁTICO BASADA EN TEORÍA DE LA INFORMACIÓN

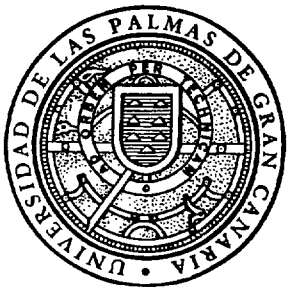
José Javier Lorenzo Navarro

Las Palmas de Gran Canaria

Mayo de 2001

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

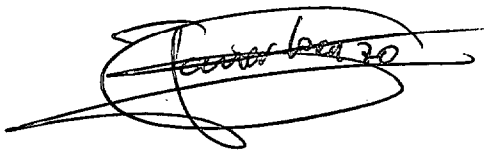
Departamento de Informática y Sistemas



TESIS TITULADA **SELECCIÓN DE ATRIBUTOS EN APRENDIZAJE AUTOMÁTICO BASADA EN TEORÍA DE LA INFORMACIÓN**, QUE PRESENTA D. JOSÉ JAVIER LORENZO NAVARRO, REALIZADA BAJO LA DIRECCIÓN DEL DOCTOR D. FRANCISCO MARIO HERNÁNDEZ TEJERA

Las Palmas de Gran Canaria, Mayo de 2001

El doctorando



José Javier Lorenzo Navarro

El director

Francisco Mario Hernández Tejera

AGRADECIMIENTOS

Quiero agradecer al director de esta tesis, Francisco Mario Hernández Tejera, el apoyo y los consejos recibidos en la realización de la misma y al constante estímulo que he recibido durante el tiempo que ha durado su elaboración.

También quiero agradecer a todos los miembros del grupo de investigación "Grupo de Inteligencia Artificial y Sistemas", a los cuales en algún momento he recurrido para solicitar su consejo o ayuda. Especialmente a Juan Méndez Rodríguez y a Jorge Cabrera Gámez, por las sugerencias aportadas en el curso de productivas discusiones.

Esta memoria está dedicada a mi familia y, muy especialmente, a Carmen María, que conoce bien el esfuerzo empleado en la preparación de esta tesis y le da sentido.

Gracias a todos.

A Carmen María

y

a mis padres

Índice General

Índice de Figuras

Índice de Tablas

Índice de Algoritmos

Lista de Símbolos

X_i : Atributo *i-ésimo*.

$Dom(X_i)$: Dominio del atributo X_i si es nominal.

n_i : Número de valores que puede tomar el atributo X_i. Cardinalidad del conjunto $Dom(X_i)$.

x_i : Valor que toma el atributo *i-ésimo* en un caso concreto, $x_i \in Dom(X_i)$.

$\mathbf{X}$: Vector atributos.

n : Número de atributos del vector $\mathbf{X}$.

$\mathcal{X}$: Espacio de atributos. $\mathcal{X} = Dom(X_1) \times Dom(X_2) \times \cdots \times Dom(X_n)$

Y : Clase o concepto.

$\mathcal{Y}$: Conjunto de valores que puede tomar la clase o concepto.

y : Valor de la clase o concepto para una muestra determinada, $y \in \mathcal{Y}$

$\mathcal{D}$: Conjunto de datos de aprendizaje.

$\langle \mathbf{X}^{(j)}, Y^{(j)} \rangle$: Múestra etiquetada *j-ésima* del conjunto de aprendizaje $\mathcal{D}$.

N : Número de muestras en el conjunto de datos $\mathcal{D}$.

$\mathbf{A} \backslash \mathbf{B}$: Conjunto $\mathbf{A}$ menos los elementos contenidos en el subconjunto $\mathbf{B}$.

Resumen

El trabajo realizado en esta tesis se enmarca en el campo del Aprendizaje Automático. Más concretamente se centra en el Aprendizaje Supervisado donde se persigue la inducción de descripciones generales a partir de casos particulares de un problema, cuya solución se incluye como elemento en el aprendizaje. Un tipo de problemas en los que se aplica el Aprendizaje Supervisado es el aprendizaje de conceptos, donde el objetivo consiste en obtener un mecanismo (clasificador) que permita generalizar e indicar si un nuevo caso, no utilizado en el proceso de inducción, pertenece o no al concepto o clase aprendido. Los casos utilizados en el proceso de inducción en muchos problemas vienen expresados como muestras o tuplas que incluyen medidas o los atributos obtenidos del caso en estudio además de la solución o clase a la que pertenece.

En este escenario, la calidad del conocimiento inducido depende fuertemente de la calidad de las medidas utilizadas en lo que a representatividad del concepto se refiere. Ello es debido a que, por un lado, no se puede utilizar un número infinito de medidas, y por otro algunas técnicas desarrolladas en Aprendizaje Supervisado degradan su rendimiento cuando la "calidad" de estas medidas no es suficientemente buena.

Por tanto un problema existente es la selección de los atributos más relevantes para la inducción del conocimiento; problema que se estudia en esta tesis y para el cual se propone una solución. Así mismo se emplea esta solución en un problema concreto de Aprendizaje en Visión Artificial. Para abordar el problema, se utilizarán conceptos que proceden de la Teoría de la Información, ya que el marco formal que se desarrolla estará basado en el estudio de la similitud existente entre un canal de información en el sentido que se estudia en Teoría de la Información y un clasificador.

El contenido de esta tesis se encuentra dividido en seis capítulos que van desde una visión de los diferentes paradigmas de Aprendizaje Automático hasta la aplicación del método propuesto en una arquitectura para la inducción de clasificadores en un contexto de sistemas de visión.

El primer capítulo está orientado a dar una visión panorámica del problema del

aprendizaje, comenzando por un recorrido por distintos paradigmas de Aprendizaje Automático y haciendo mayor hincapié en el Aprendizaje Inductivo y dentro de éste en el Aprendizaje Supervisado. A continuación se exponen diferentes técnicas de inducción de conceptos y mecanismos de clasificación según el principio en el que se basan: métodos simbólicos, redes neuronales o algoritmos genéticos. Como algunos clasificadores degradan su rendimiento cuando tienen como entradas atributos no relevantes para el problema, es necesario disponer de técnicas que permitan medir el rendimiento de un clasificador. Una medida de rendimiento utilizada con frecuencia es la tasa de error del clasificador, por tanto se exponen diferentes técnicas de estimación de la tasa de error, así como diferentes comparativas realizadas en algunos trabajos, donde se comprueba el mejor marco de utilización de cada una de las técnicas expuestas.

El segundo capítulo está dedicado a presentar brevemente algunos de los conceptos de la Teoría de la Información que serán utilizados luego en el desarrollo del marco formal y la implementación práctica que se propone, la medida GD.

El problema de la selección de atributos se basa en el concepto de relevancia, entendiendo como atributos relevantes en un problema dado a los que mejor definen el concepto o clase y por tanto los que deben ser seleccionados. El problema que existe es que la definición de relevancia no es única, sino que como se puede ver en el tercer capítulo, existen varias definiciones dependiendo de para qué considera cada autores que son relevantes los atributos. Esta disparidad de criterios a la hora de definir la relevancia, tiene su reflejo en los diferentes trabajos existentes sobre selección de atributos, ya que éste no es un problema exclusivo del Aprendizaje Automático, sino que también se ha estudiado en campos como el Reconocimiento de Formas o el Análisis Multivariante. Por tanto se hace imprescindible una revisión de la bibliografía existente. Para ello se ha seguido la clasificación propuesta por Doak que establece diferentes categorías de métodos de selección de atributos, en función de la medida utilizada para comprobar la calidad de los atributos seleccionados y la estrategia que se sigue para buscar el conjunto de atributos relevantes según la medida utilizada.

En el capítulo cuarto es donde se presenta el marco formal de esta tesis. Este marco se basa en el estudio de la similitud conceptual entre canal de información y clasificador. Esta similitud da lugar a un conjunto de definiciones sobre relevancia de atributos basadas en conceptos de la Teoría de la Información. A partir de este conjunto de definiciones se propone una medida de calidad relativa de conjuntos de atributos y un método para la selección de atributos, la medida GD. Esta medida recoge la posible dependencia existente entre los atributos para medir de forma ponderada la información que aporta cada uno de los atributos de un conjunto al proceso de inducción. Debido

a que en algunos problemas pueden existir valores desconocidos para algunos atributos, se estudia la influencia de este hecho en la medida GD y se propone un esquema de sustitución de estos valores perdidos de forma que permita la utilización de las muestras incompletas minimizando el sesgo en el resultado final.

Para evaluar la calidad del método propuesto se diseña un marco experimental y se desarrollan una serie de experimentos. Estos experimentos, cuyos resultados obtenidos se presentan en el capítulo quinto, se han orientado hacia tres objetivos distintos:

a) El primer objetivo que se pretende con los experimentos es comprobar la calidad de la medida en si, es decir, como se ve influenciada por el número de valores de los atributos y la dependencia de estos atributos con la clase. Para ello se utilizó el experimento diseñado por Kononenko.

b) Otro objetivo que se persigue, es comprobar la respuesta de la medida GD en problemas donde se conoce el resultado a priori, es decir, problemas sintéticos con la dependencia entre los atributos y la clase conocida.

c) Por último, debido a que los problemas sintéticos anteriores no se corresponden en muchos casos con problemas reales, sobre todo en dominios con datos provenientes de sensores como es el caso de los problemas de Percepción Artificial, se realizó un tercer bloque experimental comparando la calidad de los atributos seleccionados por la medida GD frente a los seleccionados con otros métodos. La calidad se mide como la tasa de error obtenida con diferentes clasificadores, validando estadísticamente los resultados para evitar el posible sesgo que puede introducir la toma aleatoria de muestras.

En el último capítulo se presenta una arquitectura para el aprendizaje de clasificadores en un problema de Visión, donde la medida GD se incluye como un módulo más del mismo y que se demuestra su utilidad, al permitir obtener clasificadores con un número reducido de atributos que dan lugar a buenos resultados sobre el conjunto de imágenes disponibles para realizar el aprendizaje.

Capítulo 1

Introducción

Este capítulo se presentan los objetivos que se pretenden alcanzar en esta tesis. También se introduce el Aprendizaje Automático como un área de la Inteligencia Artificial así como una clasificación de los diferentes tipos de Aprendizaje Automático. Luego se centrará en el Aprendizaje Inductivo Supervisado donde se desarrolla la propuesta realizada en esta tesis, presentándose diferentes aproximaciones dentro de esta categoría y diferentes métodos y algoritmos para el aprendizaje de clasificadores. Por último se incluye una breve descripción de diferentes técnicas para la estimación del error de los clasificadores inducidos.

1.1 Objetivos y Definiciones

Esta tesis tiene como objetivo el estudio y propuesta de un método de selección de atributos basado en Teoría de la Información en el marco del Aprendizaje Supervisado. Los conceptos involucrados así como una breve descripción de diferentes paradigmas de Aprendizaje Automático se explican en este capítulo.

La utilización de la Teoría de la Información como base teórica para el método que se pretende desarrollar, supone ubicarnos en un marco formal donde se obtenga la relación entre conceptos de esta teoría y los manejados en el entorno del Aprendizaje Supervisado. De esta forma, se podrá abordar el problema de la selección de atributos haciendo uso de estos conceptos que han sido ampliamente aplicados en diferentes campos, incluyendo otras aproximaciones a la selección de atributos. Este marco formal deberá incluir entre otras, la definición del conjunto de atributos relevantes que será el que mejor define el concepto o la clase en estudio.

Un aspecto importante en la realización de esta tesis es la obtención de una propuesta de implementación práctica de los conceptos teóricos planteados en el marco formal a desarrollar, que permita la selección de los atributos más relevantes según la definición de relevancia propuesta, sin que ello suponga la estimación funciones de probabilidad con alto coste computacional.

Una vez que se disponga del método práctico para la selección de atributos, este debe ser probado en diferentes escenarios. Para ello será necesario diseñar un marco experimental que incluya una comprobación del método propuesto en problemas de aprendizaje donde el resultado es conocido a priori, es decir, donde se conocen qué atributos definen la clase. Además de éstos, se incluirán otros problemas obtenidos de bases de datos públicas y que son utilizados por otros autores de forma que los resultados puedan ser comparados con otros trabajos en el mismo campo. Independientemente, en el marco experimental se comprobará la calidad del método que se propone mediante la comparación con otros métodos existentes en la bibliografía, validando los resultados estadísticamente para evitar sesgos indeseados en las conclusiones.

Por último, se estudiará la utilización del método propuesto en un problema de visión, proponiendo una arquitectura que permita su integración en relación con otros módulos que deben conformar esta arquitectura, como son la selección de muestras, pixels en este caso, o la inducción de clasificadores que resuelvan el problema de la clasificación de los pixels entre diferentes clases de interés definidas previamente por el usuario.

Una vez expuestos los diferentes objetivos que se proponen alcanzar, se deben establecer algunas definiciones de conceptos que se manejarán a lo largo de este documento. Partimos de las definiciones, ampliamente utilizadas en multitud de trabajos recogidas por Kohavi en su tesis doctoral (Kohavi, 1995*b*) y en otro trabajo del mismo autor (Kohavi y Provost, 1998).

Definición 1.1. *Un* ***atributo*** *o también denominado* ***característica*** *es la descripción de alguna medida de una muestra. Los atributos tienen un dominio definido por el tipo de atributo, determinando dicho dominio los valores que puede tomar el atributo.*

Definición 1.2. *Una* ***muestra, ejemplo*** *o* ***caso*** *es una lista de valores de atributos que se corresponde con la definición de las entidades tratadas en el problema de Aprendizaje Automático en estudio, por ejemplo pueden corresponder a un paciente, a un solicitante de una tarjeta de crédito o a una planta. En algunos casos la representación de las muestras son más complejas incluyendo relaciones entre muestras o entre partes que componen la muestra.*

Los dos grandes grupos en los que se pueden clasificar los atributos son: nominales y continuos. Los ***atributos nominales*** pueden tomar un número finito de valores discretos, entre los que no se puede establecer una relación de orden, por ejemplo el color o la marca de un coche. En algunos casos se suele denominar a los atributos nominales que toman solo dos valores: cierto o falso, como **atributos booleanos** o ***lógicos***. Si entre los valores discretos que puede tomar un atributo se puede establecer una relación de orden, entonces se denomina como ***atributos ordinales*** , por ejemplo la temperatura se puede considerar como baja, templada o alta. Los ***atributos continuos***, normalmente un subconjunto de los números reales, son aquellos en los que se puede establecer una relación de orden. En algunos casos para un determinado atributo no se conoce su valor, bien porque se ha perdido o porque no se ha podido obtener, denominándose a estos valores como ***valores perdidos***. Las causas para que aparezca un valor perdido pueden ser varias: no se puede medir, se ha producido un funcionamiento erróneo del aparato de medida, el atributo no se puede aplicar o el valor de atributo no se pudo conocer.

En cada muestra existe un atributo (nominal o continuo) denominado ***etiqueta*** que indica la clase a la que pertenece la muestra. Una muestra no etiquetada es una muestra que no posee etiqueta, es decir la lista de los valores de los atributos. En el marco del aprendizaje supervisado, es necesario disponer de un conjunto de muestras etiquetadas, donde en general no se supone ninguna ordenación entre las mismas, a partir de cual se llevará a cabo el proceso de aprendizaje. A este conjunto se le denomina ***conjunto de aprendizaje***.

En la Tabla 1.1 se muestra un ejemplo de un conjunto de aprendizaje que consta de 14 muestras. Cada una compuesta por cuatro atributos: Cielo, Temperatura, Humedad y Viento, con 3, 3, 2 y 2 valores respectivamente. Los atributos en este caso son nominales (Cielo y Viento) y ordinales (Temperatura y Humedad) y la etiqueta representa la decisión de jugar al tenis o no en función de las condiciones meteorológicas dadas por los atributos.

Definida una terminología básica, se pasa a describir los símbolos utilizados a lo largo de este documento. El atributo i-ésimo se representará como X_i y su dominio como $Dom(X_i)$, por tanto cada muestra no etiquetada se corresponderá con un punto del espacio $\mathcal{X} = Dom(X_1) \times Dom(X_2) \times \cdots \times Dom(X_n)$ donde n es el número de atributos. Una muestra no etiquetada se especificará como $\mathbf{X}$ y a su valor denominado ***vector de atributos*** (vector que contiene los valores de los atributos para dicha muestra) como $\mathbf{x}$, donde x_i corresponde al valor de cada uno de los atributos del vector de atributos.

La salida asociada a cada muestra se indicará como Y y a su valor y, donde $y \in \mathcal{Y}$

Atributos				Etiqueta
Cielo {soleado,cubierto,lluvia}	Temperatura {frío,templada,calor}	Humedad {alta,normal}	Viento {si,no}	
soleado	calor	alta	no	no
soleado	calor	alta	si	no
cubierto	calor	alta	si	si
lluvia	templada	alta	no	si
lluvia	frío	normal	no	si
lluvia	frío	normal	si	no
cubierto	frío	normal	si	si
soleado	templada	alta	no	no
soleado	frío	normal	no	si
lluvia	templada	normal	no	si
soleado	templada	normal	si	si
cubierto	templada	alta	si	si
cubierto	calor	normal	no	si
lluvia	templada	alta	si	no

Tabla 1.1: Ejemplo de conjunto de aprendizaje

siendo $\mathcal{Y}$ el conjunto de posibles valores que puede tomar la etiqueta de cada muestra. En función de la naturaleza de $\mathcal{Y}$ el problema será de regresión si este conjunto es el de los números reales, mientras que si es un conjunto de muestras finito estamos ante un problema de clasificación. Por tanto $\mathcal{X} \times \mathcal{Y}$ es el espacio al que pertenecen las muestras etiquetadas y $\mathcal{D}$ es el conjunto de aprendizaje cuyo elemento j-ésimo es un par del tipo $\langle \mathbf{X}^{(j)}, Y^{(j)} \rangle$, denominado como N al número de muestras contenidas en el conjunto. Para que el proceso de aprendizaje tenga utilidad, las muestras en el conjunto de aprendizaje deben tener una distribución similar a la que poseen los miembros de la población, y esto se consigue mediante un muestreo aleatorio de los elementos de la población que forman parte del conjunto de aprendizaje.

En Aprendizaje Supervisado se parte del conjunto $\mathcal{D}$, y se pretende obtener un mecanismo que realice la clasificación correcta de las muestras contenidas en dicho conjunto, así como de nuevas muestras no utilizadas en el proceso de aprendizaje y para las cuales no se conoce su clase. Este mecanismo es el clasificador, que se define como

Definición 1.3. *Un **clasificador** es una función que asigna a una muestra no etiquetada una etiqueta. Todo los clasificadores poseen una estructura de datos interna para realizar la asignación de una etiqueta a un ejemplo.*

Por ejemplo, un clasificador basado en un árbol de decisión tiene internamente almacenado un árbol de decisión que se utiliza para la asignación de la etiqueta a un ejemplo mediante el recorrido desde la raíz del árbol hasta un nodo hoja. Un clasificador

basado en el vecino más cercano se basa en asignar la etiqueta de la muestra almacenada más cercano a la muestra a etiquetar. En el Perceptrón la estructura subyacente es el de una máquina lineal con tantas unidades de salida como etiquetas posibles puede tomar una muestra. A la muestra no etiquetada se asigna la etiqueta de la unidad de salida que posee mayor valor.

Definición 1.4. *Un **algoritmo de inducción** tiene como objetivo la construcción de un clasificador a partir de un conjunto de aprendizaje. Por ejemplo ID3 o C4.5 son algoritmos de inducción para la construcción de árboles de decisión. Mientras que las técnicas basadas en descenso según el gradiente como la propagación hacia atrás del error (backpropagation) se utilizan para la construcción de clasificadores del tipo Perceptrón.*

Como ya se comentó anteriormente un clasificador $\mathcal{C}$ asocia a una muestra no etiquetada $\mathbf{x}$ un grado de pertenencia a una clase $y \in \mathcal{Y}$, mientras que un algoritmo de inducción $\mathcal{I}$ efectúa la correspondencia de un conjunto de aprendizaje $\mathcal{D}$ en un clasificador $\mathcal{C}$. Para indicar la etiqueta que asigna el clasificador $\mathcal{C}$ que ha sido obtenido a partir del conjunto de aprendizaje $\mathcal{D}$ a la muestra $\mathbf{x}$ se utilizará la expresión $\mathcal{I}(\mathcal{D}, \mathbf{x})$.

1.2 ¿Qué es Aprendizaje Automático?

A la anterior pregunta no existe una única respuesta en la bibliografía, aunque la más referenciada es la enunciada por Simon (Simon, 1983) que define Aprendizaje Automático como:

> (Simon, 1983) "Aprendizaje denota cambios en el sistema que son adaptativos en el sentido de que permiten al sistema realizar la misma tarea, o una tomada de la misma población, la próxima vez de una forma más eficiente y efectiva."

Otras definiciones debidas a otros autores posteriores, y que por tanto se encuentran en la misma línea que la propuesta por Simon son:

> (Carbonell, 1989) "Aprendizaje se puede definir operacionalmente como la habilidad para realizar nuevas tareas que no podía realizar anteriormente o realizar anteriores tareas mejor (más rápidas, más exactas, etc.) como resultado de los cambios producidos por el proceso de aprendizaje."

(Forsyth, 1989) "El aprendizaje es un fenómeno que se muestra cuando un sistema mejora su rendimiento en una determinada tarea sin necesidad de ser reprogramado."

(Weiss y Kulikowski, 1991) "Un sistema que aprende es un programa de computador que toma decisiones en base a la experiencia acumulada contenida en casos resueltos satisfactoriamente. A diferencia de los sistemas expertos, que resuelven los problemas utilizando un modelo por computador del razonamiento del experto humano, un sistema de aprendizaje puro puede utilizar muchas técnicas diferentes para explotar el potencial computacional del computador, sin importar su relación con el proceso cognitivo humano."

(Anzai, 1992) "Cuando un sistema genera automáticamente una nueva estructura de datos o programa a partir de una existente y de esta forma irreversible cambia con algún propósito por un determinado tiempo, es lo que llamamos aprendizaje automático"

(Langley, 1996) "Aprendizaje es la mejora en el rendimiento en ciertos entornos por medio de la adquisición de conocimiento como resultado de la experiencia en dicho entorno"

(Mitchell, 1997) "Un programa de ordenador se dice que aprende de la experiencia E con respecto a una clase de tareas T y a la medida de rendimiento P, si su rendimiento en las tareas que pertenecen a T medido según P, se incrementa con la experiencia E"

Todas las definiciones anteriores tienen en común una serie de elementos siendo el más destacable que el aprendizaje tiene como fin el mejorar el rendimiento del sistema de forma autónoma, es decir sin la intervención de un operador externo. También se observa que el Aprendizaje Automático no está concebido de forma generalista, sino que se centra en mejorar tareas concretas del sistema. En este aspecto se puede observar un enfoque de ingeniería en la forma que se aborda el problema, ya que no tiene como fin la explicación del proceso de aprendizaje genérico. Por último, indicar que las técnicas de Aprendizaje Automático no tienen porque seguir las pautas del razonamiento humano como indica Weiss (Weiss y Kulikowski, 1991), entre otras cosas porque aún no se conoce con total certeza cual es el mecanismo que siguen los humanos para realizar el razonamiento siendo un tema de estudio actual en áreas como la Filosofía o la Psicología.

Aunque el despegue del Aprendizaje Automático se produce en los años ochenta, de ahí que las primeras definiciones mostradas anteriormente daten de esos años, la búsqueda de sistemas con capacidad de aprender se remonta a los primeros días de los computadores. Así, a finales de la década de los años cincuenta, coetáneas con la aparición del transistor, aparece una clase de máquinas diseñadas por Rosenblatt (Rosenblatt, 1958), denominadas Perceptrón que parecían ofrecer a muchos investigadores un modelo natural y potente de máquina de aprendizaje. Posteriormente se consideró que las expectativas que se crearon en lo referente a las prestaciones del Perceptrón eran excesivamente optimistas, aunque los conceptos matemáticos que surgieron de su desarrollo se pueden considerar los cimientos del paradigma de aprendizaje de las Redes Neuronales Artificiales (Rumelhart et al., 1986*a*; Rumelhart et al., 1986*b*).

La adquisición del conocimiento por parte de los sistemas Aprendizaje Automático se puede realizar de diferentes formas, igual que ocurre en los humanos que no tienen una única forma de aprender, aunque todos los paradigma de aprendizaje se pueden encuadrar en las definiciones antes enunciadas, ya que todos tienen como objetivo común el incremento del rendimiento del sistema que adquiere el conocimiento. La taxonomía de los diferentes métodos de aprendizaje se puede realizar en base a diferentes parámetros, por lo que existen diferentes clasificaciones de éstos no siendo excluyentes unas de otras. En (Forsyth, 1989) se realiza la clasificación en función de ocho parámetros que pueden tener dos posibilidades lo que da lugar a 256 posibles tipos de aprendizaje, estos parámetros son:

- Propósito general vs específico.
- Aprendizaje incremental vs una sola pasada.
- Jerárquico vs no jerárquico.
- Determinista vs no determinista.
- Evaluación lógica vs cuantitativa.
- Características unitarias vs predicados estructurales.
- Reglas comprensibles vs caja negra.
- Lenguaje fijo vs extensible.

Carbonell (Carbonell et al., 1983) propone una clasificación de los métodos de aprendizaje en función de tres parámetros: estrategia de aprendizaje utilizada, representación del conocimiento y el dominio de aplicación. A continuación se muestran los distintos tipos de aprendizaje clasificados de acuerdo a la estrategia de clasificación propuesta por Carbonell, y que coincide con la realizada por Michalski (Michalski, 1983) y

Kodratoff (Kodratoff, 1988). Según ella los métodos de aprendizaje se pueden agrupar en::

- Aprendizaje por Repetición o por Implantación Directa.
- Aprendizaje por Instrucción.
- Aprendizaje por Deducción.
- Aprendizaje por Analogía.
- Aprendizaje Supervisado.
- Aprendizaje no Supervisado.

Una representación jerárquica de estas categorías se puede ver en la Figura 1.1. Otra clasificación de los diferentes paradigmas de aprendizaje, con aspectos más de técnicos que la anterior, se basa en diferentes tipos de algoritmos utilizados para llevar a cabo la tarea del aprendizaje, ya que según Langley (Langley, 1996, pág. 5) "si el aprendizaje automático es una ciencia, ésta es claramente una ciencia de algoritmos". Según esta taxonomía se pueden distinguir los siguientes paradigma de Aprendizaje Automático:

- Redes neuronales.
- Aprendizaje basado en casos.
- Algoritmos genéticos.
- Inducción de reglas.
- Aprendizaje analítico.

Otro aspecto importante que caracteriza a los sistemas de Aprendizaje Automático es cómo el conocimiento adquirido mejora el rendimiento de ese sistema en un determinado entorno. La respuesta a la anterior cuestión equivale a determinar qué conocimiento puede adquirir dicho sistema. El conocimiento que puede adquirir un sistema de Aprendizaje Automático se puede englobar en los siguientes grandes grupos:

Aprendizaje de Conceptos El objeto del aprendizaje es obtener una función que relacione cada caso de un determinado dominio con un concepto mediante la asignación de un grado de pertenencia a dicho concepto, siendo normalmente la tasa de error de la función obtenida por el proceso de aprendizaje el parámetro que se intenta mejorar.

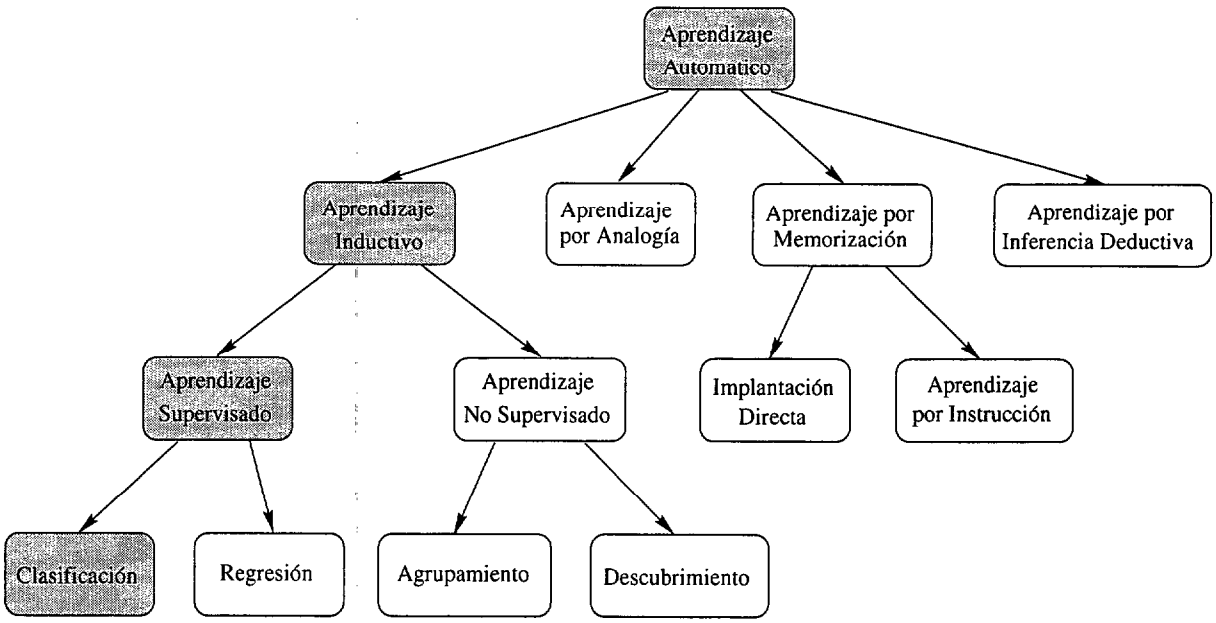


Figura 1.1: Organización jerárquica de las técnicas de Aprendizaje Automático.

Clasificación Es una generalización del aprendizaje de conceptos, donde se tiene un número arbitrario de conceptos y por tanto la función obtenida en el proceso de aprendizaje realizará la asignación de un elemento del dominio a uno o más conceptos. Si el problema es mutuamente exclusivo, es decir que un caso no puede pertenecer a más de un concepto, el problema de clasificación se puede convertir en un número de problemas de aprendizaje de conceptos igual al número de conceptos en los que se clasifican los casos.

Resolución de problemas El conocimiento a adquirir en este caso consiste en una secuencia de pasos o reglas de control que permitan resolver un determinado problema. En la adquisición de este tipo de conocimiento, el incremento del rendimiento del sistema se suele medir como el incremento de eficiencia más que la tasa de error a la hora de resolver el problema.

Descripción de Conceptos Un tipo de conocimiento bastante importante que se pretende obtener mediante las técnicas de Aprendizaje Automático es el relacionado con la descripción de un concepto, bien por el descubrimiento de patrones o por la formulación de una teoría que caracteriza un conjunto de entidades. El incremento de rendimiento que se busca viene dado por la adecuación de los datos observados a las descripciones obtenidas.

A continuación se presenta un modelo para el Aprendizaje Automático (Fig. 1.2). Este modelo recoge los elementos principales que definen el Aprendizaje Automático

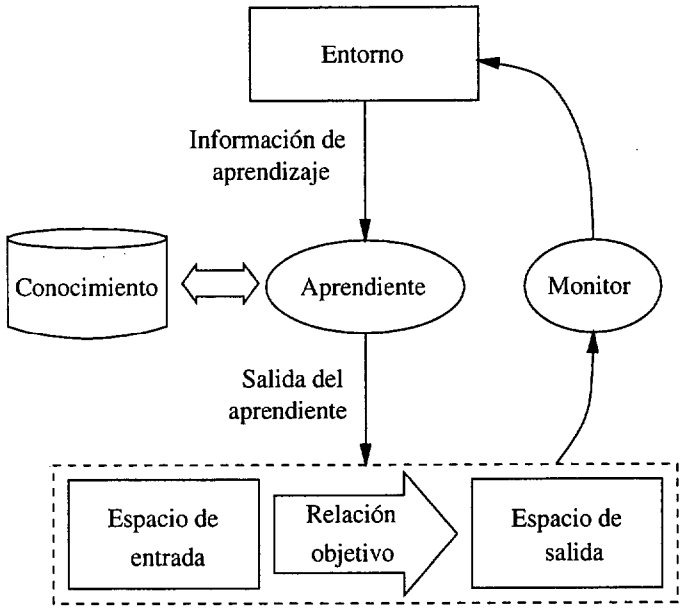


Figura 1.2: Modelo de Aprendizaje

donde el aprendiente tiene como objetivo la obtención de una relación entre un espacio de entrada y uno de salida a partir de la interacción con un entorno en el que se desarrolla el proceso de aprendizaje. La finalidad de la obtención de esta relación es el incremento de rendimiento en un conjunto de tareas que vienen representadas en el modelo por la relación a obtener.

En el modelo propuesto los espacios de entrada y salida se corresponden con representaciones del entorno de aprendizaje, que pueden tener igual o diferente contenido semántico. El aprendiente además de una interacción con el entorno, interacciona con una base de conocimiento donde se almacenará el conocimiento obtenido mientras que en otros casos proveerá conocimiento previo para llevar a cabo el proceso de aprendizaje.

En las siguientes secciones se da una breve descripción de los distintos tipos de aprendizaje propuestos por Carbonell (pág. 8).

1.3 Aprendizaje por Memorización y Aprendizaje por Instrucción

En esta categoría se recogen aquellos métodos en los que no se produce ningún proceso de aprendizaje propiamente dicho, ya que todo el proceso de adquisición de conocimiento es realizado por un operador externo al sistema. No obstante desde el punto de vista

del modelo presentado antes, este tipo de aprendizaje representaría la transformación del espacio de entrada al espacio de salida, por lo que independientemente de no existir un proceso de aprendizaje propiamente dicho, si existe una mejora del rendimiento en el sistema y desde ese punto de vista sí existe aprendizaje en el sistema.

Dentro de esta categoría se tiene por un lado el *Aprendizaje por Repetición* o *Implantación Directa*, en el cual el conocimiento se introduce directamente en forma de programa por un operador o programador, no existiendo ningún tipo de estructuración del conocimiento. Otros métodos se encuadrarían en el *Aprendizaje por Instrucción* en el que el conocimiento que se implanta en el sistema sí posee una estructura pero la tarea de la adquisición y estructuración está a cargo de un operador y el sistema lo adapta al esquema de representación interno utilizado. En Artificial Intelligence Encyclopaedia (Shapiro, 1992) cuando se describe este último tipo de aprendizaje se comenta al sistema NANOKLAUS (Hass y Hendrix, 1983) como un ejemplo del aprendizaje por instrucción en el que se construye una base de conocimiento jerárquica por medio de la conversación con un usuario.

1.4 Aprendizaje por Deducción

En el aprendizaje por deducción se parte de un conjunto de conceptos generales y otro de casos y mediante la presentación de los casos al sistema se comprueban si son particularidades del concepto general y se procede a un proceso de mejora del conocimiento existente simplificando las hipótesis existentes o incrementando la eficiencia, mediante un análisis que busca qué parte del conocimiento es más útil. La transformación del conocimiento existente se realiza con relaciones de equivalencia ya que se debe mantener los hechos que en él se expresan sin modificarlos. Estas relaciones de equivalencia introducidas permiten la adaptación del conocimiento a un nuevo entorno. Kodrakoff (Kodratoff, 1988) divide el aprendizaje por deducción en dos categorías: *Aprendizaje Guiado por Especificaciones* y *Aprendizaje Basado en Ejemplos.* En el aprendizaje guiado por especificaciones el conocimiento proviene de una especificación y el proceso deductivo transforma las relaciones existentes en un algoritmo que actualiza esas relaciones. En aprendizaje basado en ejemplos, los ejemplos son usados para mostrar qué nuevas reglas son interesantes para derivar del conocimiento existente.

Un tipo de aprendizaje por deducción basado en ejemplos es el *Aprendizaje basado en Explicación* (*Explanation-based Learning EBL*). Los sistemas basados en este tipo de aprendizaje parten de la definición de un concepto abstracto y de un conocimiento

del dominio y por deducción derivan una definición operacional del concepto. La derivación del concepto funcional se obtiene a partir de una explicación de porqué un ejemplo es una particularización del concepto abstracto. Esta explicación identifica las características relevantes del ejemplo que constituyen las condiciones suficientes para describir el concepto. En el EBL, al igual que en el aprendizaje inductivo, el propósito es dar una descripción del concepto que permita reconocer nuevas particularizaciones de dicho concepto. La diferencia con el aprendizaje inductivo está en que en los sistemas EBL se hace uso de una teoría del dominio que dirige el proceso de análisis. Algunos sistemas que se basan es este paradigma son STRIPS (Fikes et al., 1972), GENESIS (Mooney y DeJong, 1985), el sistema propuesto por Minton (Minton, 1984) capaz de aprender en el ámbito del ajedrez, la aproximación EBG (Mitchell et al., 1986) y PRODIGY (Minton et al., 1989).

STRIPS y su técnica de creación de macro-operadores es quizás el precursor que más influenció el resto de los trabajos en EBL. Este sistema intenta obtener una solución a un problema de planificación mediante la aplicación de una secuencia de acciones, dando como resultado un plan. Una vez resuelto el problema, el plan obtenido se puede convertir en un conjunto de macro-operadores para resolver nuevos problemas similares. Este hecho aparece en el aprendizaje por analogía que se explica en la siguiente sección, aunque en EBL se considera un plan como la explicación de un determinado objetivo que es alcanzable. La formación de un macro-operador da las condiciones suficientes que deben cumplir los miembros de la clase, en este caso el objetivo a alcanzar.

En el sistema propuesto por Minton (Minton, 1984) para jugar al ajedrez, el programa puede aprender mediante el análisis de los casos en los que su oponente pudo forzarlo a efectuar una jugada desventajosa. Así el programa puede aprender combinaciones donde un jugador obliga al otro a realizar una jugada que conlleva una perdida. Después de caer en una de estas jugadas, el programa analiza porqué la jugada pudo realizarse y aprende una regla que permite evitar la jugada o bien utilizarla contra un jugador contrario. Para ello el sistema reconstruye la secuencia de acciones que han dado lugar a la jugada y mediante análisis identifica el conjunto de condiciones que se deben dar, y en futuras jugadas cuando se dan estas condiciones puede concluir que es una jugada que hará incurrir en una perdida.

En EBG se describe un modelo de aprendizaje basado en explicaciones que articula muchos de los aspectos que son comunes a varios sistemas EBL. Por un lado establece un significado preciso para el término "explicación" mediante la identificación de las explicaciones que obtiene el sistema con las pruebas que se realizan. También se caracteriza este tipo de sistemas mediante la especificación de cuáles deben ser las

entradas y salidas al sistema. Según el modelo EBG las entradas de un sistema de aprendizaje basado en explicación son las siguientes:

- *Definición del concepto objetivo*, que consiste en la definición del concepto que el sistema debe aprender.
- *Ejemplo de aprendizaje*, es un ejemplo del concepto objetivo.
- *Teoría del dominio*, consiste en un conjunto de reglas y hechos necesarios para construir las explicaciones de cómo un ejemplo de aprendizaje es un ejemplo del concepto objetivo.
- *Criterio de operación*, que es un predicado sobre las definiciones del concepto que especifica la forma en la cual se debe expresar el concepto aprendido

En cuanto a las salidas del sistema EBL solo es una condición suficiente para reconocer el concepto objetivo. De acuerdo al modelo, los sistemas EBL demuestran que el ejemplo de aprendizaje es una instancia del concepto objetivo y obtienen la *precondición más débil* de la demostración. Esta precondición es la condición suficiente (más débil) bajo la cual la demostración es válida.

PRODIGY es un *resolvedor* de problemas que adquiere nuevo conocimiento analizando sus experiencias e interactuando con un experto. Este sistema está basado en el Análisis de Medias-Fines (*Mean-Ends Analysis*) e incluye varios módulos aparte del aprendizaje basado en explicaciones. El módulo EBL de PRODIGY utiliza como conceptos objetivos los cuatro tipo de búsqueda de control que implementa el sistema y son: SUCEEDS, FAILS, SOLE-ALTERNATIVE y GOAL-INTERFACE y dependen del fallo o acierto de las reglas de control seleccionadas. En cuanto a la teoría del dominio en PRODIGY coexisten dos conjuntos de reglas. Un conjunto de reglas que describen la resolución del problema y es el que se obtiene como resultado del proceso de aprendizaje. El otro conjunto de reglas que dispone PRODIGY es el que representa el dominio de la tarea en la que se emplea el sistema y que describe los operadores disponibles para resolver el problema. Para construir las explicaciones se parte de un ejemplo y se utiliza el método que definen los autores como *Especialización Basada en Explicación.* Con este método un concepto se especializa recuperando un axioma que describa dicho concepto y recursivamente especializando este axioma hasta encontrar primitivas. De esta forma la secuencia de especialización demuestra que el ejemplo es una particularización válida del concepto objetivo. Por último el criterio de operación que va a permitir utilizar el concepto aprendido, una regla de control en este caso, es extendido en este sistema incluyendo la característica adicional de utilidad. Así cuando se aprende una reglas de

control, ésta debe mejorar el rendimiento del sistema ya que de lo contrario se descarta. Para calcular la utilidad de una regla se tienen en cuenta, por un lado el coste de comparación de la regla y el promedio de ahorro cuando se aplica la regla, y por otro la proporción del número de veces que se aplica la regla frente al número de veces que se comprueba su posible utilización. Así una regla aprendida que se utiliza pocas veces termina por desecharse.

1.5 Aprendizaje por Analogía

En esta aproximación al aprendizaje se intenta transferir el conocimiento sobre una tarea bien conocida a una menos conocida. Por tanto este tipo aprendizaje es una potente herramienta para explotar la experiencia que se tiene de casos anteriores de planificación y resolución de problemas, que guardan una cierta relación con la tarea actual. Dos elementos importantes a tener en cuenta en esta aproximación son: ¿cómo recuperar los casos anteriores que son similares a uno dado? y ¿cómo modificar los casos recuperados para poderlos aplicar en la resolución de problema actual?. En el primer caso se pueden buscar analogías entre diferentes partes del problema como pueden ser las condiciones de partida, el objetivo a cumplir o los pasos que se dan para resolver dicho problema.

Winston (Winston, 1980) presenta un sistema de razonamiento por analogía basado en redes semánticas que representan relaciones entre los elementos que componen frases escritas en inglés. La entrada al sistema la constituyen las frases, sujetas a una serie de restricciones para facilitar el proceso de análisis. A partir del texto escrito el sistema construye redes semánticas cuyos nodos corresponden con los sujetos de las frases y los nodos con las acciones representadas por verbos o por características, denominadas *plots*. La analogía entre situaciones se obtiene mediante un proceso de comparación de grafos, obteniendo todos los posibles emparejamientos entre los dos grafos. A cada uno de los emparejamientos obtenidos se le asocia una puntuación que depende de la similitud entre los nodos y las características. Para reducir el espacio de búsqueda en el emparejamiento de los grafos cuando tienen cierta dimensión, se utiliza restricciones aportadas por el usuario. Del grado de coincidencia de los grafos se pueden obtener reglas específicas para un dominio o reglas generales a partir de la comparación de casos correspondientes a dominios diferentes. En este último caso se presenta como ejemplo la obtención de la regla de linealidad que cumple la ley de Ohm para resistencias y la relación de presión en tubos.

Carbonell (Carbonell, 1983) utiliza este tipo de aprendizaje para la resolución de problemas mediante el método de Análisis de Medios-Fines basándose en la hipótesis de que cuando se le presenta a una persona un nuevo problema se utiliza para su resolución técnicas o estrategias utilizadas en problemas ya resueltos anteriormente y que guardan bastante similitud con el problema actual.

El conocimiento de situaciones anteriores se puede utilizar de diferentes formas. La más sencilla es hacer uso de la misma solución aportada para resolver la situación pasada en la situación actual, donde la solución normalmente consiste en una secuencia de operadores y estados intermedios. Otra posibilidad es no utilizar toda la solución completa sino generar macro-operadores a partir de secuencias y subsecuencias de operadores (Fikes et al., 1972), y utilizar estos macro-operadores para resolver las nuevas situaciones que se le plantean al sistema. Este esquema de reutilización del conocimiento tiene un inconveniente que es el crecimiento combinatorio del número de soluciones que se pueden tomar, llegándose al caso en que aplicar el MEA al nuevo problema a resolver puede ser más rápido que hacer una búsqueda de las soluciones anteriores, ya que se van creando soluciones más complejas que se aplican a problemas cada vez más concretos.

Como ya se ha comentado, el primer paso en el aprendizaje por analogía consiste en buscar problemas anteriormente resueltos cuya solución se pueda transformar para resolver el nuevo problema. Esta fase se puede considerar como un proceso de "recordar" situaciones pasadas. Una vez se ha encontrado una solución candidata, ésta se debe transformar para que pueda se utilizada en la resolución de uno nuevo. Carbonell define la *métrica de diferencia* D_T, que es una combinación de las diferencia de cuatro elementos entre el problema resuelto y el nuevo problema. Estos elementos son: los estados iniciales, el estado final, el conjunto de restricciones de ambos problemas y por último de lo que define como *aplicabilidad* que corresponde con la proporción de precondiciones de operadores en el problema resuelto y en el nuevo. Una vez definida esta medida la transformación de la solución de un problema a uno nuevo se puede encontrar mediante la aplicación del MEA en un espacio transformado, que Carbonell denomina *espacio del problema transformado por analogía (T-espacio)*, donde los estados son soluciones potenciales en el espacio inicial de los problemas, el estado inicial es la solución obtenida para el problema resuelto, el estado final es una especificación para la resolución del nuevo problema y los operadores en el T-espacio, denominados T-operadores para evitar confusión con los operadores en el espacio de los problemas, transforman una solución en otra posible solución. Entre estos operadores se pueden encontrar la inserción de un operador en la secuencia de la solución, la eliminación, la división, la unión, ...

El proceso de transformación por analogía, antes comentado, va a proporcionar un método de explotar la experiencia pasada de una forma flexible, ya que se requiere que el nuevo problema solo sea estructuralmente similar más que completamente idéntico a uno o más de los problemas anteriormente resueltos. Por tanto el aprendizaje en este sistema puede consistir en el almacenamiento de casos resueltos. Otra posibilidad es la de obtener planes para la resolución de problemas generalizados, es decir planes que se pueden aplicar a un conjunto de problemas iguales que solo varían en algún parámetro. En este caso el aprendizaje puede ocurrir tanto en la fase de recuperación de casos y en la de transformación por analogía.

Un método de aprendizaje similar al aprendizaje por analogía es el *Razonamiento basado en Casos* (*Case-based Reasoning*) (Aamodt y Plaza, 1994). La idea que soporta el CBR es la misma que el aprendizaje por analogía; la resolución de un problema consiste de alguna forma en la reutilización de información de problemas resueltos con anterioridad en la situación actual. Los problemas se almacenan como casos que son una descripción esquemática de un problema. Esta descripción puede ser una n-tupla o descriptores que caracterizan cada elemento del dominio del problema. Un caso solo contiene aquellos descriptores que fueron relevantes para resolver dicho problema y por tanto dos casos en la memoria de casos pueden contener conjuntos disjuntos de descriptores. Es tarea del sistema CBR realizar el proceso de analogía trabajando solo con similaridades parciales entre los casos. Muchos de los sistemas basados en CBR utilizan funciones para medir la similaridad entre casos. A este respecto se debe indicar que en la memoria del sistema existe una colección de casos resueltos cuya única diferencia con el caso actual es que este último no contiene el descriptor que contiene la solución del problema, y por tanto este descriptor no interviene en la función de similaridad.

El ciclo de trabajo de un sistema CBR se puede ver en la Figura 1.3. El ciclo comienza con la introducción de un caso por parte del usuario para ser resuelto. Después este caso se utiliza para buscar los más similares de los almacenados en la memoria de casos. Como resultado de esta búsqueda se pueden tener uno o varios casos que encajan bien con el problema a resolver. Éstos se tienen que utilizar para construir una nueva solución, que se puede realizar de diversas formas. La más sencilla es utilizar la solución de uno de los casos encontrados para resolver el problema actual. Otra posibilidad más elaborada es la adaptación de la solución encontrada al nuevo problema mediante la modificación de algunos de los parámetros utilizados siguiendo algún criterio prefijado. Si se han recuperado varios casos de la memoria de casos se debe decidir si cada uno contribuye con una parte diferente a la solución o bien si se adaptan todos simultáneamente. Y por último, está la utilización de los pasos dados en la solución del

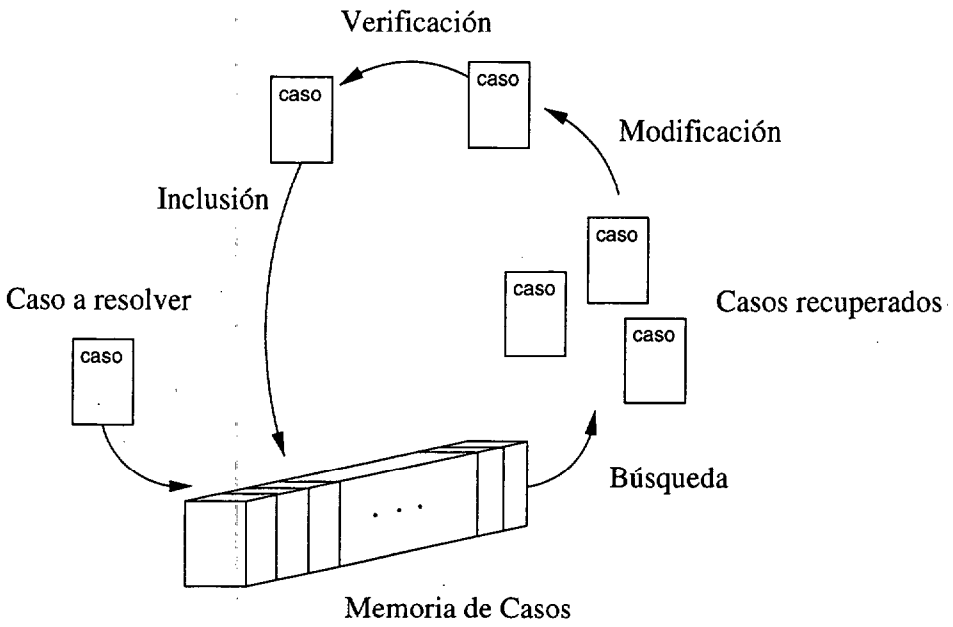


Figura 1.3: Ciclo de trabajo de un sistema CBR

caso recuperado en la resolución del nuevo problema. Para poder realizar esto, los casos deben contener además de los descriptores las decisiones tomadas en cada momento. Esta aproximación en los sistemas CBR coincide con el aprendizaje por analogía antes comentado, y por este motivo algunas veces se referencia el aprendizaje por analogía como razonamiento basado en casos y viceversa. Una vez obtenida una solución, ésta debe ser validada por un experto en el dominio o mediante la ejecución de un modelo alternativo del dominio.

El aprendizaje en un sistema CBR puede consistir en principio en la adición de los casos resueltos a la memoria de casos. Sin embargo si esta inclusión se hace sin ningún tipo de control, el rendimiento del sistema decae principalmente por dos motivos. El primero es el aumento del tiempo de respuesta ya que la búsqueda de casos similares cada vez es más lenta y por otra parte pueden existir valoraciones erróneas por parte de la medida de similaridad por la inclusión de casos irrelevantes. Para resolver estos inconvenientes primero se puede realizar un proceso de análisis para identificar casos irrelevantes, y luego se pueden sustituir varios casos de la memoria por un único caso prototipo.

El elemento clave en el razonamiento basado en casos es la medida utilizada para obtener la similaridad entre dos situaciones o casos. La similaridad entre dos casos se puede dividir en una similaridad local y una global. La similaridad local hace referencia

a la comparación de la misma característica en dos casos diferentes y la similaridad global se encarga de combinar las similaridades locales en una única medida. Así si C_i es un caso descrito por el conjunto de descriptores $\{a_1, a_2, \ldots, a_n\}$ la similaridad global con otro caso C_j viene dada por la expresión:

$$SF(C_i, C_j) = F(Sf_1(a_1, b_1), Sf_2(a_2, b_2), \ldots, Sf_n(a_n, b_n))$$

donde Sf_i es la función similaridad local entre los descriptor de los dos casos. Esta función depende del tipo de descriptor, así para descriptores booleanos o perteneciente a un conjunto finito de valores puede ser 1 si son iguales y 0 si no lo son. Cuando los descriptores son numéricos existen diferentes posibilidades, diferencia absoluta de valores, diferencia ponderada, etc. En cuanto a la función de similaridad global $SF(C_i, C_j)$ se suelen utilizar dos aproximaciones: una combinación lineal de las similaridades locales, o la utilización de redes de discriminación donde en cada nodo se toma una decisión en base al valor de una similaridad local de un descriptor.

1.6 Aprendizaje Inductivo

El objetivo de la inferencia inductiva o aprendizaje inductivo es la formulación de hipótesis generales plausibles que expliquen unos hechos conocidos y que además sea capaz de predecir nuevos hechos. La diferencia con la deducción se encuentra que en lugar de partir de axiomas generales, se parte de hechos concretos para llegar a los axiomas generales.

Este tipo de aprendizaje ha sido estudiado por filósofos y psicólogos desde mucho antes de la aparición del Aprendizaje Automático. En el campo de la Filosofía el proceso de la inducción ha despertado interés ya que no solo se encuentra presente en el proceso de aprendizaje diario, sino que ha sustentado el método científico durante siglos. Un ejemplo de proceso de inducción que aparece bastante referenciado es el del amanecer:

> Ayer el sol salió por el Este y se puso por el Oeste.
> Toda mi vida el sol ha salido por el Este y se ha puesto por el Oeste.
> Nadie recuerda que esto no haya sido así.
> En toda la historia siempre el sol ha salido por el Este y se ha puesto por el Oeste.
>
> Por tanto, el sol mañana hará lo mismo.

Este tipo de inferencias es lógicamente inválida y por tanto los filósofos han

intentado encontrar una justificación que las valide, de forma que las conclusiones que se obtengan se puedan verificar de la misma forma que se hace con la deducción lógica. El filósofo David Hume en el siglo XVIII observó este proceso y afirmó que era debido a que el número de aserciones que se pueden hacer de forma inductiva es infinito, mientras que el número de pruebas que lo pueden confirmar es pequeño. Otros filósofos que han estudiado el problema de la inducción han sido Francis Bacon, John Stuart Mill, Bertrand Russell o Ludwing Wittgenstein, desde el punto de vista de la búsqueda de un conjunto de reglas que permitan formalizar el proceso de la inducción, de la misma forma que existen las reglas lógicas para el proceso deductivo. A pesar de ello, algunos elementos identificados por estos filósofos han influenciado el desarrollo de sistemas de Aprendizaje Automático.

Por ejemplo Francis Bacon ya en el siglo XVII puso de manifiesto la importancia de las evidencias negativas y la tendencia de los humanos a pasarlas por alto. Otro aspecto que introdujo es que el proceso inductivo tenga sentido, la hipótesis obtenida debe ir más allá de los hechos a partir de los cuales se obtiene. Así siempre que ocurre la confirmación de la hipótesis por nuevas observaciones, la confianza en la teoría inducida se ve reforzada. Este es un elemento clave que se encuentra en el Aprendizaje Automático cuando se intenta medir el rendimiento del método en estudio como se verá en la sección de estimación del error.

John Stuart Mill dos siglos más tarde estableció cuatro métodos experimentales para inducir reglas generales a partir de casos particulares.

Método de las coincidencias Si dos o más ejemplos de un fenómeno tienen un único factor en común, el factor en el cual todas las muestras coinciden es la causa o efecto del fenómeno.

Método de las diferencias Si una instancia positiva de un fenómeno y una negativa tienen todas las circunstancias en común excepto una, esta circunstancia en la cual los dos ejemplos difieren es el efecto o causa, o al menos una parte indispensable, del fenómeno en estudio.

Método de los residuos Si se elimina de cualquier fenómeno alguna parte que se conoce que es el efecto de ciertos antecedentes, entonces lo que queda del fenómeno es el efecto de los antecedentes restantes.

Método de la variación concominante Si un fenómeno varía regularmente de la misma forma siempre que otro lo hace de una forma particular, el primero está conectado con en segundo por alguna cadena de causalidad.

Estos métodos se centraban básicamente en la determinación causal como elemento clave en la investigación científica, aunque los mismos están bastante relacionados con la construcción de programas capaces de aprender (en el sentido que se vio en la definiciones de aprendizaje dadas al comienzo de este capítulo) como se verá a continuación. Forsyth (Forsyth, 1989, pág. 7) reescribe los métodos anteriores utilizando una notación utilizada en Aprendizaje Automático y se puede ver que estos métodos tienen una relación con muchos algoritmos recientes. Reescribiendo los métodos en base a probabilidades se tiene que:

1. $P(A|C) = 1 \Rightarrow P(C|A) \gg 0$
2. $P(no\ A|no\ C) = 1 \Rightarrow P(C|A) \gg 0$
3. $P(no\ A|C) = 0$ o $P(A|no\ C) = 0 \Rightarrow P(C|A) \gg 0$
4. $A = F(C) \Rightarrow C = f(A)$

En realidad el cuarto método es el más general e incluye a los tres anteriores ya que la función puede ser bastante compleja, y uno de los elementos que se buscan en los programas de Aprendizaje Automático es esta función de concomitancia que se establece en el cuarto método propuesto por John Stuart Mill. Aunque también existen otros elementos que conforman estos programas aparte de la función que relaciona la entrada con la salida.

Bertrand Russell también estudió el principio de la inducción del que dijo no era susceptible de ser probado con algún fundamento a partir de la experiencia, ya que consideró que este principio es fundamentalmente probabilístico. Si dos cosas se encuentra que van juntas muchas veces y nunca por separado, entonces "un número suficiente de asociaciones hará la probabilidad de una nueva asociación casi cierta, y hará que se aproxime a la certeza sin límite". La contribución de Wittgenstein ha sido fundamental para enfatizar la simplicidad. En uno de sus tratados establece que "el procedimiento de inducción consiste en aceptar como verdad la ley más simple que pueda concordar con nuestra experiencia". Esta definición del proceso de inducción está bastante relacionada con el principio de la Cuchilla de Occam que ha influenciado bastante los trabajos en Aprendizaje Automático en los últimos años con bastantes trabajos en selección de atributos, y que es la base principal de esta tesis. Las aportaciones anteriores de la filosofía se pueden resumir en los cuatro elementos siguientes (Forsyth y Rada, 1986):

- No obviar las evidencias negativas.
- Buscar variaciones de concomitancia entre los factores casuales y los resultados.

- Cuantos más casos de asociación observados, mayor probabilidad de que la asociación sea cierta.

- Preferencia de las generalizaciones simples a las complejas.

Un problema que no fue estudiado en principio por los anteriores filósofos es cómo se selecciona la hipótesis que va a explicar los hechos observados. Una aportación en este aspecto es la que hace Wittgenstein retomando el principio de la Cuchilla de Occam que tiende a preferir la hipótesis más sencilla de las que explican unos hechos. Otro criterio es el propuesto por Karl Popper (Popper, 1959) que propone construir hipótesis que sean simples y fáciles de refutar. Este puede ser un criterio válido para la elección de las hipótesis en un sistema automático, pero para ello se debe formalizar la simplicidad y la facilidad de refutación, algo que no es sencillo ya que no existen medidas universales para estos términos.

Desde el punto de vista computacional, el objetivo de la inducción es encontrar un conjunto de reglas, considerando reglas en sentido general no solo lógicas, que puedan explicar unos hechos. Este conjunto de reglas puede ser en principio infinito, por lo que es necesario hacer uso de algún conocimiento previo del problema para acotar este número de posibles hipótesis, y establecer una preferencia por una hipótesis frente a otra. Estas preferencias se pueden hacer en dos sentidos (Michalski, 1983). Por un lado se puede hacer que la preferencia la definan una serie de propiedades que deben cumplir las hipótesis, este criterio es necesario cuando el conjunto de las reglas utilizadas para explicar los hechos es completo, es decir, pueden expresar cualquier hipótesis. Otra posibilidad es definir una preferencia mediante la restricción de las reglas que pueden utilizarse para expresar las hipótesis.

El aprendizaje inductivo se puede dividir en dos categorías: *Aprendizaje Inductivo No Supervisado* y *Aprendizaje Inductivo Supervisado*. En el aprendizaje inductivo supervisado, o simplemente aprendizaje supervisado, al sistema se le proporciona un conjunto de hechos etiquetados y el sistema debe obtener el conjunto de reglas que expliquen estos hechos. La segunda categoría se le denomina aprendizaje no supervisado ya que los casos utilizados en el proceso de aprendizaje no se encuentran etiquetados como pertenecientes a un concepto concreto. El objetivo en este tipo de sistemas es la obtención de descripciones que especifiquen las propiedades comunes a los objetos que pertenecen a una misma clase, esto se puede conseguir mediante la formulación de una teoría que caracterice un conjunto de entidades, el descubrimiento de patrones de regularidad en los casos presentados al sistema o la realización de una descripción taxonómica de un conjunto de entidades.

1.6.1 Aprendizaje No Supervisado

Como se ha comentado anteriormente el aprendizaje no supervisado tiene como objetivo obtener una descripción que permita caracterizar un conjunto de objetos. Al no encontrarse etiquetados los objetos o entidades, es tarea del sistema encontrar las clases o conceptos que se encuentran recogidos en dicho conjunto. La descripción puede ser jerárquica, existiendo conceptos generales además de otros más particulares o subconceptos que se especializan en casos cada vez más concretos. Otra forma de aprendizaje no supervisado es el relacionado con la obtención de un plan mediante la interacción del sistema en un determinado entorno y utilizando la realimentación que el sistema obtiene de éste para modificar su comportamiento y obtener una mejor solución al problema. A continuación se comentan en más detalle estos dos tipos de aprendizaje no supervisado.

Agrupamiento Conceptual

En esta categoría de aprendizaje no supervisado se parte de un conjunto de descripciones de objetos para obtener como resultado un esquema de clasificación de estos objetos. La clasificación puede consistir en: i) encontrar un agrupamiento de los objetos en conceptos, ii) obtener una descripción intencional para cada uno de los conceptos obtenidos, o iii) determinar una organización jerárquica de los objetos. La estructura más utilizada en el agrupamiento conceptual son las redes de discriminación donde cada nodo de esta red corresponde con un concepto, de forma que cualquier nodo corresponde a un concepto más concreto que el concepto representado en su nodo padre, siguiendo una relación *"es un"*.

Un elemento importante en el agrupamiento conceptual es el aprendizaje incremental, que no solo consiste en el procesamiento secuencial de los objetos, sino que la adición de un nuevo objeto no supone reprocesar todos los anteriores. Esto lleva a la integración del proceso de aprendizaje con el proceso de clasificación, ya que el aprendizaje viene dirigido por cada clasificación de un nuevo objeto. Esto supone una diferencia con otros esquemas de aprendizaje donde el proceso de aprendizaje está completamente diferenciado de la utilización del conocimiento adquirido, el cual se "congela" una vez se concluye la fase de aprendizaje.

En agrupamiento conceptual, las técnicas tradicionales utilizadas en agrupamiento o taxonomía numérica no son válidas porque la asignación de los objetos a las clases se realiza mediante la utilización de medidas de similaridad numéricas que consideran todos los atributos que definen los objetos. Además no toman en cuenta ningún tipo de contexto o concepto útil para hacer la caracterización de las clases o configuraciones de

objetos resultantes, dando lugar a clases que no poseen una interpretación fácil, es decir son medidas libres de contexto. Esto lo referencia Michalski (Michaslki y Stepp, 1983) como que "las medidas numéricas no pueden capturar las propiedades de Gelstalt de los agrupamientos de objetos", que son las propiedades que caracterizan a un agrupamiento como un todo y que no son derivables de las propiedades individuales. Para poder reconocer estas propiedades de Gelstalt el sistema debe disponer de la habilidad de reconocer configuraciones de objetos que correspondan a conceptos, mediante un proceso de búsqueda de un resumen o descripción de la clase correspondiente al concepto.

Como ya se comentó anteriormente, el conocimiento en el agrupamiento conceptual se representa como una red de discriminación y la clasificación de un nuevo objeto se realiza descendiendo por la jerarquía de conceptos desde el más general, que corresponde al primer nodo de la red. A diferencia de un árbol de decisión, en la red de discriminación se puede terminar en cualquier nivel ya que cada nodo corresponde con un concepto. Otra diferencia está en que el descenso por la red no se realiza en función del valor de un solo atributo. Una vez que se ha clasificado el objeto, se puede utilizar esa clasificación para realizar predicciones sobre aspectos desconocidos del objeto clasificado, ya que la pertenencia a una clase define una serie de aspectos de todos los elementos que pertenecen a la misma.

Entre los sistemas de agrupamiento conceptual clásicos nos podemos encontrar CLUSTER/2 (Michaslki y Stepp, 1983), COBWEB (Fisher, 1987) y CLASSIT (Gennari et al., 1989). Se describirá brevemente a continuación el sistema COBWEB que se ha utilizado como base para otros sistemas como CLASSIT o ISAAC (Talavera y Béjar, 1998). En COBWEB el conocimiento se almacena como en los otros sistemas en forma de red de discriminación donde cada nodo corresponde con un concepto. La diferencia que posee COBWEB en este aspecto con otros sistemas como CLUSTER, es que la descripción de los conceptos no se realiza mediante una expresión lógica como una disyunción, sino como un concepto probabilístico, que etiqueta cada nodo en el árbol de clasificación. Además en cada nodo del árbol se almacenan todos los atributos de los objetos.

La medida utilizada en COBWEB para realizar el proceso de clasificación es la denominada *category utility*, y que se utiliza en la búsqueda a través de la red de discriminación. Esta medida posee las propiedades que se buscan en cualquier medida utilizada en el agrupamiento numérico, similaridad de los objetos que forman una misma clase (intraclase) y disimilitud entre los objetos de diferentes clases (interclase). La

similaridad intraclase se mide mediante la probabilidad condicional,

$$P(X_i = x_{ij}|Y_k)$$

que corresponde con la probabilidad de que el objeto perteneciendo al concepto k tenga el valor x_{ij} para el atributo X_i. Cuanto mayor es esta probabilidad mayor es el número de objetos de la clase que comparten el valor, y más predecible es el valor del atributo para los miembros de la clase. La disimilitud interclase es una función de la probabilidad condicional,

$$P(Y_k|X_i = x_{ij})$$

En este caso, cuanto mayor es esta probabilidad, menor es el número de objetos en clases diferentes que comparten el mismo valor para un atributo y por tanto más predictivo es este valor para la clase.

Las anteriores probabilidades condicionales se promedian para todas las clases y todos los posibles valores de los atributos, obteniendo

$$\sum_{k=1}^{n}\sum_{i}\sum_{j} P(X_i = x_{i,j})P(X_i = x_{ij}|Y_k)P(Y_k|X_i = x_{ij}) = \sum_{k=1}^{n} P(Y_k)\sum_{i}\sum_{j} P(X_i = x_{ij}|Y_k)^2 \quad (1.1)$$

En la anterior expresión el término elevado al cuadrado corresponde al número esperado de valores de atributos que pueden ser estimados para un miembro cualquiera de la clase Y_k, ya que se supone que un valor puede ser estimado con una probabilidad igual a su probabilidad de aparición. La definición de la *category utility* se define como el incremento en el número esperado de valores de atributos que pueden ser correctamente estimados dada una determinada partición, con respecto al número de valores que se pueden estimar sin el conocimiento de dicha partición.

$$\text{category utility} = \frac{\sum_{k=1}^{n} P(Y_k)\sum_i\sum_j P(X_i = x_{ij}|Y_k)^2 - \sum_i\sum_j P(X_i = x_{ij}|Y_k)^2}{n}$$

En base a la medida anterior COBWEB incorpora de forma incremental los objetos al árbol de clasificación, es decir, cada objeto es presentado una sola vez al sistema y en función del mismo se va realizando el proceso de aprendizaje. La incorporación de un objeto al sistema se realiza utilizando alguno de los siguientes operadores:

1. Clasificar el objeto como perteneciente a una clase existente,
2. crear una nueva clase,
3. combinar dos clases en una sola, o
4. dividir una clase en varias clases.

Un inconveniente de COBWEB es que está diseñado solo para atributos nominales. CLASSIT utiliza el mismo esquema de COBWEB pero extendido para atributos continuos y para ello realiza una estimación de las distribuciones de probabilidad. Otro elemento que no incorpora COBWEB y si lo hace CLASSIT es la descripción estructural de los objetos que conforman la clase.

Aprendizaje por Refuerzo

El objetivo en este tipo de aprendizaje es la obtención en funcionamiento continua (*on-line*)de una función de asignación de un conjunto de datos de entrada a una determinada salida mediante un proceso de acierto-error en un entorno dinámico, sin la necesidad de un conjunto de muestras u objetos etiquetados, sino que es la evolución del agente en el entorno el que va obteniendo el refuerzo por el acierto o error de las acciones que realiza en dicho entorno. Por tanto la función objetivo en este aprendizaje debe maximizar un índice de rendimiento denominado señal de refuerzo. Un atractivo que presenta este tipo de aprendizaje es que se realiza mediante la asignación de recompensa o castigo sin necesidad de especificar de que manera se tiene que llevar a cabo la tarea a aprender. La existencia de esta recompensa en el aprendizaje por refuerzo implica que algunos autores no lo consideren como aprendizaje no supervisado sino como un tipo de aprendizaje que se encuentra a medio camino entre el supervisado y el no supervisado. El aprendizaje por refuerzo se puede dividir en asociativo y no asociativo. En el primero el objetivo es encontrar aquella función que realice la asociación entre una determinada entrada y la salida correspondiente, mientras que el segundo busca una solución que funcione de forma correcta en un determinado entorno.

En la Figura 1.4 se muestra un modelo del aprendizaje por refuerzo, se puede observar que el elemento de aprendizaje interactúa con el entorno mediante un bucle de percepción-acción, ya que recibe como entrada alguna indicación del estado del entorno de un conjunto de posibles estados $\mathcal{S}$ y puede elegir entre un conjunto de acciones $\mathcal{A}$, que modificará el estado de entorno. El valor de esta modificación se comunica al elemento de aprendizaje mediante un valor escalar denominado *señal de refuerzo, r*. El objetivo

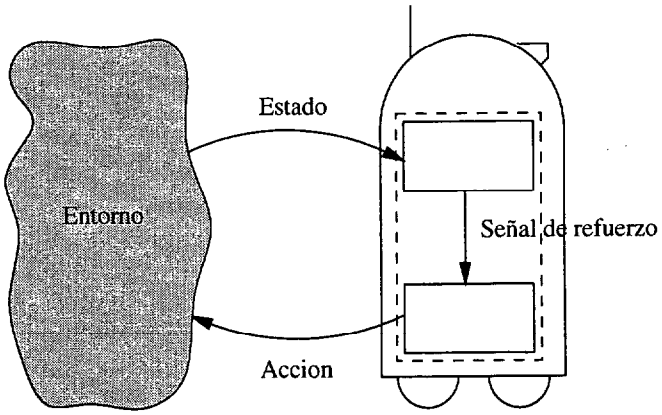


Figura 1.4: Aprendizaje por Refuerzo

es tomar en cada momento aquella acción que tienda a maximizar la suma de los valores de la señal de refuerzo a largo plazo. La función que realiza la asignación de estados a acciones es a lo que se denomina *política* π. Debido a que el entorno es dinámico, éste no será determinista en el sentido que encontrándose en un determinado estado y tomando una acción no siempre tiene que pasar al mismo estado, aunque si se suele considerar que el entorno es estacionario.

Como el objetivo del aprendizaje por refuerzo es encontrar una política π para seleccionar la secuencia de acciones que dé como resultado un comportamiento óptimo, es necesario tener un modelo del comportamiento óptimo. Uno de los más utilizados es el denominado Modelo de Descuento de Horizonte Infinito que toma en cuenta la recompensa R a largo plazo, aunque aquellas que se reciben en el futuro se descuentan de forma geométrica según un factor γ,

$$R = \sum_{t=0}^{\infty} \gamma^t r_t \tag{1.2}$$

donde $0 \leq \gamma < 1$ y r_t es la recompensa recibida t instantes de tiempo después.

En el caso general del problema del aprendizaje por refuerzo, las acciones no solo determinan la recompensa inmediata sino también el siguiente estado en el que se va a encontrar el entorno. Así, el agente debe tener capacidad de afrontar el problema del aprendizaje a partir de una recompensa retardada, ya que se puede tomar un número de acciones con poca recompensa para llegar a un estado con una gran recompensa, por lo que debe detectar qué acciones son preferibles basándose en una recompensa que puede venir en algún momento futuro. Este tipo de problemas pueden ser modelados como Procesos de Decisión de Markov (PDM) (Puterman, 1994), que formalmente puede

definirse como una tupla ($\mathcal{S}$,$\mathcal{A}$,R,T), donde:

- un conjunto de estados $\mathcal{S}$,
- un conjunto de acciones $\mathcal{A}$,
- una función de recompensa $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$,
- una función de transición de estados $T : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$.

La función de recompensa asigna a cada par de estado-acción un valor de recompensa, mientras que la función de transición de estados asigna una probabilidad de pasar a un determinado estado s' cuando el entorno tiene el estado s y se toma la acción a. Si las anteriores funciones son conocidas y utilizando técnicas de programación dinámica, se puede determinar la política óptima según el modelo de descuento de horizonte infinito (Bertsekas, 1987).

Sin embargo en el aprendizaje por refuerzo la función de recompensa y la función de transición de estados no es conocida y el sistema debe obtener una aproximación. Por lo tanto el principal problema con que se enfrenta el aprendizaje por refuerzo es la asignación temporal de créditos, cómo saber si la acción que se va a tomar es correcta, cuando ésta podría tener efectos a largo plazo. Una posibilidad es dejar evolucionar la situación hasta el final y evaluar la elección de dicha acción, sin embargo en muchos problemas no existe un final claramente definido. Otra opción es utilizar evidencias obtenidas de las iteraciones para ajustar el valor estimado de un estado basándose en la recompensa inmediata y el valor estimado del siguiente estado. Los algoritmos que se basan en esta aproximación se les conoce como métodos de Diferencias Temporales (Sutton, 1988).

Un algoritmo encuadrado dentro de los métodos anteriores es el *Adaptative Heuristic Critic (AHC)* (Barto et al., 1983), que está compuesto por dos elementos. Un elemento se denomina *crítico* y convierte la señal de refuerzo externa en una señal de refuerzo de más alto nivel. Esta nueva señal de refuerzo junto con el estado son las entradas al elemento de aprendizaje por refuerzo que actuará para maximizar el refuerzo que le pasa el *crítico*. Los dos elementos que componen el algoritmo AHC se pueden integrar conjuntamente en el algoritmo Q-learning (Watkings y Dayan, 1992). El funcionamiento de este algoritmo se basa en considerar correcta la elección de las acciones hasta el estado s, seleccionar la acción a de forma que el descuento esperado del refuerzo sea máximo.

1.6.2 Aprendizaje Supervisado

A diferencia del aprendizaje deductivo donde se dispone de unas reglas generales y a partir de ellas comprueba si casos particulares cumplen estas reglas generales. El aprendizaje inductivo el proceso es a la inversa ya que tiene como objetivo "la habilidad para generalizar a partir de la experiencia pasada de forma que se puedan abordar nuevas situaciones que se encuentran relacionadas con la experiencia acumulada" (Mitchell, 1980). Si la experiencia acumulada a la que hace referencia Mitchell se encuentra recogida como un conjunto de ejemplos pertenecientes a un concepto, se suele definir como *Aprendizaje Supervisado.* En el aprendizaje supervisado, a diferencia del no supervisado, la generalización a la que se hacía referencia anteriormente consiste en la adquisición de un determinado concepto o resolución de un problema como una secuencia de acciones. Aunque la descripción que se hará en esta sección estará centrada básicamente en el aprendizaje de conceptos es igualmente extensible a la resolución de problemas.

La inducción de conceptos en un sentido general es la obtención de reglas (mecanismos) de clasificación que permitan clasificar tanto los ejemplos utilizados en la inducción de las reglas como de nuevos ejemplos. Los ejemplos utilizados son de la forma $\langle \mathbf{X}^{(j}, Y^{(j} \rangle$, donde $\mathbf{X}^{(j}$ es un conjunto de observaciones o medidas e $Y^{(j}$ es la clasificación correcta que le corresponde y que es asignada por un tutor o supervisor externo al sistema, denominándose características o atributos al conjunto de medidas $\mathbf{X}^{(j}$. Las reglas se pueden considerar de forma genérica como una función $f(\mathbf{X}^{(j}) = Y^{(j}$ (en la práctica este tipo de función puede ser una función lógica, una lista de decisiones, una red neuronal artificial, una función lineal umbralizada, etc.).

En la Figura 1.5 se muestra el modelo propuesto por Forsyth (Forsyth, 1989) para el aprendizaje supervisado. La parte superior del modelo se corresponde a la supervisión de los casos por parte de un experto o tutor. Normalmente este proceso no se realiza simultáneamente con el proceso de aprendizaje, sino que los casos son presentados al experto para su clasificación y luego se almacenan los resultados o bien se procede a realizar un muestreo aleatorio de casos previamente clasificados por un experto. En la parte inferior se encuentra representado el esquema de aprendizaje que tiene como entrada los mismos casos que son dados al experto externo. Según las reglas de clasificación, el sistema da como salida una clasificación que se compara con la realizada por el experto, y en función de la diferencia entre ambas el proceso de aprendizaje modifica las reglas de clasificación existentes haciendo uso de una base de reglas que dispone a tal efecto.

El aprendizaje de conceptos se puede describir como un proceso de búsqueda a

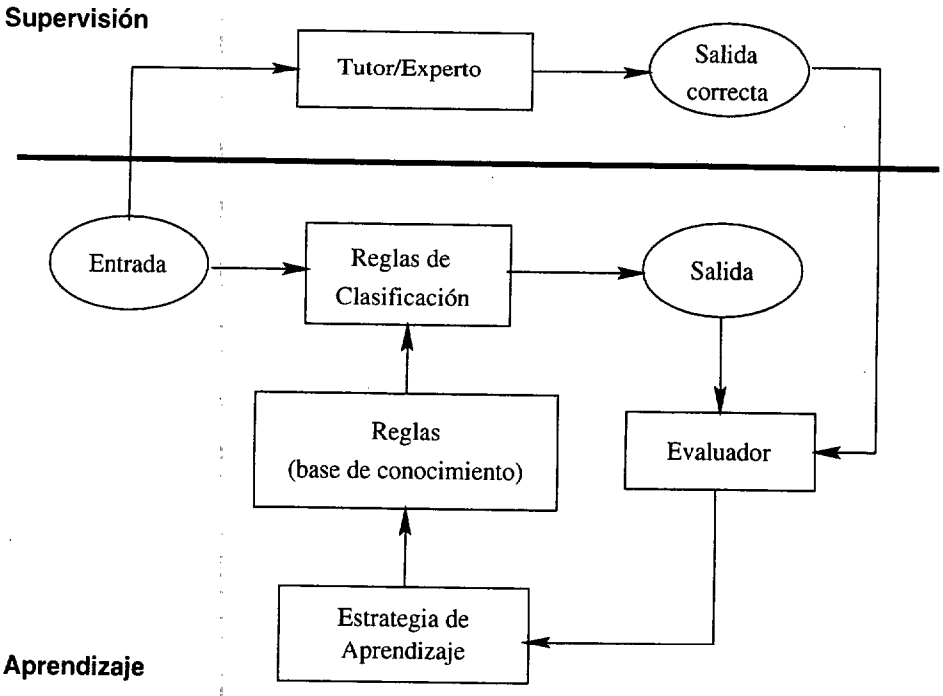


Figura 1.5: Esquema del aprendizaje supervisado.

través de un espacio de hipótesis que aproxima la función objetivo. Este espacio de búsqueda viene definido por la representación utilizada por la hipótesis y el objetivo es encontrar aquella que mejor se ajuste al conjunto de ejemplos. Por tanto un aspecto importante es la selección de la representación de las hipótesis, ya que esto define implícitamente el espacio de búsqueda y el conjunto de conceptos que pueden ser representados y por tanto aprendidos. Una estrategia para seleccionar la representación de las hipótesis puede ser la de aumentarlo tanto como sea posible para que permita de esta forma aprender todos los posibles conceptos. Sin embargo esta estrategia supone un crecimiento muy grande del espacio de búsqueda, aunque elimina cualquier restricción sobre el concepto a aprender, por lo que en principio cualquier concepto se va a poder expresar en dicho espacio. Suponiendo que el crecimiento del espacio de búsqueda se considere un problema asumible, existe otro problema y es que el algoritmo de aprendizaje será incapaz de obtener un concepto que permita generalizar más allá del conjunto utilizado para el aprendizaje.

Lo anterior es una propiedad de la inferencia inductiva (Mitchell, 1997): "un aprendiente que no haga ninguna suposición a priori sobre la identidad del concepto objetivo, no tiene una base racional para clasificar nuevas instancias de este concepto". Por tanto el algoritmo tiene que estar sesgado (*biased*) por la suposición de que el concepto objetivo puede ser representado por el tipo de hipótesis utilizado. Este sesgo que se introduce puede seguir uno de los siguientes modelos (Mitchell, 1980): *Modelo*

de Espacio de Hipótesis Reducido o *Modelo de Preferencia.* En el Modelo de Espacio de Hipótesis Reducido las reglas pertenecen a un espacio reducido de hipótesis y en el proceso de aprendizaje se selecciona aquella hipótesis que es consistente con el conjunto de ejemplos con el menor error posible. En el Modelo de Preferencia no se restringe el espacio de posibles hipótesis sino que se establece una ordenación por preferencia, de forma que se elige la hipótesis con mayor grado de preferencia y que sea consistente con el conjunto de ejemplos. Normalmente en este modelo se intenta minimizar algún tipo de medida de la complejidad sintáctica de la hipótesis.

En el modelo de Preferencia, una restricción que se utiliza con bastante frecuencia es la simplicidad y generalidad de las reglas generadas que conecta con el Principio de la Cuchilla de Occam (Blumer et al., 1987), que establece que si varias hipótesis explican los mismos hechos es preferible escoger aquel que sea más simple. En esta tesis se intentará seguir esta restricción de simplicidad, ya que se parte de la base que la explicación más sencilla es la que utiliza menor número de atributos siempre que la capacidad de generalización obtenida sea similar a la obtenida con un mayor número de atributos.

Aunque el concepto del sesgo introducido por Mitchell solo hace referencia a una condición necesaria para que el conocimiento inducido por un algoritmo de aprendizaje pueda clasificar entidades fuera del conjunto de aprendizaje, no hace referencia a qué problemas pueden ser aprendidos por un determinado conjunto de hipótesis, ni en cuanto tiempo. Existe un área dentro del aprendizaje automático denominada *Teoría del Aprendizaje Computacional* que estudia la complejidad de las muestras, complejidad computacional o cota de errores para distintos problemas y algoritmos de aprendizaje. Dentro de este campo un modelo bastante utilizado es el PAC (*Probably Aproximately Correct*) (Valiant, 1984). Este modelo se basa en la siguiente definición.

Definición 1.5. *Sea un concepto c definido sobre un conjunto de instancias X de longitud l y un algoritmo de aprendizaje L que utiliza el espacio de hipótesis H. El concepto c se puede aprender con probabilidad y aproximadamente correcto (PAC) por el algoritmo L, si para todo $c \in C$ definidos en X según una distribución de probabilidad $\mathcal{D}$, existe un ϵ tal que $0 < \epsilon < 1/2$ y un δ tal que $0 < \delta < 1/2$, el algoritmo de aprendizaje L dará con una probabilidad de al menos $(1 - \delta)$ como resultado una hipótesis $h \in H$, de forma que el error sobre todo el conjunto X será menor que ϵ, en un tiempo que es polinomial en ϵ, δ, l y el tamaño del concepto c.*

La definición anterior de un concepto que se puede aprender PAC establece por un lado que el algoritmo de aprendizaje debe dar con una probabilidad alta una hipótesis del concepto con poco error, y por otro lado que este resultado se obtiene en un tiempo que es

polinomial con los distinto parámetros que definen el problema, como las tolerancias de probabilidad, error, el tamaño del espacio de casos y del concepto. En el trabajo original de Valiant se demuestra que un concepto definido por atributos booleanos y que se pueda expresar como conjunción de los atributos que lo definen cumple la definición anterior. Sin embargo si el concepto es definido como una disyunción de los atributos, el concepto no se puede aprender en tiempo polinomial. Pero como se demuestra en el mismo trabajo, una clase de conceptos más general como es la conjunción de conjunciones de k términos sí se puede aprender en un tiempo polinomial. Extensiones a otro tipo de conceptos y dominios no solo booleanos y libres de ruido se pueden encontrar en (Natarajan, 1991; Kearns y Vazirani, 1994).

Otro modelo de la Teoría del Aprendizaje Computacional, es el *Mistake Bound* (Littlestone, 1988; Littlestone et al., 1991) que a diferencia del modelo PAC hace referencia a la cota de errores, es decir cual es el número de errores que comete un algoritmo de aprendizaje antes de converger a una hipótesis correcta. La hipótesis correcta puede ser en el sentido PAC antes comentado o bien la exacta. Para algoritmos denominados HALVING el número máximo de errores que comete antes de llegar a la solución exacta es $\log_2 |H|$, donde $|H|$ es la dimensión del espacio de hipótesis. Este tipo de algoritmos son aquellos que asignan la pertenencia al concepto de un ejemplo siguiendo un criterio de mayoría de las hipótesis candidatas en un determinado momento, de forma que si se comete un error al clasificar un ejemplo se debe a que más de la mitad de las hipótesis son erróneas y por tanto se eliminarían. Por tanto si en cada ejemplo mal clasificado se eliminan la mitad de las hipótesis, el número máximo de fallos es $\log_2 |H|$.

El cálculo del número óptimo de errores antes de que un algoritmo converja a una solución exacta viene determinado por la siguiente relación:

$$VC(C) \leq Opt(C) \leq M_{HALVING}(C) \leq \log_2 |H| \tag{1.3}$$

Donde $VC(C)$ es la dimensión Vapnik-Chervonenkis del conjunto de conceptos (Natarajan, 1991, pág. 18) y $M_{HALVING}(C)$ es el número máximo de errores que comete un algoritmo del tipo HALVING sobre todas las secuencias de ejemplos para el conjunto de conceptos C.

Un problema importante subyacente a cualquier tarea de aprendizaje inductivo es la información que se le proporciona al sistema. En el caso de Aprendizaje Inductivo esta información viene dada, como se comentó anteriormente, por ejemplos etiquetados. Cada uno de los ejemplos viene definido por un conjunto de descriptores, características o atributos. La determinación de qué descriptores o medidas son los relevantes para el

concepto que se quiere representar junto con la inducción de las reglas de clasificación son las dos tareas principales del problema del aprendizaje inductivo. La importancia que posee este proceso de seleccionar los atributos relevantes viene recogida en el siguiente párrafo (Michalski, 1983),

> "la determinación de esos descriptores es una parte importante de cualquier problema de aprendizaje inductivo. Si ellos capturan las propiedades esenciales de los objetos, el papel del proceso de aprendizaje es simplemente incorporar esos descriptores en una expresión que constituya una declaración apropiada. Si los descriptores seleccionados son completamente irrelevantes a la tarea de aprendizaje, ningún sistema de aprendizaje será capaz de construir una declaración útil."

En el mismo sentido se expresa Weiss (Weiss y Kulikowski, 1991) cuando define los atributos como "el conjunto de observaciones potencialmente relevantes a un problema dado", indicando que de todas las posibles medidas u observaciones que pueden extraerse para un problema no todas van a ser consideradas en el proceso de aprendizaje. Por tanto se establece la necesidad de una definición de qué atributos son los relevantes en un determinado problema de inducción y es el que va a ser tratado en esta tesis. Un análisis más detallado de este problema así como de diferentes soluciones aportadas por distintos autores se encuentra en el Capítulo 3 y la propuesta que aquí se hace se encuentra recogida en el Capítulo 4.

1.7 Aprendizaje de Conceptos y Procedimientos de Clasificación

En esta sección se definen por un lado algunos paradigmas y algoritmos de aprendizaje de conceptos o generalización y por otro algunos procedimientos de clasificación. Estos algoritmos corresponden al aprendizaje supervisado, marco en el que va a estar encuadrada esta tesis. Entre los distintos paradigmas (ver página 8) nos centraremos en los métodos simbólicos, redes neuronales artificiales y algoritmos genéticos.

1.7.1 Métodos Simbólicos

Algoritmo CANDIDATE-ELIMINATION y AQ11

Uno de los primeros algoritmos utilizados en aprendizaje supervisado simbólico es el CANDIDATE-ELIMINATION (Mitchell, 1980), que implementa el concepto de Espacio de Versiones. El Espacio de Versiones es el conjunto de todas las hipótesis que son consistentes con el conjunto de aprendizaje y contiene todas las versiones plausibles del concepto. Este esquema de aprendizaje es utilizado para el aprendizaje de conceptos donde cada muestra puede pertenecer al concepto o no, siendo necesario que en el conjunto de aprendizaje existan ejemplos y contraejemplos del concepto a aprender. El concepto se define como una conjunción de varios atributos de los que definen las muestras por lo que el conjunto de hipótesis son todas las posibles conjunciones que se puedan realizar con los atributos. El funcionamiento del método se basa en la definición de la relación de orden parcial *más general que.*

Definición 1.6. *Si h_j y h_k son funciones lógicas definidas sobre $\mathcal{X}$, entonces h_j* ***es más general o igual que*** *h_k si y solo si:*

$$\forall \mathbf{X} \in \mathcal{X} \ : \ h_k(\mathbf{X}) = 1 \rightarrow h_j(\mathbf{X}) = 1$$

Definición 1.7. *Si h_j y h_k son funciones lógicas definidas sobre $\mathcal{X}$, entonces h_j* ***es más general*** *h_k si y solo si h_j es más general o igual que h_k y h_k no es más general o igual que h_j.*

Una vez establecidas las anteriores definiciones, el funcionamiento del procedimiento se basa en reducir el Espacio de Conceptos al Espacio de Versiones, que está representado por los miembros más generales G y más específicos S. La hipótesis más general es aquella que es consistente con todas las muestras del conjunto de aprendizaje y no existe otra más general que también sea consistente con el conjunto de aprendizaje, es decir la hipótesis más general que no cubre ninguna muestra negativa. La hipótesis más específica por el contrario es aquella que es consistente con el conjunto de aprendizaje pero no existe otra que sea menos general y que también sea consistente, o lo que equivale a la hipótesis más específica que cubre todos los ejemplos positivos. El Espacio de Versiones es el subconjunto del Espacio de Conceptos limitado por todas las hipótesis más generales y por las más específicas.

Un método para construir el espacio de versión consiste en la inicialización de S a la hipótesis más específica de todas, es decir una hipótesis que no contenga ningún

ejemplo y G a la más general, es decir que sea consistente con todos ejemplo. Luego a medida que se van encontrando ejemplos pertenecientes al concepto y no pertenecientes, se va haciendo más específico el conjunto G y más general el conjunto S de forma que al final solo quedan aquellas hipótesis que son consistentes con el conjunto de aprendizaje.

Este método es muy sensible al ruido, y presupone la no existencia del mismo en el conjunto de aprendizaje, ya que en otro caso el Espacio de Versiones puede no contener hipótesis consistentes con el conjunto de muestras. En algunas situaciones en lugar de una sola hipótesis, el Espacio de Versiones resultante incluye más de una. En este caso la clasificación de una nueva muestra viene dada por el procedimiento de la mayoría, que consiste en clasificar la muestra como perteneciente a la clase si cumple la mayoría de las hipótesis, en caso contrario no se considera como perteneciente al concepto. Este procedimiento como procedimiento teórico es óptimo en el sentido que obtiene el concepto siempre que sea expresable por el espacio de hipótesis utilizado (conjunciones lógicas, etc.) pero como el ruido influye bastante en los resultados es válido para dominios bien definidos como integración numérica o juego del ajedrez. Una implementación del algoritmo CANDIDATE-ELIMINATION es el *Focussing* (Mitchell, 1997).

Como se comentó anteriormente el algoritmo CANDIDATE-ELIMINATION fue diseñado para el aprendizaje de conceptos, sin embargo no es válido para problemas de clasificación donde las muestras puedan pertenecer a más de una clase. El algoritmo AQ11 (Michalski y Chilausky, 1980) basándose en el anterior permite resolver problemas de clasificación. Para ello primero se separan todas las muestras del conjunto de aprendizaje por clases, y luego aplica el algoritmo CANDIDATE-ELIMINATION a cada conjunto considerando las muestras que contiene como perteneciente al concepto y el resto como contraejemplos de ese concepto. Un inconveniente que aparece si se aplicara el método tal y como se ha indicado es que las hipótesis obtenidas para las distintas clases pueden no ser mutuamente exclusivas. Esto supone que al clasificar una nueva muestra, ésta pueda clasificarse como perteneciente a más de una clase. Para evitar esto el algoritmo AQ11 va utilizando las hipótesis que se van aprendiendo como contraejemplos de las nuevas clases. Así se asegura que las nuevas hipótesis que se van obteniendo no se solapan con las ya existentes. Una característica no deseable de este algoritmo es que el resultado depende del orden en que se muestran las clases, ya que para las primeras clases las hipótesis van a ser más generales que las obtenidas para las últimas clases, aparte que heredan todas las restricciones del algoritmo CANDIDATE-ELIMINATION.

Aunque el anterior algoritmo CANDIDATE-ELIMINATION ha sido utilizado en problemas reales con buenos resultados, no es muy útil en situaciones donde el ruido

afecte a las muestras debido a ser muy sensible al ruido.

Algoritmo ID3

Un algoritmo de aprendizaje de mecanismos de clasificación en vez de generalización de conceptos es el ID3 y su evolución el C4.5 (Quinlan, 1986; Quinlan, 1993) para problemas con atributos nominales. Este algoritmo construye un árbol de decisión similar al algoritmo CLS (Hunt et al., 1966), aunque a diferencia de éste incluye dos mejoras que son: el procedimiento de ventana que permite trabajar eficientemente con conjuntos grandes de muestras de aprendizaje, y por otra parte utiliza una heurística basada en Teoría de la Información.

Antes de introducir el procedimiento para construir el árbol de decisión, se explicará la clasificación utilizando este tipo de métodos, que tiene cierta similitud con el proceso de clasificación utilizado en agrupamiento conceptual (pág. 22) con redes semánticas. El árbol consiste en un conjunto de nodos, donde cada nodo no hoja contiene una condición sobre un atributo de los que definen el problema. La clasificación de un nuevo ejemplo se realiza comenzando por la raíz del árbol y comprobando en cada nodo el valor del atributo asociado a dicho nodo y descendiendo por la rama correspondiente al valor del atributo para el ejemplo a clasificar. Por tanto cada nodo tiene tantos nodos hijo como valores puede tomar el atributo asociado al nodo. La clasificación del ejemplo viene dada por el nodo hoja al cual se llega después de descender por el árbol.

Un ejemplo de un árbol de decisión se puede ver en la Figura 1.6 y que se corresponde al generado a partir del conjunto de aprendizaje mostrado en la Tabla 1.1. Recorriendo el árbol se obtiene que se jugará al tenis si está nublado, o si está cubierto y hay poca humedad, o si está lloviendo y el viento es flojo. La transformación de un árbol de decisión como un conjunto de reglas es inmediata, aunque para problemas complejos el árbol puede tener una profundidad apreciable lo que dará como resultado reglas con unos antecedentes grandes.

Dentro de los algoritmos de inducción de árboles de decisión se encuentran el CART propuesto por Breiman (Breiman et al., 1993) y el ID3 y C4.5 propuestos por Quinlan (Quinlan, 1986; Quinlan, 1993). El C4.5 es una evolución del ID3 aunque comparten básicamente el mismo esquema de aprendizaje. A continuación se expondrá brevemente el funcionamiento de ID3 para la generación de árboles de decisión y que realiza mediante un proceso de partición de las muestras de aprendizaje según los valores de un determinado atributo, y es esa partición la que define la estructura del árbol.

El procedimiento se basa en una técnica de divide y vencerás, ya que en cada

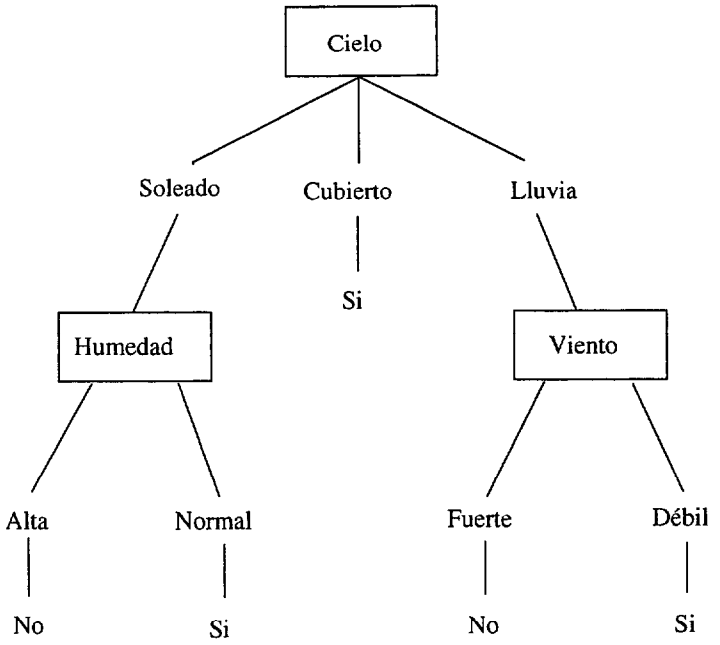


Figura 1.6: Árbol de decisión

nodo se selecciona un atributo que no ha sido previamente seleccionado y se particiona el conjunto de aprendizaje en función de los valores que puede tomar el atributo seleccionado. A continuación se crean tantos nodos como particiones, y sobre cada uno de los nodos se repite el proceso utilizando como conjunto de aprendizaje la partición correspondiente. Este proceso se repite recursivamente hasta que todas las muestras de una partición tengan la misma etiqueta.

La selección de qué atributo se utiliza en cada nodo para realizar la partición se hace en función de diferentes medidas, entre las que cabe destacar la información mutua (Quinlan, 1986), la relación de ganancia (Quinlan, 1993), el índice de Gini (Breiman et al., 1993) o la distancia de Mántaras (López de Mántaras, 1991). Hacemos notar que cualquiera de estas medidas puede ser utilizada en un entorno diferente al del ID3 para realizar un proceso de selección de atributos, ya que todas intentan seleccionar como raíz de cada subárbol al atributo más relevante. En la parte experimental de esta tesis se ha utilizado la distancia de Mántaras en la comparativa ya que de todas las medidas es la que guarda mayor similitud conceptual a la propuesta que realizamos.

Aprendizaje Bayesiano

Desde el punto de vista de la Filosofía, Bertrand Russell (pág. 18) consideraba el proceso de inducción como un procedimiento básicamente probabilístico. Precisamente el fundamento en el que se basa el aprendizaje bayesiano es también probabilístico, ya que

la observación de nuevos ejemplos da lugar al aumento o disminución de la probabilidad de una hipótesis, y no descartarla por completo como en otros algoritmos como el CANDIDATE-ELIMINATION o el ID3.

El aprendizaje bayesiano y más concretamente el clasificador bayesiano, han sido bastante utilizado en Reconocimiento de Formas (Duda y Hart, 1973; Tou y Gonzalez, 1974; Fukunaga, 1990) aunque aquí se presenta en el contexto del aprendizaje de conceptos. Para esto se hace uso del espacio de hipótesis y el objetivo es encontrar la hipótesis más probable conocido un conjunto de aprendizaje, mediante el uso del teorema de Bayes y conociendo las probabilidades a priori de las distintas hipótesis y del conjunto de aprendizaje. Así, se denomina $P(h)$ a la probabilidad de la hipótesis h, $P(\mathcal{D})$ a la probabilidad del conjunto de aprendizaje y $P(\mathcal{D}|h)$ a la probabilidad de observar el conjunto de datos $\mathcal{D}$ cuando verifican la hipótesis h; el objetivo del aprendizaje bayesiano es obtener la hipótesis más probable dado el conjunto de aprendizaje $\mathcal{D}$, a partir de, $P(h|\mathcal{D})$ que se denomina probabilidad a posteriori de la hipótesis, y que se puede obtener como:

$$P(h|\mathcal{D}) = \frac{P(\mathcal{D}|h)P(h)}{P(\mathcal{D})}$$

En la expresión anterior se puede observar que a medida que aumenta $P(h)$ o $P(\mathcal{D}|h)$ la probabilidad a posteriori también aumenta. Sin embargo si aumenta la probabilidad del conjunto de aprendizaje la probabilidad a posteriori disminuye, esto es comprensible ya que cuanto más probable sea $\mathcal{D}$ independientemente de h menos probable es la evidencia de que h dependa de $\mathcal{D}$.

Por tanto un algoritmo de aprendizaje que utilice un espacio de hipótesis H y que obtenga como resultado la hipótesis más probable conocido el conjunto de aprendizaje, debe obtener la hipótesis con Máxima Probabilidad a Posteriori (h_{MAP}).

$$\begin{aligned} h_{MAP} &= \max_{h \in H}\{P(h|\mathcal{D})\} = \\ &= \max_{h \in H}\left\{\frac{P(\mathcal{D}|h)P(h)}{P(\mathcal{D})}\right\} = \\ &= \max_{h \in H}\{P(\mathcal{D}|h)P(h)\} \end{aligned} \tag{1.4}$$

En muchos problemas de aprendizaje no existe una evidencia que dé mayor preferencia a una hipótesis frente a otra por lo que a priori todas tienen la misma probabilidad, pudiéndose eliminar de la ecuación (1.4) la probabilidad a priori de las hipótesis y la obtención de la h_{MAP} equivale a encontrar el máximo de $P(\mathcal{D}|h)$, lo que se conoce como

Verosimilitud del Conjunto de Aprendizaje conocida la hipótesis, y cualquier hipótesis que maximice la anterior probabilidad se le conoce como de Máxima Verosimilitud (h_{ML}).

$$h_{ML} = \max_{h \in H} \{P(\mathcal{D}|h)\}$$

La utilización de las anteriores expresiones para el aprendizaje de conceptos puede hacerse enumerando todas las hipótesis y haciendo uso del teorema de Bayes para obtener la h_{MAP} o h_{ML}, que son equivalentes si todas las hipótesis son a priori equiprobables. Pero este algoritmo, denominado de Fuerza Bruta, es bastante costoso si el espacio de hipótesis tiene una gran dimensión. Sin embargo se puede demostrar (Mitchell, 1997) que cualquier algoritmo de aprendizaje que sea consistente, es decir aquel que no comete ningún error sobre el conjunto de aprendizaje, obtiene como resultado una hipótesis que es h_{MAP} siempre que se consideren las hipótesis con igualdad de probabilidad a priori y un conjunto de aprendizaje exento de ruido y determinista, es decir, $P(\mathcal{D}|h) = 1$ si h es consistente con $\mathcal{D}$.

Hasta ahora el problema que se ha visto es el de obtener la hipótesis más probable, sin embargo en problemas de clasificación, lo que se busca es la clasificación más probable de una muestra conocido un conjunto de aprendizaje. En principio una solución podría ser la de asignar la muestra a la hipótesis más probable, pero no siempre es éste el resultado óptimo. El mejor resultado se obtiene teniendo en cuenta las predicciones realizadas por todas las hipótesis. Así, si una muestra puede pertenecer a una clase $y_k \in \mathcal{Y}$, la probabilidad de pertenencia correcta conocido el conjunto de aprendizaje $\mathcal{D}$ es

$$P(y_k|\mathcal{D}) = \sum_{h_i \in H} P(y_k|h_i)P(h_i|\mathcal{D}) \tag{1.5}$$

Por tanto la clasificación óptima de la muestra es la máxima de las anteriores,

$$\max_{y_k \in \mathcal{Y}} \left\{ \sum_{h_i \in H} P(y_k|h_i)P(h_i|\mathcal{D}) \right\}$$

Cualquier sistema que utilice la regla anterior para realizar el proceso de clasificación se le denomina *Clasificador Bayesiano Óptimo*, y se demuestra (Duda y Hart, 1973) que ningún otro método utilizando el mismo espacio de hipótesis y el mismo conocimiento a priori del problema puede mejorar este método en promedio.

Un inconveniente en el uso del clasificador bayesiano óptimo es la estimación de las densidades de probabilidad que aparecen en la expresión anterior que pueden corresponder a variables aleatorias de una alta dimensión y aparece por tanto la denominada *Maldición de la Dimensionalidad de Bellman* (*Bellman's Curse of Dimensionality*) (Bellman, 1957), ya que el número de muestras necesarias para estimar las funciones de probabilidad en (1.5) crece exponencialmente con la dimensión del vector de atributos.

Una aproximación que se suele utilizar en problemas de clasificación como los que se van a tratar en esta tesis y que se pueden encontrar en otros campos como el Reconocimiento de Formas es el denominado *Clasificador "Naive Bayes"* o *Bayesiano Simplificado* en el que las muestras están definidas con un conjunto de atributos como el definido en la Sección 1.1. En este caso lo que se intenta obtener es la asignación a la clase más probable conocida la muestra:

$$\mathbf{X} = \mathbf{x} \rightarrow y_k \text{ si } P(y_k|\mathbf{X}) = \max_{y_j}\{P(y_j|\mathbf{X})$$

Aunque para hacer uso de la expresión anterior también es necesario estimar distribuciones de probabilidad multivariables, el clasificador Naive Bayes se basa en la simplificación de que todos los atributos son independientes por lo que la probabilidad condicional anterior se convierte en un producto de probabilidades condicionales pero dependientes de una variable aleatoria unidimensional.

$$\mathbf{X} = \mathbf{x} \rightarrow y_k \text{ si } P(y_k|\mathbf{X}) = \max_{y_j}\{P(\mathbf{X}|y_j)P(y_j)\} = \max_{y_j}\{P(y_j)\prod_{i=1}^{n} P(X_i|y_j)\} \qquad (1.6)$$

Es evidente que la aproximación (1.6) es mejor cuanto más se acerquen los atributos a la condición de independencia.

La cuestión que se plantea ahora con este clasificador es la estimación de las probabilidades utilizadas. En el caso de atributos con valores discretos, la estimación de probabilidades se puede realizar por el simple recuento de los casos. Para el caso de atributos continuos la estimación se puede hacer considerando que siguen una distribución normal, con lo que se estima la media y varianza de dicha distribución. Otra posibilidad es la utilización de la series de funciones continuas (Duda y Hart, 1973), como pueden ser el método de las ventanas de Parzen. En este trabajo se ha utilizado en el apartado experimental la implementación que hace de este clasificador la librería $\mathcal{MLC}++$ (Kohavi et al., 1996), donde la estimación para atributos nominales se hace mediante el recuento de los casos y para los continuos mediante la asunción de la distribución

normal.

La aplicación del clasificador Naive Bayes al conjunto de aprendizaje mostrado en la Tabla 1.1 para decidir si jugar al tenis con las siguientes condiciones meteorológicas $\langle cielo\ soleado, frio, humedad\ alta, con\ viento \rangle$; consistiría en obtener el conjunto de probabilidades a partir de los datos del conjunto de aprendizaje y a partir de las mismas y haciendo uso de la regla expresada en la ecuación (1.6) tomar la decisión. Una vez calculadas las probabilidades se obtiene que:

$$P(si)P(soleado|si)P(frio|si)P(hum.alta|si)P(viento|si) = 0.0053$$
$$P(no)P(soleado|no)P(frio|no)P(hum.alta|no)P(viento|no) = 0.206$$

Por lo que la decisión será la de no jugar, ya que el clasificador asigna la mayor probabilidad a esta opción a partir de las probabilidades estimadas en el conjunto de aprendizaje.

Clasificadores Basados en Ejemplos

A diferencia de los métodos anteriores, en este tipo de clasificadores no se construye un modelo para todo el espacio de hipótesis sino que se realizan distintas aproximaciones definidas en determinadas zonas del espacio. Esta característica hace estos métodos adecuados para aproximaciones de funciones complejas y difíciles de modelar en su totalidad. El modelado a trozos de la función viene dado por el modo de realizar la clasificación de estos métodos, en los que se considera cada muestra como un punto del espacio $\mathcal{X}$ y para la clasificación de una nueva muestra se recupera un conjunto de las instancias almacenadas en memoria.

El caso más sencillo de este tipo de clasificadores es el *k Vecinos más Cercanos (k-NN)* (Duda y Hart, 1973). En este clasificador no se asume ninguna función implícita sino que la clasificación se realiza para cada nueva muestra, mediante la asignación de dicha muestra a la clase que es mayoritaria en las k muestras más próximas del conjunto de aprendizaje, siendo el caso más sencillo cuando se asigna a la clase de la muestra más cercana. El modelado a trozos de la función se puede observar en este caso mediante la utilización de los diagramas de Voronoi (Fig. 1.7), donde todos los puntos dentro del mismo poliedro se encuentran a la mínima distancia de la muestra de aprendizaje circunscrita por el poliedro. Una familia de clasificadores basados en este modelo son los IB propuestos por Aha (Aha et al., 1991).

Un elemento importante que define el resultado de este tipo de métodos es la fun-

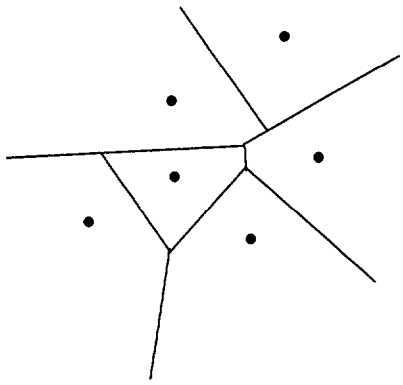


Figura 1.7: Diagrama de Voronoi para el clasificador del vecino más próximo

ción de distancia utilizada para la búsqueda de los vecinos más próximos. Normalmente se utiliza la distancia euclídea para el caso de los atributos continuos o la distancia de Hamming si se trata de atributos nominales, aunque se han propuesto otro tipo de distancias como la K*, propuesta por Cleary (Cleary y Trigg, 1995), basada en Teoría de la Información. Esta distancia se basa en el cálculo de la entropía de las posibles transformaciones que se pueden utilizar para llegar a una determinada muestra partiendo de otra.

El proceso de aprendizaje en este tipo de clasificadores consiste en almacenar todas las muestras, ya que el concepto se almacena como pares $\langle \mathbf{X}^{(j)}, Y^{(j)} \rangle$, que se corresponden con los elementos del conjunto de aprendizaje. La clasificación de una muestra se realiza mediante el cálculo de su distancia a todas las muestras almacenadas, y asignar la misma etiqueta de la que se encuentra más cercana. El caso anterior consiste en la clasificación por el vecino más próximo, sin embargo si se consideran los k vecinos más próximos, la asignación se realiza de acuerdo a la etiqueta que es mayoritaria en los k vecinos más próximos. Un ejemplo entre la diferencia del vecino más próximo y los k vecinos más próximos se puede ver en las Figuras 1.8 y 1.9. En la primera se asigna la etiqueta positiva a la muestra incógnita $\mathbf{X}_?$, mientras que en la segunda se asigna la etiqueta negativa ya que de los 5 vecinos más próximos cuatro de ellos poseen esta etiqueta.

Como se puede deducir fácilmente, el proceso de clasificación conlleva el mayor coste computacional ya que se deben calcular las distancias de todas las muestras almacenadas a la muestra a clasificar. Por tanto una mejora importante en la velocidad en este tipo de métodos se puede conseguir mediante la utilización de técnicas de indexación eficientes de las muestras de aprendizaje almacenadas en memoria. Un método que permite recuperar los vecinos más próximos de forma más eficiente a costa de un mayor uso de memoria es el *kd-tree* (Samet, 1990). En este método las muestras de aprendizaje se

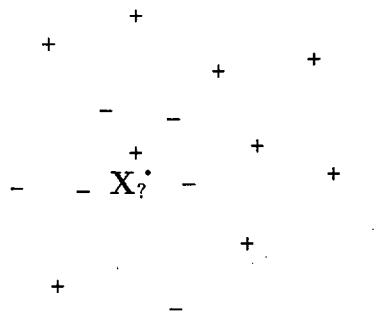


Figura 1.8: Clasificación según la estrategia del vecino más próximo

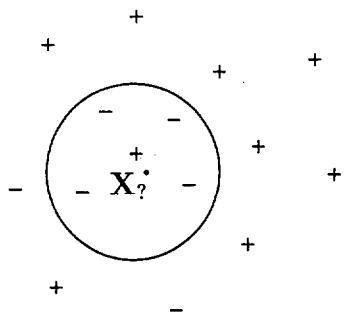


Figura 1.9: Clasificación según la estrategia de los 5 vecinos más próximos

almacenan en los nodos de un árbol de forma que las muestras más cercanas se encuentran en el mismo nodo hoja o en un nodo hoja cercano, mientras que los nodos internos al árbol realizan la ordenación de la muestra a clasificar hacia los nodos hojas cuyas muestras se pueden encontrar más cercanas. Otras alternativas para reducir el tiempo de clasificación pueden ser reducir el conjunto de muestras almacenadas manteniendo siempre la consistencia (Ritter et al., 1975; Toussaint et al., 1984), reduciendo de esta forma el número de cálculos de distancias a realizar. Una propuesta debida a Sánchez (Sánchez et al., 1997) se basa en obtener a partir del conjunto de muestras un árbol de decisión equivalente al clasificador del vecino más cercano, haciendo corresponder con cada nodo del árbol una frontera de decisión de las que se obtendría en el clasificador más cercano.

Un aspecto de estos métodos que puede ser un inconveniente es que en el cálculo de la distancia todos los atributos poseen igual peso y se utilizan todos en el proceso de clasificación. Esto puede llevar a errores en el proceso de clasificación en casos donde exista un gran número de atributos irrelevantes. Por ejemplo en un problema de dimensión 20 donde solo 2 atributos sean relevantes, dos muestras con igual valor de los atributos relevantes pueden ser clasificadas de diferente forma si el resto de los atributos no relevantes tienen diferentes valores. Para evitar este inconveniente se han propuesto técnicas que asignen distintos pesos a los distintos atributos, que correspondería a un acortamiento de los ejes asociados a atributos poco relevantes y al alargamiento de los asociados a atributos más relevantes. Este concepto de modificar los ejes en función de su importancia ha sido utilizada previamente en la distancia estadística de Mahalanobis (Duda y Hart, 1973) que hace uso de la matriz de covarianza para modificar el ángulo entre los ejes asociados a las distintas variables en función de su grado de correlación. Una medida que se ha utilizado para realizar la ponderación de los atributos, aparte de la correlación cruzada entre los atributos como en la distancia de Mahalanobis, ha sido la información mutua (Wettschereck y Dietterich, 1995), de forma que aquellos

atributos cuya información mutua con la clase sea mayor tienen más peso en el cálculo de la distancia euclídea que otros con menor información mutua. Debido a que el tema principal de esta tesis es la selección de atributos, el tratamiento de este problema se realizará en mayor profundidad en el Capítulo 3.

Este tipo de métodos tiene una estrecha relación con el Razonamiento Basado en Casos (Sección 1.5), ya que en este último el proceso de funcionamiento (Fig. 1.3) incluye la utilización de casos almacenados en la memoria de casos, recuperando aquellos que poseen una mayor similitud para resolver el problema incógnita, aunque se utiliza en general relaciones más complejas que una medida de distancia. A estos clasificadores se les suele denominar algoritmos perezosos (*lazy learners*) ya que suelen exhibir las siguiente características (Aha, 1997):

- Posponen el procesamiento de las entradas hasta que se recibe una solicitud de información ya que la información de entrada se mantiene almacenada para un uso posterior.
- Las solicitudes de información se obtienen mediante la combinación de la información almacenada en forma de muestras.
- Se descarta la estructura construida para responder a una solicitud de información y los posibles resultados intermedios.

1.7.2 Redes Neuronales Artificiales

El estudio de las redes neuronales artificiales se remonta a los años cuarenta a partir del trabajo de McCulloch y Pitts (McCulloch y Pitts, 1943), donde se describe un cálculo lógico de las redes neuronales. En el libro de Hebb, *The Organization of Behaviour*, el autor incluye lo que luego se ha conocido como la regla de aprendizaje Hebbiano que establece que el peso asociado a una conexión (sinapsis) entre dos neuronas aumenta a medida que la activación de una de ellas proviene de la otra a través de dicha sinapsis. Quince años más tarde de la aparición del trabajo de McCulloch y Pitts, Rosenblatt (Rosenblatt, 1958) da una nueva aproximación al reconocimiento de patrones con la presentación del *Perceptrón* y del Teorema de Convergencia del Perceptrón. Relacionada con el perceptrón se encuentra la regla de Widrow que se basa en utilizar el mínimo error cuadrado promedio utilizado en el *Adaline* (Widrow y Hoff, 1960), que únicamente se diferencia del perceptrón en el método de aprendizaje. Una versión de múltiples capas del Adaline es el Madaline (Widrow, 1962). Rumerlhart (Rumelhart et al., 1986*b*) propone el Algoritmo de Retropropagación del Error que es uno de los

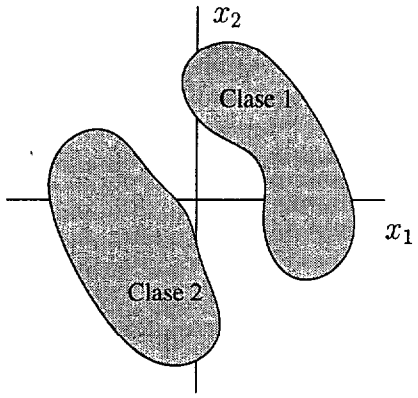


Figura 1.10: Clases linealmente separables

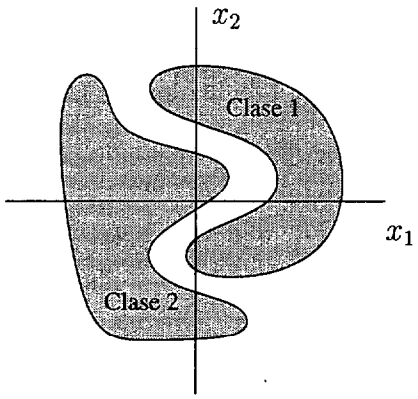


Figura 1.11: Clases no linealmente separables

algoritmos más utilizados para el entrenamiento de perceptrones multicapa. En 1988, Broomhead (Broomhead y Lowe, 1988) describe un procedimiento para la construcción de redes neuronales utilizando funciones de base radial que proporciona una alternativa a los perceptrones multicapa, Poggio y Girosi (Poggio y Girosi, 1990) enriquecen aún más el sustrato teórico de este tipo de redes aplicando la teoría de regularización de Tikhonov. Una introducción histórica más profunda de las redes neuronales se puede encontrar por ejemplo en (Haykin, 1994, Cap. 1).

El perceptrón es quizás la forma más simple de una red neuronal que se puede utilizar para la clasificación de clases o conceptos que sean linealmente separables, es decir que las muestras positivas y negativas de la clase se pueden separar mediante un hiperplano en el espacio de características $\mathcal{X}$, en la Figura 1.10 se muestra un ejemplo para dimensión 2.

La estructura del perceptrón se puede ver en la Figura 1.12 y el funcionamiento del mismo consiste en la suma de las entradas ponderadas por un peso cada una de ellas, además existe un valor umbral representado en la Figura por w_0. Dependiendo si la suma ponderada de las entradas es mayor o menor que el umbral, la salida será +1 o -1. La salida del perceptrón se puede ver como la composición de dos funciones, por un lado se encuentra la suma ponderada que se denominará función de activación

$$v(\mathbf{X}) = \sum_{i=0}^{n} w_i x_i \tag{1.7}$$

y por otro lado está la función signo que es la función de salida del perceptrón. De esta forma se puede clasificar una muestra como perteneciente o no a una clase dependiendo

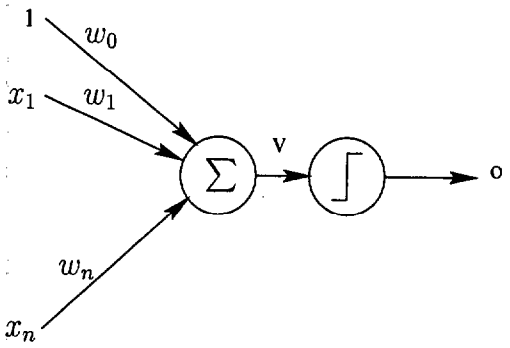


Figura 1.12: Perceptron de una sola capa

del valor de salida del perceptrón.

$$\varphi(v) = \begin{cases} +1 & \text{si } v \geq 0 \\ -1 & \text{si } v < 0 \end{cases} \tag{1.8}$$

El proceso de aprendizaje consiste en obtener los pesos de cada una de las entradas y el valor del peso umbral. Para ello se puede utilizar la *Regla de Entrenamiento Perceptrón* que se demuestra que converge a una solución en un número finito de pasos si las clases son linealmente separables. En el proceso de aprendizaje los pesos se consideran como elementos de un vector, vector de pesos, cuyos elementos se inicializan a valores aleatorios, que luego se modifican iterativamente según la siguiente regla:

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \eta \Delta \mathbf{w}_i \tag{1.9}$$

donde $\Delta \mathbf{w}_i$ es el incremento del vector de pesos en la iteración *i-ésima*. La expresión de este incremento según la regla de aprendizaje del perceptrón es:

$$\Delta \mathbf{w}_i = \begin{cases} 0 & \text{si } \mathbf{x} \text{ está bien clasificada} \\ \mathbf{x} & \text{si } \mathbf{x} \text{ está mal clasificada} \end{cases} \tag{1.10}$$

En la expresión anterior η es un factor de ponderación para ajustar la velocidad de cambio del vector de pesos, y es la salida esperada para la muestra actual y o es la salida del perceptrón con el vector de pesos actual, que será $+1$ ó -1. Analizando la ecuación (1.10) se puede observar que si la clasificación por parte del perceptrón de la muestra actual es correcta, no existe incremento en el vector de pesos mientras que si es incorrecta el incremento será de proporcional a $\mathbf{x}$, positivo o negativo dependiendo de los valores de y y o.

Para clases linealmente separables la regla (1.10) converge a una solución, sin

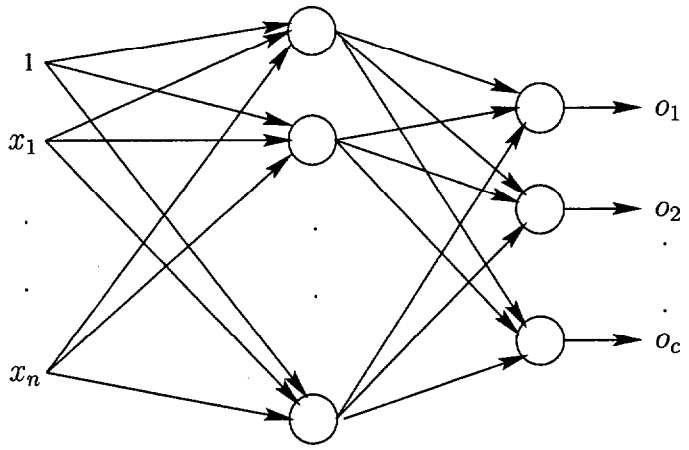


Figura 1.13: Perceptron multicapa

embargo, si las clases no son linealmente separables, es preciso una configuración de red por capas que se conoce como Perceptrón Multicapa y cuya estructura se puede ver en la Figura 1.13. La red consta de múltiples capas donde las entradas a las unidades de cada capa corresponden con las salidas de la capa anterior. Las unidades que se utilizan en esta red son iguales a las del perceptrón con una función de salida $\varphi(v)$ derivable, a diferencia de la función signo utilizada en el perceptrón. Una función muy utilizada es la función sigmoide.

$$\varphi(v) = \frac{1}{1 + e^{-\alpha v}} \tag{1.11}$$

siendo el parámetro α una constante positiva que controla la inclinación de la función sigmoide.

El algoritmo de aprendizaje más utilizado en este tipo de redes es la Retropropagación del Error o más conocido por el término inglés, *Backpropagation* (Rumelhart et al., 1986*b*). El backpropagation se basa en la *Regla Delta* o *Regla de Widrow-Hoff* (Widrow y Hoff, 1960) que utiliza como medida a optimizar la suma de los errores cuadráticos entre los valores esperados de las muestras y los resultantes del perceptrón multicapa, $E(\mathbf{w})$, que depende de los valores del vector de pesos actual del perceptrón.

$$E(\mathbf{w}) = \frac{1}{2}\sum(y - o)^2 \tag{1.12}$$

La expresión anterior se puede ver en el espacio de pesos como una hipersuperficie de forma que el punto para el que la función de error es mínima corresponde a la configuración de pesos óptima. Por tanto en cualquier punto de esa superficie el gradiente apuntará en la dirección de máxima pendiente, y modificando el vector de pesos con el

vector gradiente negado se tiende iterativamente a la configuración del mínimo de la función objetivo (1.12). Cuando se llega a un mínimo, el gradiente será nulo en ese punto por lo que el vector de pesos no se modificará.

$$\nabla E(\mathbf{w}) = \left\{ \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_n} \right\} \tag{1.13}$$

A diferencia del perceptrón, donde la modificación de los pesos dependen del error que se comete en la salida, en el perceptrón multicapa este error solo se puede calcular para las unidades de la capa de salida que es para las únicas que se conoce la salida correcta. El algoritmo de aprendizaje backpropagation se basa en la propagación del error que se comete en la capa de salida hacia las capas internas de la red. La expresión de la regla para modificar los pesos iterativamente se puede ver en (Haykin, 1994, pág. 142).

Los inconvenientes del descenso según el gradiente son bien conocidos, como la lentitud en algunos casos para alcanzar el mínimo o el problema de los mínimos locales que pueden atrapar al sistema en una configuración no óptima.

Diversos autores han estudiado la capacidad de los perceptrones multicapa para representar diversos tipos de funciones. Por ejemplo las funciones lógicas se pueden representar exactamente con un perceptrón multicapa de una capa oculta, aunque el número de unidades en la capa oculta crece exponencialmente con el número de entradas. Para funciones continuas acotadas Hornick (Hornick et al., 1989) demuestra que se pueden representar con un error tan pequeño como se desee mediante una red con una capa oculta y con unidades que utilizan funciones de salida tipo sigmoide, y funciones generales se pueden aproximar mediante redes con dos capas ocultas (Cybenko, 1989). En el caso de funciones $f : [0,1]^n \rightarrow R^m$, $f(\mathbf{x}) = \mathbf{y}$, el teorema de Kolmogorov (Hecht-Nielsen, 1989) establece que puede ser representada exactamente con una red de una capa oculta de $2m + 1$ unidades.

Otro tipo de redes neuronales son las basadas en funciones de base radial (RBFN). Estas redes tienen cierto parecido con el método de las funciones de potencial (Duda y Hart, 1973). La utilización de las funciones de base radial para la construcción de redes neuronales se debe a Broomhead (Broomhead y Lowe, 1988), aunque otros autores como Lee (Lee y Kil, 1988), Moody (Moody y Darken, 1989*a*), Musavi (Musavi et al., 1994*b*) o Poggio (Poggio y Girosi, 1990) han realizado contribuciones al diseño, teoría y aplicaciones de este tipo de redes.

La arquitectura de las redes basadas en funciones de base radial se muestra en la

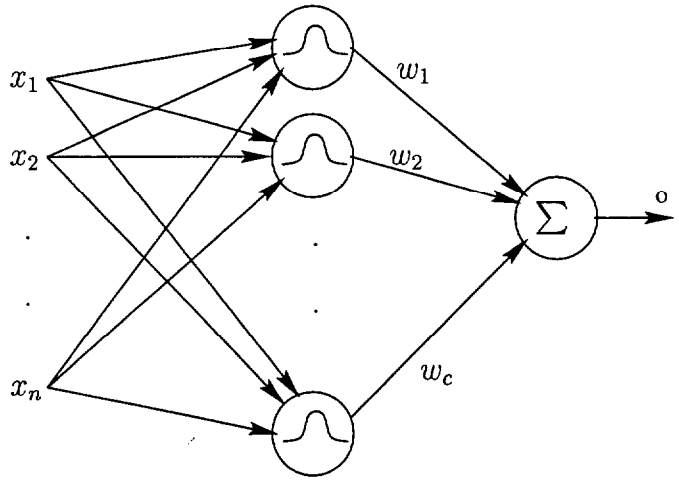


Figura 1.14: Red neuronal basada en funciones de base radial

Figura 1.14. Esta redes constan de tres capas, una de entrada, una capa oculta y la de salida. El número de unidades de la capa de entrada es igual a la dimensionalidad del problema, n. En la capa oculta existen tantas unidades como funciones de base radial (RBF) tenga la red, ya que cada una de estas unidades corresponde con una RBF. La interconexión entre la capa de entrada y la capa oculta es completa como puede verse en la Figura 1.14. La capa de salida contiene una unidad que corresponde con un perceptrón, utilizando como función de salida una de tipo sigmoide (1.11). Esto es así porque el proceso de clasificación utilizando este tipo de redes se puede justificar con el teorema de Cover (Cover, 1965) sobre separabilidad de patrones, que establece que un problema de clasificación es más probable que sea linealmente separable en un espacio de mayor dimensión que en el espacio original.

Una función de base radial bastante utilizada es la gaussiana multidimensional, $G(\mathbf{x}, \boldsymbol{\mu}, \Sigma)$,

$$G(\mathbf{x}, \boldsymbol{\mu}, \Sigma) = e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

donde $\boldsymbol{\mu}$ es el vector de medias y Σ es la matriz de covarianza. El vector de medias establece la localización en el espacio de características de la función, mientras que la matriz de covarianza define la forma del hiperelipsoide definido por los punto de igual valor de la gaussiana. Igual que la gaussiana se pueden utilizar otro tipo de funciones definidas por un punto que establece la posición de la función y para el cual el valor de la función sea máximo, y decrezca monótonamente hacia cero a medida que la distancia con el punto central aumenta.

Los algoritmos de aprendizaje para estas redes neuronales se basan en la obtención

de los parámetros que definen las RBF y los pesos de las conexiones entre la unidades de la capa oculta y la de salida. Lee (Lee y Kil, 1988) utiliza un algoritmo basado en el descenso según el gradiente, denominado *Hierarchically Self-Organization Learning*, que optimiza la suma de errores al cuadrado aprendiendo de forma supervisada tanto el número de unidades de la capa oculta, los parámetros que la definen así como los pesos con la capa de salida. El aprendizaje de unidades (RBF) en la capa oculta se realiza por un proceso de acomodación que comienza inicialmente con una sola unidad. A medida que se presentan muestras a la red si son cubiertas por algunas de las RBF existentes se realiza un proceso de ajuste de los parámetros de la red, si no es cubierta por una de las RBF existente se añade una nueva, que equivale a una nueva unidad de la capa oculta. El inconveniente de esta aproximación es que debido a la alta dimensionalidad del espacio de búsqueda existe mayor probabilidad de parar en un mínimo local además, de que el proceso de convergencia es más lento.

Musavi (Musavi et al., 1994*b*) realiza el aprendizaje de la red en dos fases. En una primera fase obtiene las RBF, gaussianas multidimensionales que van a formar parte de la red, mediante un proceso de agrupamiento supervisado que es básicamente una modificación del K-Medias (Duda y Hart, 1973). Como resultado de este primer proceso se obtienen las RBF y los parámetros de las mismas. Después mediante un proceso basado en el descenso según el gradiente, similar al utilizado para el perceptrón, obtiene los pesos de las conexiones entre la capa oculta y la de salida.

Desde el punto de vista del espacio de hipótesis, en las redes neuronales cada hipótesis puede considerarse como cada posible asignación de pesos y parámetros que definen las unidades de la red, es decir forma un espacio N-dimensional Euclídeo. Este espacio es continuo a diferencia de otros espacio de hipótesis vistos anteriormente como los árboles de decisión u otros algoritmos de aprendizaje simbólicos. El hecho de que el espacio sea continuo unido a que la función de error es diferenciable permite realizar un proceso de búsqueda bien definido en dicho espacio. El sesgo inductivo que subyace tanto en los perceptrones con funciones de salida sigmoidal como en las redes basadas en funciones de base radial, es el de encontrar la interpolación suave de los puntos del conjunto de aprendizaje, algo que se aprecia con facilidad en las superficies de decisión obtenidas en las RBFN utilizando funciones gaussianas.

Los dos tipos de redes neuronales explicados anteriormente son solamente una muestra de las diferentes arquitecturas y algoritmos de aprendizaje existentes en el campo de las redes neuronales. Una discusión en más profundidad de las redes neuronales artificiales se puede encontrar por ejemplo en Hecht-Nielsen (Hecht-Nielsen, 1989), Haykin (Haykin, 1994).

1.7.3 Algoritmos Genéticos

Los algoritmos genéticos se definieron como imitación de algunos procesos observados en la evolución natural. Aunque no se conoce con total certeza el mecanismo que guía la evolución en todos sus aspectos, la introducción en un algoritmo de ciertos elementos estudiados en la evolución podrían ayudar a resolver algunos problemas complejos. Holland (Holland, 1975) fue quien introdujo el concepto de algoritmo genético debido a que eran algoritmos que realizaban simulaciones de poblaciones de cromosomas que se codifican como cadenas de bits. Algunas características que incorporan los algoritmos genéticos y que comparten con la evolución natural son que la manipulación de los cromosomas se hace independientemente del problema tratado, solo se conoce la evaluación de cada cromosoma que se produce y es esta evaluación la que guía el proceso de selección de nuevos cromosomas. En (Goldberg, 1989) se puede encontrar una amplia descripción de los algoritmos genéticos y en (Davis, 1991) se muestra la aplicación de los algoritmos genéticos a diferentes problemas.

Los mecanismos que unen un algoritmo genético con el problema a resolver son la codificación de los cromosomas y una función de evaluación que devuelve una medida del grado de ajuste del cromosoma en la resolución del problema. La codificación varía de un problema a otro. Holland utilizaba cadenas de bits, codificación que es aún la más utilizada ya que como se verá es la más adecuada para las operaciones que realizan los algoritmos genéticos. Por ejemplo, un conjunto de clausulas del tipo *si-entonces* se puede representar eligiendo una codificación para las precondiciones de las reglas y otra para las conclusiones de las mismas como se muestra en (Holland, 1986; Grefenstette, 1988). Otro ejemplo de la utilización de las cadenas de bits en la codificación de problemas es su utilización para codificar redes neuronales (Harp y Samad, 1991). La función de evaluación o función de ajuste es el mecanismo que conecta el algoritmo genético con el problema a resolver. Esta función toma como entrada un cromosoma y devuelve un número que indica como es de buena la solución codificada en el cromosoma en la resolución del problema. Esta función permite al algoritmo genético seleccionar aquellos individuos o cromosomas mejor adaptados para que formen parte de una nueva generación. El esquema básico de un algoritmo genético se puede ver en el Algoritmo 1. El funcionamiento de los algoritmos genéticos consiste en iteraciones donde se van modificando el conjunto de cromosomas o individuos que forman parte de la población, que es un conjunto de individuos de tamaño fijo. Inicialmente se genera una población de forma aleatoria y se calcula la función de evaluación o ajuste para todos los individuos de esa primera población.

Algoritmo 1 Algoritmo genético básico

Inicializa la población: P ← Genera p individuos aleatoriamente.
Calcular función de evaluación para todos los $h \in P$ $Fitness(h)$.
while $\max_h\{Fitness(h)\} < fitness\_umbral$ **do**
Crear una nueva generación: P_S
Reproducción: Seleccionar aleatoriamente $(1-r)p$ individuos de P y añadirlos a P_S. La probabilidad $Pr(h_i)$ de seleccionar en individuo h_i es:

$$pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$

Cruce: Seleccionar aleatoriamente $\frac{rp}{2}$ pares de individuos de P de acuerdo a la probabilidad anterior. Para cada par $\langle h_1, h_2 \rangle$, producir dos descendientes aplicando un operador de cruce y añadirlos a al población P_S.
Mutación: Elegir el m por ciento de los miembros de la población con probabilidad uniforme. Para cada individuo seleccionar un bit aleatoriamente y cambiarlo.
Modificar la población actual: $P \leftarrow P_S$.
Calcular función de evaluación para todos los $h \in P$ $Fitness(h)$.
end while
Devolver el individuo de la población P con mayor valor de la función de ajuste.

Las siguiente generación se obtiene mediante un proceso de selección de los mejores individuos de la generación actual según la función de ajuste definida. Algunos de estos individuos se pasan a la siguiente generación sin ningún cambio mientras que otros se combinan según diferentes operadores de cruces. Al igual que ocurre en el proceso de selección natural en los individuos, se puede dar un proceso de mutación que permite explorar nuevas soluciones no incluidas en generaciones anteriores. Como se puede ver en el Algoritmo 1 el valor de la función de ajuste para un determinado individuo define la probabilidad de que se incluya en la siguiente generación. La función de ajuste a utilizar puede tener diferentes formas, entre ellas la denominada ruleta rusa o selección proporcional de ajuste, selección turno y selección por ordenación. La selección por ruleta rusa viene de la similitud de funcionamiento con este juego, ya que la probabilidad de seleccionar un individuo es proporcional al valor de la función de ajuste como se muestra en el esquema del algoritmo genético. El de ordenación se basa en ordenar por orden decreciente de valor de la función de ajuste y por tanto la probabilidad de que un individuo sea seleccionado viene dada por la posición que ocupa y no tanto por el valor de la función. A continuación se explican los tres operadores utilizados en el Algoritmo 1 para obtener los individuos de una determinada generación partiendo de los individuos mejor adaptados de la generación anterior.

El operador de cruce produce dos nuevos descendientes a partir de dos individuos

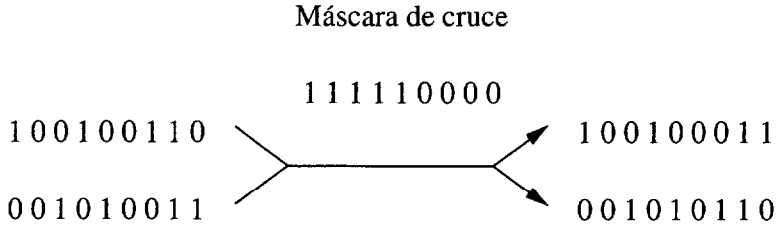


Figura 1.15: Operador de cruce de un punto

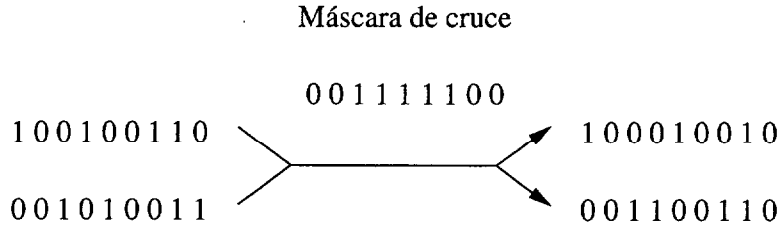


Figura 1.16: Operador de cruce de dos puntos

mediante la copia de parte del conjunto de bits que forman parte de cada uno de los individuos padres en cada uno de los descendientes. Así cada padre proporciona un conjunto de bits a cada uno de los descendientes, la elección de qué bits de cada padre contribuye a cada descendiente se obtiene mediante la denominada máscara de cruce que pueden ser de un solo punto, dos puntos o uniforme. En la Figura 1.15 se muestra la máscara de cruce y un ejemplo de los descendientes de dos individuos utilizando este operador de cruce de un solo punto de cruce, y en la Figura 1.16 se muestra el de dos puntos de cruce.

En la Figura 1.17 se puede ver el operador de cruce uniforme, en este caso para cada uno de los bits que componen los padres se asignan aleatoriamente de forma uniforme a uno y a otro descendiente.

Otro operador que se puede ver en el Algoritmo 1 es el de mutación. Este operador simula las mutaciones que se producen en la evolución natural y gracias a las cuales se consigue que las poblaciones progresen, ya que algunas de estas mutaciones producirán individuos mejor adaptados que sus progenitores y por tanto tienen mayor probabilidad

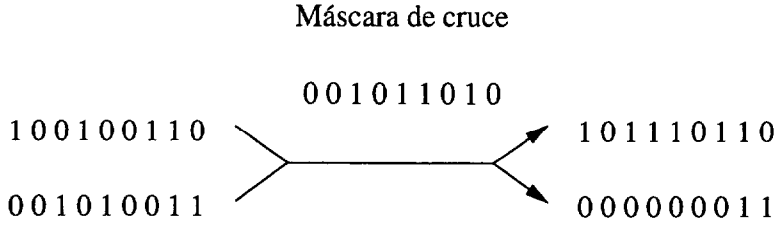


Figura 1.17: Operador de cruce uniforme

de perpetuarse en generaciones futuras. La simulación de este proceso se realiza en los algoritmos genéticos mediante el cambio de ciertos bits de algunos individuos en cada población. La probabilidad de estas mutaciones suele ser muy baja ya que de lo contrario no se conservarían características de los individuos mejor adaptados. En las implementaciones existen dos formas de realizar las mutaciones, una es complementando el bit seleccionado y la otra es una vez que se ha seleccionado el bit se le asigna el valor 0 ó 1 aleatoriamente y con la misma probabilidad.

Aparte de los operadores de cruce y de mutación existen otros operadores aunque no son muy utilizados, uno de ellos es el operador de inversión. Con este operador se seleccionan dos puntos en la cadena de bits del individuo y los bits entre estos dos puntos se reordenan en sentido inverso.

Un aspecto importante en el estudio de los algoritmos genéticos es la caracterización matemática de la evolución de la población a través de sucesivas generaciones. Esta caracterización se puede obtener a partir del Teorema de Esquemas de Holland (Holland, 1975) que se basa en el concepto de esquemas (schema). Un esquema se puede considerar como un bloque que va a formar parte del individuo mejor adaptado a la resolución del problema, y que recoge alguna característica que debe poseer dicho individuo. Estos esquemas al principio pueden no darse en un solo individuo sino estar repartidos entre todos los individuos que forman la población. El teorema propuesto por Holland demuestra que aquellos esquemas que codifican las características que deben tener los individuos mejor adaptados sobrevivirán a lo largo de diferentes generaciones mientras que otros esquemas que aporten menos a la solución de problema tenderán a desaparecer.

Desde el punto vista del espacio de hipótesis, los algoritmos genéticos no realizan un proceso de búsqueda con trayectoria ordenada como ocurre en otro métodos de aprendizaje, sino que se obtienen nuevas hipótesis a partir de la recombinación (operador de cruce) y mutación (operador de mutación) de partes de anteriores hipótesis. De esta forma en cada paso el conjunto de hipótesis en consideración (población) se modifica mediante la sustitución de parte de esa población por nuevos individuos más adaptados (según la función de evaluación) para la resolución del problema.

Una consecuencia de los algoritmos genéticos es la programación genética donde la población en lugar de estar formada por cadenas de bits está compuesta por programas de computador. Koza (Koza, 1992) presenta los conceptos básicos y algunas aplicaciones de la programación genética en la resolución de problemas. La utilización de las técnicas de los algoritmos genéticos en la programación genética parte del hecho de que en muchos problemas de Inteligencia Artificial se busca un programa que

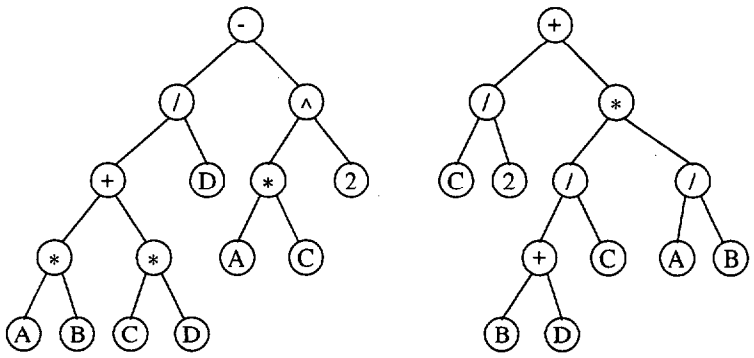


Figura 1.18: Dos programas ejemplos

permita resolver un determinado problema. Debido a que los algoritmos genéticos recorren el espacio de búsqueda, espacio de programas en este caso, obtienen los individuos (programas) mejor adaptados (más adecuados) en la resolución de un problema.

Los programas en programación genética se representan como árboles jerárquicos que equivalen al árbol de análisis del programa correspondiente (Fig. 1.18). En estos árboles cada nodo no hoja corresponde con una función que posee tantos nodos hijo como parámetros tiene la función. Los nodos hojas del árbol corresponden con elementos terminales que pueden ser variables o constantes. Las funciones que pueden componer un programa deben cumplir la propiedad de cierre que establece que todas las funciones deben aceptar como argumento cualquier valor o tipo de datos devuelto por cualquier otra función o cualquier elemento terminal. Por ejemplo si se utiliza la programación genética para obtener funciones lógicas, la propiedad de cierre está garantizada porque las funciones lógicas solo admiten como argumentos valores booleanos y las variables utilizadas también tomarán valores booleanos. Sin embargo en un programa general existen diferentes tipos de datos (booleanos, enteros, reales, etc.) y es necesario modificar los operadores proporcionados por el lenguaje para que la propiedad de cierre se cumpla. Otra propiedad que debe cumplir el conjunto de terminales y funciones, es la propiedad de suficiencia que establece que la solución de problema se pueda obtener con los conjuntos definidos.

El funcionamiento de la programación genética es similar a los algoritmos genéticos. Inicialmente existe una población de programas generada aleatoriamente, y luego en sucesivas iteraciones se seleccionan los mejores individuos según una función de ajuste para pasar a la siguiente generación, otros se cruzan y por últimos se puede mutar algún individuo de la población resultante. Al igual que en los algoritmos genéticos se pueden utilizar otro tipo de operadores. La generación de la población inicial se hace de forma aleatoria seleccionando primero una función para formar el nodo raíz del árbol

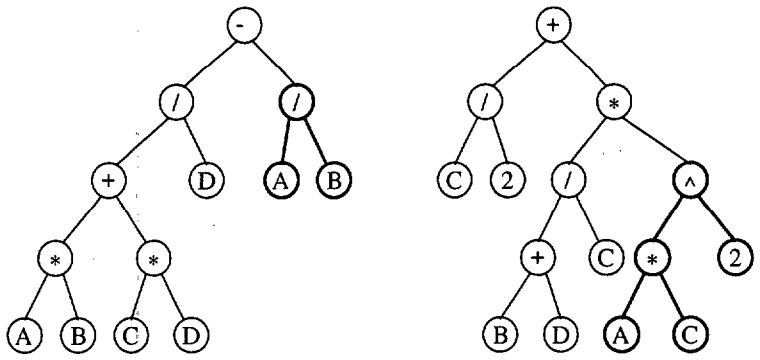


Figura 1.19: Ejemplo de operador cruce en los dos programas de la Figura 1.18

y a partir del mismo hacerlo crecer seleccionando aleatoriamente funciones y terminales para el resto de los nodos. La selección de las funciones y elementos terminales para los nodos interiores y hojas de árbol se puede hacer siguiendo diferentes estrategias que den lugar a árboles todos con el mismo número de nodos o con diferentes tamaños (Koza, 1992, sec. 6.2).

Una vez definida la población inicial se realizan sucesivas iteraciones donde se van seleccionando los individuos (programas) que mejor resuelven el problema. En este caso es necesario definir una función de evaluación o ajuste que mida la adecuación de cada individuo en la resolución del problema. La obtención de la medida se hace definiendo un conjunto de casos de prueba y ejecutando el programa (individuo) para cada uno de los casos y obteniendo por ejemplo la tasa de error en el conjunto. Por tanto los individuos (programas) que resuelvan un mayor número de casos estarán mejor adaptados y tendrán mayores probabilidades de ser seleccionados. El operador de cruce se realiza mediante la selección de partes de los programas padre e intercambiándolos para dar lugar a los programas hijos. En la Figura 1.19 se puede ver un ejemplo del operador cruce aplicado a los programas mostrados en la Figura 1.18, las partes del árbol que se han cruzado entre los dos programas aparecen con líneas más gruesas. La mutación puede producirse en los nodos hojas, intercambiando los elementos terminales o bien a nivel de los nodos interiores del árbol que corresponden con funciones, pero en este caso es necesario tener en cuenta los argumentos de la anterior función y la nueva ya que pueden no coincidir.

La programación genética se ha utilizado en la resolución de diferentes problemas. En el texto de Koza (Koza, 1992) se explican varias aplicaciones desde la simple ordenación de bloques para la obtención de la palabra "UNIVERSAL" hasta otras más complicadas como el control de un robot simulado. Koza (Koza et al., 1996) describe también la utilización de la programación genética en un problema tan complicado como

es la obtención de circuitos electrónicos que implementan un filtro, utilizando para ello el simulador de circuitos electrónicos SPICE por lo que cada programa consiste en la descripción de un circuito electrónico para dicho simulador.

En el sistema PADO (Teller y Veloso, 1994) se describe una arquitectura que aprende a asignar etiquetas de salida correctas a señales. Una aplicación de dicho sistema es el reconocimiento de objetos en escenas visuales, tarea bastante compleja. La población en PADO, a diferencia de la programación genética tradicional, no utiliza funciones sino que utiliza lo que definen como programas. A diferencia de las funciones que realizan una correspondencia entre entrada y salidas, los programas o algoritmos incluyen parámetros que además pueden ser pasados de forma iterativa o recursiva. Otro elemento que incluye PADO es que puede aprender a distinguir diferentes clases a la vez, por lo que en la población existen individuos que deben reconocer diferentes clases. Para ello la función de evaluación o ajuste tiene en cuenta la clase que reconocen porque el proceso de recombinación entre individuos se realiza entre aquellos que están mejor adaptados a reconocer un determinado tipo de clase. Para esto, todos los elementos de la población se dividen en conjuntos donde los elementos que lo componen poseen un alto valor de la función de ajuste para reconocer una determinada clase. Luego el operador de cruce y mutación solo se aplica a individuos del mismo grupo, por lo que ambos operadores no son los estándar si no unos que los autores denominan SMART. Una vez se han recombinando todos los elementos de los grupos se vuelven a unir y se vuelve a proceder a la evaluación de la función de ajuste y división en grupos. Los individuos (programas) en PADO están expresados en un lenguaje propio y consisten en grafos de nodos dirigidos que tienen una parte de acción y una de decisión de salto, ya que cada arco entre nodos indica un salto.

1.8 Evaluación del Aprendizaje mediante Estimación del Error

En la Sección 1.2 se indicaba que el objetivo del Aprendizaje Automático es el diseño de sistemas capaces de adaptarse de manera que realicen más eficientemente la tarea para la que están diseñados. Una clase de sistemas que cumplen con este objetivo son los clasificadores, por tanto se intenta obtener clasificadores que sean cada vez más eficientes en su tarea de asociar muestras no etiquetadas a muestras etiquetadas (pág. 4). La forma más común de medir la eficiencia de un clasificador es la tasa de error. Cada vez que se le presenta un nuevo caso, debe tomar una decisión sobre la clase correcta para ese caso.

El error real que cometerá el clasificador se define como la tasa de error (1.14) cuando el número de casos tiende asintóticamente a un número muy grande de nuevos casos y que converge en el límite a la distribución de la población de donde son tomados.

$$error_{\mathcal{D}} = \frac{número\ de\ errores}{número\ de\ muestras} = \frac{r}{N} \tag{1.14}$$

En la expresión anterior es necesario definir qué es un error. Un error es una clasificación incorrecta: al clasificador, que da como salida un valor discreto perteneciente al conjunto $\mathcal{Y}$, se le presenta una muestra y si el resultado no es el mismo que el esperado se considera un error. Si todos los errores tienen igual importancia entonces el error cometido el clasificador es el que se muestra en la ecuación (1.14).

Estadísticamente el error real se define como el límite de la ecuación (1.14) cuando el número de muestras es muy grande, o cuando el número de casos posibles es finito se puede obtener el error real con la evaluación de la ecuación (1.14) sobre todos los casos. Ambas definiciones no son operativas ya que la obtención de un número muy alto de casos no es abordable en la mayoría de los problemas, o bien porque si se tiene un número finito de casos y se dispone de todas las soluciones para comprobar la tasa de error, la inducción de un clasificador no tiene sentido.

La pregunta que surge es si se puede estimar la tasa de error real a partir del error empírico que se obtiene de la evaluación de la expresión (1.14) en el conjunto de aprendizaje del que se dispone para la inducción del clasificador, y que en muchos problemas consiste en un número pequeño de muestras. Para ello es necesario hacer uso de estimadores, que deben tener dos características: ser estimadores centrado y tener una varianza pequeña. En el diseño de clasificadores a partir de un conjunto de muestras, se busca construir un clasificador que tenga un error pequeño, por lo que es necesario estimar el error que tendrá el clasificador en nuevas muestras no incluidas en el conjunto de aprendizaje, pero que se distribuya de forma similar.

En general si se dispone de un conjunto de datos D del que se conoce un determinado parámetro y para todos los elementos dicho conjunto, y se quiere estimar el valor de esa función para el dominio global, se debe buscar un estimador, que es una variable aleatoria, que estime el parámetro de la población a partir de la muestra contenida en el conjunto de datos D. El estimador f_D debe optimizar la expresión (1.15) que es el error cuadrático promedio entre el valor dado por el estimador para una determinada entrada, f_D, y el valor esperado para la misma entrada $E(y|x)$.

$$E[(f_D(x) - E(y|x))^2] \tag{1.15}$$

Desarrollando la expresión anterior se tiene

$$E[(f_D(x) - E(y|x))^2] = [E_D[f_D(x)] - E(y|x)]^2 + E_D[(f_D(x) - E_D[f_D(y|x)])^2]$$

El primer término de la expresión anterior es el sesgo que corresponde con la distancia entre el valor promedio de $f_D(x)$ y la salida esperada cuando se tiene la entrada x; y el segundo término es la varianza del estimador f_D. Un buen estimador debe tener un sesgo nulo, lo que se conoce como estimador centrado o no sesgado ya que en este caso el valor del estimador coincide en promedio con el valor estimado. Pero algunos estimadores centrados pueden tener un valor alto de la expresión (1.15) debido a que posee una gran varianza. Por tanto para que el estimador tenga utilidad debe ser centrado y con una varianza baja. La expresión (1.14) es un estimador centrado del error ya que sigue una distribución Binomial y en esta distribución el valor esperado de r es Np donde p es el valor del error real. Por lo que si N es constante, el valor esperado de r/N es p.

Una forma de dar la incertidumbre asociada a una estimación es hacer uso de los intervalos de confianza. Estos intervalos dan los extremos entre los que se puede encontrar el valor estimado así como la probabilidad (nivel de confianza) de que se encuentre entre los mismos. Aunque como se comentó anteriormente que la proporción de error en el conjunto de aprendizaje sigue una distribución binomial, si el número de muestras es superior a 30, se puede aproximar por una normal y la expresión del intervalo de confianza viene dada por

$$error_{\mathcal{D}} \pm z_n \sqrt{\frac{error_{\mathcal{D}}(1 - error_{\mathcal{D}})}{N}} \tag{1.16}$$

donde z_n depende del valor del nivel confianza y corresponde al valor de la distribución normal que deja a cada lado un porcentaje igual a la mitad de 1 menos el nivel de confianza. La expresión anterior es válida si $N \geq 30$ y el valor de $error_{\mathcal{D}}$ no esta cerca de 0 ó 1, además de que el clasificador debe dar como salida un valor discreto.

A continuación se describirán distintas técnicas de estimación del error. A excepción de la estimación por medio del error aparente, el resto de las técnicas tratan de simular la situación que se produce cuando se lleva a cabo el proceso de inducción de un clasificador, es decir de una población se obtiene una muestra que se utiliza para realizar el proceso de aprendizaje y luego se supone extrapolable el rendimiento del clasificador inducido sobre el resto de la población no vista. Para ello se utiliza la técnica de submuestreo aleatorio que se ilustra en la Figura 1.20, donde la población posee una

distribución desconocida F, de esta población se extrae una muestra, que conforma el conjunto de aprendizaje, con una distribución F' que debe ser bastante próxima a la de la población si se han obtenido las muestras de forma aleatoria y en número suficiente. Para simular este proceso descrito sobre el conjunto de aprendizaje disponible, éste se divide en diferentes particiones. El clasificador se induce a partir de un número de estas particiones y su eficiencia se comprueba en aquellas no utilizadas en el proceso de inducción. Una revisión de estas diversas técnicas se pueden encontrar en (Weiss y Kulikowski, 1991; McLachlan, 1992; Liu y Motoda, 1998*b*).

1.8.1 Error Aparente

La estimación del error con este tipo de estimador se obtiene como la proporción de muestras del conjunto de aprendizaje que han sido erróneamente clasificadas.

$$error_{app} = \frac{1}{N} \sum_{(\mathbf{X}^{(j)}, Y^{(j)}) \in \mathcal{D}} \delta(\mathcal{I}(\mathcal{D}, \mathbf{X}^{(j)}), Y^{(j)}) \tag{1.17}$$

Donde $\delta(\mathcal{I}(\mathcal{D}, \mathbf{X}^{(j)}), Y^{(j)}) = 1$ si $\mathcal{I}(\mathcal{D}, \mathbf{X}^{(j)}) \neq Y^{(j}$ y 0 en otro caso. Como se ve en la ecuación (1.17) el clasificador se entrena y se comprueba su tasa de error utilizando el mismo conjunto de aprendizaje, por lo que la estimación es optimista ya que la muestras a clasificar ya han sido vistas previamente por el clasificador. Este efecto en ciertos clasificadores con una alta capacidad de memorización como puede ser un clasificador basado en el vecino más cercano, lleva a considerar una tasa de error bastante baja, resultando la misma mayor en muestras no vistas anteriormente por el clasificador en el proceso de aprendizaje.

Esta subestimación del error que produce este método se debe al sesgo favorable que introduce en su estimación. Para ciertas distribuciones de clases y atributos existen expresiones que dan una medida de dicho sesgo. Para el caso concreto de dos clases distribuidas según una normal, el sesgo que introduce este estimador es conocido y se obtiene a partir de la distancia de Mahalanobis (McLachlan, 1992), por lo que sumado al error aparente da como resultado el error real.

El uso de este estimador tiende a generar clasificadores que poseen un mal comportamiento cuando se presentan nuevos casos, es decir tendrán poca capacidad de generalización. Por supuesto, siempre que no exista un gran número de muestras en cuyo caso la tasa de error estimada coincidirá con la tasa de error real. A este efecto de poca capacidad de predicción se le denomina sobreespecialización o sobreajuste.

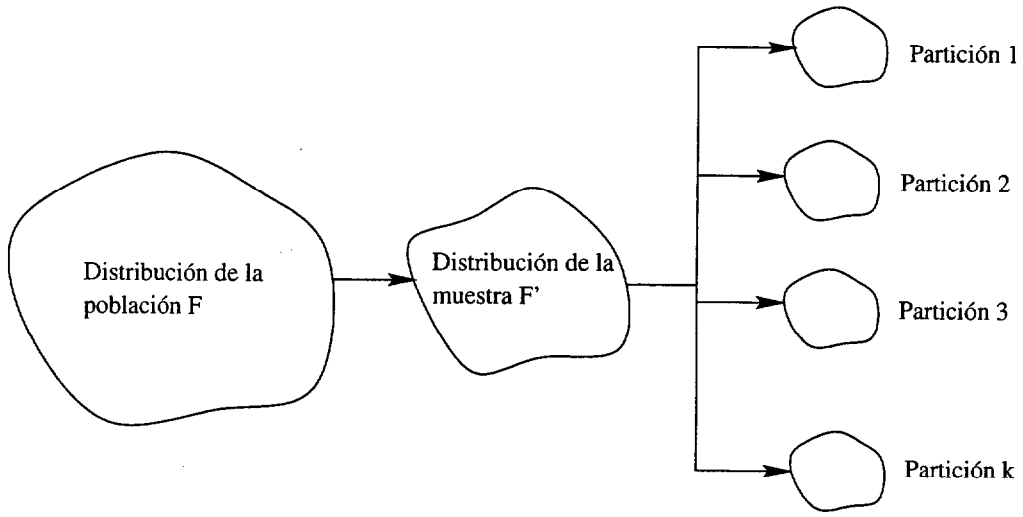


Figura 1.20: Esquema de muestreo aleatorio de una población.

El caso extremo de sobreajuste mediante la minimización de la tasa de error aparente se presenta en la construcción de una tabla índice (*look-up table*) con las muestras del conjunto de aprendizaje, donde el error aparente es nulo mientras que la capacidad de generalización es mínima.

1.8.2 Entrenamiento y Prueba o *Holdout*

El problema de la sobreespecialización que aparece con la utilización del error aparente se debe a que el mismo conjunto de muestras se utiliza en el proceso de inducción del clasificador y de validación del mismo, que no se corresponde con las condiciones de funcionamiento reales ya que el clasificador inducido deberá clasificar muestras que no se le han presentado previamente en el proceso de inducción. Y el rendimiento en esas nuevas muestras puede diferir bastante del que existía en el proceso de aprendizaje.

Para evitar la sobreespecialización del clasificador se utiliza la técnica de remuestreo aleatorio (*random resampling*) del conjunto de aprendizaje, que consiste en estimar la tasa de error como la tasa de error promedio de los clasificadores derivados de las particiones de prueba aleatoria e independiente generadas. Estas técnicas se denominan de remuestreo ya que intentan reproducir el proceso de inducción y funcionamiento (proceso de clasificación) de un clasificador. Como se puede ver en la Figura 1.20, los elementos del dominio $\mathcal{X}$ se encuentran distribuidos según una probabilidad F y de éstos se obtiene un conjunto de muestras $\mathcal{D}$ que se encuentran distribuidas según la probabilidad F', que se corresponde con F si el muestreo fue aleatorio. Si este conjunto de aprendizaje se vuelve a muestrear aleatoriamente es como si se considerara el conjunto

de aprendizaje el dominio del problema y a los subconjuntos resultante el conjunto de aprendizaje.

Un tipo de remuestreo aleatorio es la estimación por entrenamiento y prueba o *hold-out*, donde se utiliza un conjunto de muestras para obtener el clasificador y otro diferente para estimar el error, con el fin de eliminar el efecto de la sobreespecialización. Este método consiste en dividir el conjunto de aprendizaje $\mathcal{D}$ en dos: $\mathcal{D}_t$ con $N-h$ muestras y $\mathcal{D}_h$ con h muestras. El primero, $\mathcal{D}_t$, es el utilizado para llevar a cabo el entrenamiento del clasificador, el resto de las muestras no utilizadas, $\mathcal{D}_h$, se utilizan para estimar la tasa de error de clasificador, es decir:

$$error_h = \frac{1}{h} \sum_{(\mathbf{X}^{(j)}, Y^{(j)}) \in \mathcal{D}_h} \delta(\mathcal{I}(\mathcal{D}_t, \mathbf{X}^{(j)}), Y^{(j)}) \tag{1.18}$$

Una cuestión importante con el uso de este estimador es cuántas muestras se deben utilizar en el conjunto de prueba $\mathcal{D}_h$ para que la tasa de error estimada sea lo más próxima a la tasa de error real. En principio, el número de muestras necesarias para el conjunto de prueba no tiene que ser elevado, además es posible conocer cual es la diferencia de la tasa de error estimada con la real en función de este número. Una división que se utiliza con mucha frecuencia es tomar 2/3 de las muestras para el proceso de aprendizaje y el 1/3 restante para comprobar el error del clasificador.

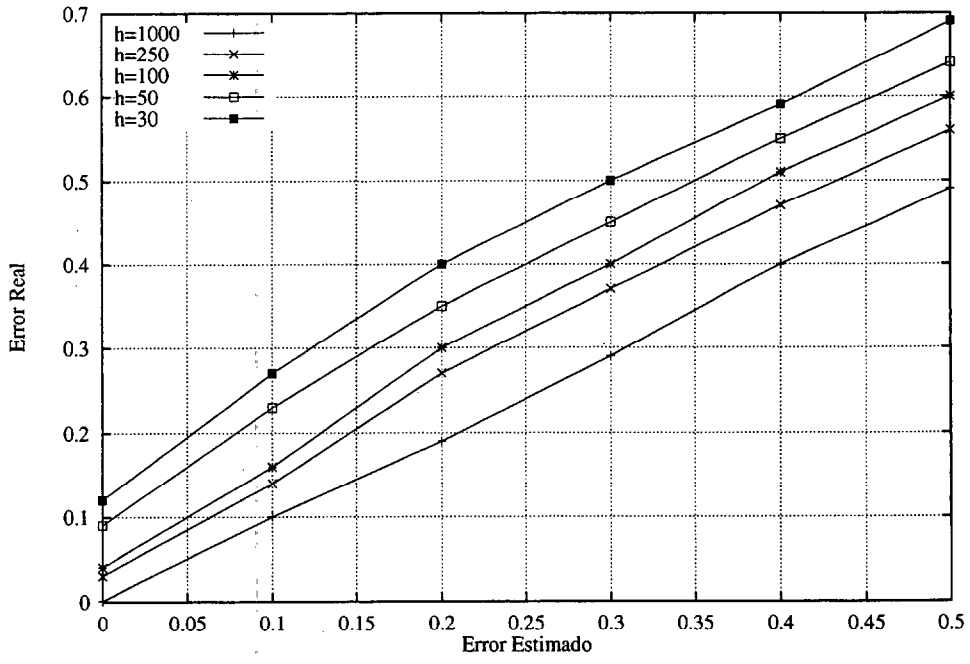


Figura 1.21: Diferencia entre el error estimado y el real con la técnica *holdout* en función de número de muestras.

En la Figura 1.21 se muestra la relación que existe entre el error estimado y la probabilidad más alta del error real para varios tamaños del conjunto de prueba, con un intervalo de confianza del 95%. Por ejemplo, para un conjunto de test de 50 muestras ($h = 50$) y un error estimado del 0% la probabilidad de que el error real este por debajo del 10% es del 95%, sin embargo para $h = 1000$ el error real estará por debajo del 1% con una probabilidad del 95%. La gráfica mostrada en la Figura 1.21 se obtiene a partir de que el error estimado mediante esta técnica sigue una distribución de probabilidad binomial independientemente de la distribución que posean las muestras (Weiss y Kulikowski, 1991).

Un inconveniente que se detecta en este estimador es que solo se utiliza una parte de las muestras disponibles para llevar a cabo el aprendizaje, con lo que se pierde información útil en el proceso de inducción del clasificador. Por ello esta técnica para estimar el error es pesimista.

1.8.3 Validación Cruzada

Para evitar la ocultación de parte de las muestras al algoritmo de inducción y la consiguiente perdida de información, se plantea otra técnica dentro del remuestreo aleatorio que se denomina validación cruzada (*cross-validation*). Con esta técnica el conjunto de aprendizaje se divide en k particiones mutuamente exclusivas: $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$ conteniendo todas aproximadamente el mismo número de muestras. En muchos casos se suele indicar este caso como validación cruzada con k particiones (*k-fold cross validation*). A partir de las k particiones se obtienen k clasificadores, utilizando como conjunto de aprendizaje para el clasificador *i-ésimo* todas las particiones menos la partición *i-ésima* y el error se estima sobre las muestras de la partición no utilizada en el aprendizaje. Por último el error se obtiene como la media de los errores de los k clasificadores.

$$error_{cv} = \frac{1}{k} \sum_{i=1}^{k} \left(\frac{1}{|\mathcal{D}_i|} \sum_{\langle \mathbf{X}^{(j)}, Y^{(j)} \rangle \in \mathcal{D}_i} \delta(\mathcal{I}(\mathcal{D} \setminus \mathcal{D}_i, \mathbf{X}^{(j)}), Y^{(j)}) \right) \tag{1.19}$$

$|\mathcal{D}_i|$ representa la cardinalidad del conjunto $\mathcal{D}_i$ y $\mathcal{D} \setminus \mathcal{D}_i$ es todo el conjunto de aprendizaje menos el conjunto $\mathcal{D}_i$. La validación cruzada es completa cuando se toma el promedio de todos los posibles conjuntos de $N - k$ muestras sobre las N muestras del conjunto de aprendizaje, lo que es computacionalmente costoso. La forma de calcular la tasa de error según la expresión (1.19) no es muy adecuada cuando el número de muestras es muy pequeño, pudiéndose optar en tal caso por la que se propone en (Kohavi, 1995*b*) y que

definiendo $\mathcal{D}_{(i)}$ al conjunto que contiene la muestra i-ésima, tiene la siguiente expresión:

$$error_{cv} = \frac{1}{r} \sum_{\langle \mathbf{X}^{(j)}, Y^{(j)} \rangle \in \mathcal{D}} \delta(\mathcal{I}(\mathcal{D} \setminus \mathcal{D}_{(i)}, \mathbf{X}^{(j)}), Y^{(j)}) \tag{1.20}$$

La utilización de este estimador supone la elección de k, el número de particiones. Breiman (Breiman et al., 1993) encuentra que para el CART los mejores resultados se obtienen con un valor de $k = 10$, a la misma conclusión llega Kohavi en su tesis (Kohavi, 1995*b*). En general este es un número de particiones aceptado en el uso de este estimador. Breiman indica que una mejora en la utilización de la validación cruzada es la estratificación que consiste en mantener en cada una de las particiones una distribución de las etiquetas similar a la existente en el conjunto de aprendizaje, para evitar una alta varianza en la estimación.

Un caso particular de este estimador es la validación cruzada dejando uno fuera (LOOCV) donde $k = N$, es decir, existirán tantas particiones como muestras tiene el conjunto de aprendizaje. En este caso el clasificador se entrena con todas la muestras menos una y se utiliza para clasificar la muestra no utilizada en el aprendizaje. El inconveniente del LOOCV es el alto coste computacional que supone el aprendizaje del clasificador N veces, por lo que no se suele utilizar cuando el número de muestras es elevado o el proceso de inducción del clasificador es computacionalmente costoso. No obstante, existen algunos trabajos que indican que para ciertos clasificadores como discriminantes cuadráticos o el vecino más cercano es posible la utilización del LOOCV de forma muy sencilla (McLachlan, 1992).

La ventaja de esta técnica de estimación es que todos los casos son utilizados para la estimación y casi todos los casos se utilizan en el aprendizaje, dando lugar a un estimador con un sesgo muy pequeño. El inconveniente que posee es que la varianza es muy alta, lo cual en muchos casos hace poco útil su utilización a pesar del pequeño sesgo. Una forma de reducir la varianza consiste en ejecutar varias veces la validación cruzada y obtener como estimación la media de todas. Con esta solución un principio que se rompe es la independencia de las muestras utilizadas en las diferentes ejecuciones (Kohavi, 1995*b*), ya que el conjunto de muestras es el mismo. Debido a que se rompe la independencia entre los distintos conjuntos de datos de las diferentes ejecuciones de la validación cruzada, el intervalo de confianza para la estimación no se puede obtener a partir de la ecuación (1.16).

1.8.4 Bootstrap

Las técnicas de estimación basadas en este concepto fueron introducidas por Efron (Efron, 1979) encontrándose explicadas en más detalle en (Efron y Tibshirani, 1993). Estas técnicas se proponen para reducir las alta variabilidad que exhibe la validación cruzada en muestras pequeñas (menos de 30 muestras), consiguiendo un aumento de eficiencia comparable a un aumento en el tamaño de la muestra de un 60% (Efron y Tibshirani, 1995).

Dentro de esta tipología existen diversos tipos de estimadores aunque entre los más utilizados se encuentra el $e0$ y el .632. En ambos casos se toman r muestras del conjunto de aprendizaje con reemplazamiento, que conformarán el conjunto de aprendizaje. Todas aquellas muestras que no formen parte del conjunto de aprendizaje se tomarán como conjunto de prueba. Al ser tomadas la muestras con reemplazamiento, se sigue una distribución binomial con lo que la probabilidad de que una muestra no sea elegida después de N intentos es $(1 - 1/N)^N \approx e^{-1} \approx 0.368$. Por tanto el número esperado de muestras diferentes que pertenezcan al conjunto de prueba será de $0.368N$ y al conjunto de aprendizaje $0.632N$.

La creación del conjunto de prueba y aprendizaje según el esquema se repite b veces, dando lugar a b conjuntos de prueba y de aprendizaje. La elección del número b no es crítico cuando es mayor que 100, aunque Efron (Efron, 1983) sugiere que no sea mayor de 200. Igual que en la validación cruzada, se obtienen b clasificadores a partir de los conjuntos de aprendizaje y para cada uno un error estimado sobre el conjunto de validación correspondiente.

El error estimado según el estimador $e0$ se obtiene como la media del error de los b clasificadores, mientras que en el estimador .632 la estimación del error se define como,

$$error_{.632} = \frac{1}{b}\sum_{i=1}^{b}(0.632 \cdot e0_i + 0.368 \cdot error_{app}) \tag{1.21}$$

donde $error_{app}$ es el error aparente (1.17) del conjunto de aprendizaje completo. El estimador .632 tiene el inconveniente de comportarse de forma pesimista cuando se utiliza con clasificadores que se ajustan perfectamente a las muestras del conjunto de aprendizaje, como puede ser el vecino más cercano, ya que en este caso el término $error_{app} \approx 0$ y solo se tiene en cuenta el término $e0$ que es prácticamente la estimación *holdout* afectada por un factor menor que 1. Para evitar este inconveniente Efron y Tibshirani (Efron y Tibshirani, 1995) proponen otro estimador perteneciente a la familia

de los *bootstrap* denominado 623+ y que asigna un mayor peso a $error_{app}$ cuando el clasificador se ajusta bien a los datos, es decir, $e0 - error_{app}$ es grande.

En (Jain et al., 1987) se comparan las técnicas de *bootstrap* $e0$ y .632 con la validación cruzada dejando uno fuera (LOOCV) para el clasificador del vecino más cercano (1-NN y 3-NN) y el clasificador cuadrático, utilizando para ello conjuntos de aprendizaje sintéticos con un error conocido. La conclusión es que el estimador $e0$ se comporta mejor para el 1-NN a nivel de sesgo y varianza, mientras que para el 3-NN el mejor resultado se obtiene con .632 lo que concuerda con los resultados de Efron (Efron, 1983).

Otro trabajo donde se comparan distintos métodos de estimación para el clasificador del vecino más cercano es el propuesto por Weiss (Weiss, 1991). En él se comparan los clasificadores 1-NN y el 3-NN usando conjuntos de aprendizaje sintéticos y con los estimadores LOOCV, $e0$, .632 y la validación cruzada con 2 particiones (2-cv). Los conjuntos de aprendizaje se toman con dos tamaños 20 y 60 muestras encontrando que para 20 muestras la varianza de LOOCV es bastante alta aunque no posee prácticamente sesgo. Comparando LOOCV con $e0$ y 2-cv, el mejor comportamiento se obtiene con el estimador 2-cv estratificado. Con respecto al .632 da buenos resultados cuando el error real no es muy alto. Por último se propone una modificación del LOOCV, que denomina Lv-1*, que consiste en tomar la estimación dada por .632 o por 2-cv en los extremos. Un inconveniente de este estimador es su no linealidad, que lo hace poco adecuado para la comparación de modelos ya que una pequeña diferencia en el LOOCV puede dar lugar a una gran diferencia en Lv-1*. Los anteriores trabajos han basado las comparativas en problemas sintéticos donde la tasa de error real es conocida.

En (Kohavi, 1995*a*) se estudia el comportamiento de la estimación basada en validación cruzada y el bootstrap .632 utilizando para ello conjuntos de aprendizaje de problemas reales y utilizando como clasificadores el C4.5 y el clasificador bayesiano simplificado. El error real se obtiene mediante 500 ejecuciones de la estimación *Holdout* y se compara con los resultados de los otros estimadores. Con respecto al sesgo, la validación cruzada es un estimador pesimista cuando se utiliza con pocas particiones (2 ó 5), cuando el número de particiones aumenta el sesgo disminuye estabilizándose prácticamente a partir de 10 particiones. Si se añade estratificación, el comportamiento mejora. En cuanto a la varianza, ésta es alta con la validación cruzada con pocas particiones y baja con el .632, sin embargo este último posee un sesgo bastante acentuado en algunos casos. Como conclusión se puede apreciar que la validación cruzada con 10 particiones y estratificación posee un buen compromiso entre sesgo y varianza. Este equilibrio es el factor más importante cuando se desean comparar diferentes modelos,

más que obtener un sesgo mínimo a costa de una alta varianza, ya que el sesgo va a afectar por igual a todos los modelos comparados. Este esquema, validación cruzada con 10 particiones y estratificación, va a ser el utilizado en el capítulo dedicado a la parte experimental de esta tesis.

Un tema bastante relacionado con la estimación del error es la comparación del rendimiento de dos métodos de clasificación o algoritmos de inducción. En este caso si el rendimiento se considera el error, entonces se comparan los errores cometidos por los métodos en evaluación utilizando test de hipótesis. Sobre este tema se volverá en el Capítulo 5.

Capítulo 2

Conceptos en Teoría de la Información

La propuesta para la selección de atributos que se presenta en esta tesis está basada en conceptos de Teoría de la Información (Cover y Thomas, 1991; Abramson, 1986; Blahut, 1991; Ash, 1965) entre otras referencias generales, por tanto es importante dar una introducción de algunos conceptos básicos de esta teoría.

La Teoría de la Información surge en los años 40 a partir de un articulo de Claude E. Shannon (Shannon, 1948) titulado "A Mathematical Theory of Communication". Como se deduce del título, la principal aportación consistía en modelar la comunicación más que la información como tal. Así, los conceptos que se presentaban en dicho articulo eran los de máxima tasa de compresión y máxima capacidad de un canal, que establecían que mientras la tasa de transmisión se mantenga por debajo de la capacidad del canal la probabilidad de error no aumenta, contradiciendo la opinión predominante en los años 40 que establecía que el aumento de velocidad de transmisión sobre un medio de comunicación aumentaba la probabilidad de error. Debido a que muchos procesos, incluido la selección de atributos, se pueden modelar como un proceso de información, la Teoría de la Información se ha utilizado en diferentes campos.

2.1 Entropía y Entropía Condicional

Aunque los conceptos de Teoría de la Información fueron definidos inicialmente para fuentes y canales de información, y símbolos pertenecientes a un determinado alfabeto, en este capítulo se presentarán estos conceptos desde el punto de vista de variables aleatorias, ya que una fuente de símbolos se puede modelar como una variable aleatoria

discreta con un dominio que se corresponde con el alfabeto. El concepto principal de la Teoría de la Información y a partir del cual se derivan los restantes es la *entropía.* Este concepto es una medida de la incertidumbre promedio acerca de los valores que puede tomar una determinada variable aleatoria discreta. A continuación se verá una definición formal de la entropía.

Definición 2.1 (Entropía). *Sea una variable aleatoria discreta A definida en el dominio $\{a_1, a_2, \ldots, a_s\}$ y con distribución de probabilidad $P(A = a_i)$ $i = 1 \ldots s$, se define la entropía $H(A)$ como,*

$$H(A) = -\sum_{i=1}^{s} P(a_i) \log P(a_i) \tag{2.1}$$

La unidad de medida de la entropía depende de la base del logaritmo. Así si es en base 2 se expresa en bits y si el logaritmo es natural se expresa en nats. A lo largo de este documento se va a considerar el logaritmo en base 2 por lo que no se va poner la base del mismo. La interpretación de la entropía es la de cota inferior del número de bits (0 ó 1 lógicos) necesarios para representar la variable aleatoria o lo que es lo mismo el número mínimo de preguntas verdadero/falso para conocer el valor de la variable aleatoria. Además como se observa en (2.1) la entropía no depende de los valores que toma la variable aleatoria sino de su distribución de probabilidad. Algunas propiedades de la entropía son:

i. Es una medida no negativa: $H(A) \geq 0$

ii. Tiene cota superior: $H(A) \leq \log s$. La igualdad se alcanza cuando todos los símbolos son equiprobables, es decir, $P(a_i) = 1/s$; $\forall i = 1 \ldots s$.

La entropía entendida como medida de la incertidumbre sobre los valores que puede tomar una variable aleatoria, se puede ver claramente en la segunda propiedad, ya que cuando todos los valores son equiprobables y por tanto la incertidumbre es máxima sobre el valor que puede tomar, la entropía es máxima e igual a $\log s$; sin embargo cuando la distribución de probabilidad es tal que $P(A = a_i) = 1$ y $P(A = a_j) = 0$ $\forall j \neq i$ la entropía es cero, ya que no existe ninguna incertidumbre sobre el valor que puede tomar la variable, siempre será a_i.

Si en lugar de una sola variable aleatoria, existe otra variable B definida en el conjunto $\{b_1, b_2, \ldots, b_r\}$ y con distribución de probabilidad $P(B = b_i)$ $i = 1 \ldots r$, y

relacionada con la variable A por la matriz de probabilidades condicionales M,

$$M \doteq \begin{bmatrix} P(b_1|a_1) & P(b_1|a_2) & \dots & P(b_1|a_s) \\ P(b_2|a_1) & P(b_2|a_2) & \dots & P(b_2|a_s) \\ \vdots & \vdots & \vdots & \vdots \\ P(b_r|a_1) & P(b_r|a_2) & \dots & P(b_r|a_s) \end{bmatrix} \tag{2.2}$$

se puede definir otro concepto que es la entropía conjunta. La matriz M en Teoría de la Información se corresponde con la matriz que define un canal de información, donde cada elemento es la probabilidad de recibir el símbolo b_j cuando ha sido enviado a_i. A partir de la matriz M y de la distribución de probabilidades $P(B)$ se puede obtener la distribución de probabilidades conjunta $P(A, B)$ y a partir de esta última definir la entropía conjunta como,

Definición 2.2 (Entropía Conjunta). *Sea $P(A, B)$ la distribución de probabilidad conjunta de las variables aleatorias discretas A y B, se define la entropía conjunta $H(A, B)$ como,*

$$H(A, B) = -\sum_{i=1}^{s}\sum_{j=1}^{r} P(a_i, b_j) \log P(a_i, b_j) \tag{2.3}$$

También se puede definir la entropía condicional de una de las variables aleatorias conocida la otra, como el promedio de la entropía de B cuando se conoce el valor a_i.

Definición 2.3 (Entropía Condicional). *Sea M la matriz de probabilidades condicionales y $P(A, B)$ la distribución de probabilidad conjunta de las variables aleatorias discretas A y B, se define la entropía condicional $H(B|A)$ como,*

$$\begin{aligned} H(B|A) &= \sum_{i=1}^{s} P(a_i) H(B|a_i) \\ &= -\sum_{i=1}^{s} P(a_i) \sum_{j=1}^{r} P(b_j|a_i) \log P(b_j|a_i) \\ &= -\sum_{i=1}^{s}\sum_{j=1}^{r} P(a_i, b_j) \log P(b_j|a_i) \end{aligned} \tag{2.4}$$

De la ecuación (2.4) se deduce que $0 \leq H(B|A) \leq H(B)$, es decir, el conocimiento de la segunda variable aleatoria nunca incrementará la incertidumbre sobre la primera. El conocimiento de la variable aleatoria A eliminará totalmente la incertidumbre existente acerca de la variable B, $H(B|A) = 0$, si la matriz (2.2) solo posee un valor

diferente de 0 e igual a 1 en cada fila (en Teoría de la Información este tipo de matriz corresponde a un canal determinista) mientras que $H(B|A) = H(B)$ cuando todos los elementos de la matriz (2.2) son iguales a $1/s$.

2.2 Entropía Relativa e Información Mutua

El concepto de entropía visto en la sección anterior hace referencia a la incertidumbre de una variable aleatoria, es decir la cantidad de información necesaria para poder describirla. A continuación se definirá el concepto de entropía relativa, que es una medida sobre la similaridad entre dos distribuciones de probabilidad.

Definición 2.4 (Entropía Relativa). *Dadas dos distribuciones de probabilidad P y Q, se define la entropía relativa, entropía cruzada o distancia de Kullback-Leibler $D(P\|Q)$ como,*

$$D(P\|Q) = \sum_{a_i} P(a_i) \log \frac{P(a_i)}{Q(a_i)} \tag{2.5}$$

La entropía relativa es una medida de similaridad entre dos distribuciones de probabilidad y aunque se le denomina distancia de Kullback-Leibler, no es realmente una distancia ya que no es conmutativa ni cumple la propiedad de la desigualdad triangular. A continuación se definirá la información mutua que corresponde a un caso particular de entropía relativa que sí es conmutativa, aunque como se demostrará se comporta más como una medida de similaridad que como una medida de distancia.

Definición 2.5 (Información Mutua). *Sean dos variables aleatorias A y B con distribuciones de probabilidad $P(A)$ y $P(B)$ y distribución de probabilidad conjunta $P(A, B)$. La información mutua entre ambas variables aleatorias $I(A; B)$ se define como la entropía relativa entre la probabilidad conjunta y el producto de probabilidades.*

$$I(A; B) = \sum_{a_i} \sum_{b_j} P(a_i, b_j) \log \frac{P(a_i, b_j)}{P(a_i)P(b_j)} \tag{2.6}$$

Reescribiendo la expresión (2.6) para obtener la relación con la entropía de las

variables aleatorias se tiene que,

$$\begin{aligned} I(A;B) &= \sum_{a_i,b_j} P(a_i,b_j) \log \frac{P(a_i,b_j)}{p(a_i)P(b_j)} \\ &= \sum_{a_i,b_j} P(a_i,b_j) \log \frac{P(a_i|b_j)}{P(a_i)} \\ &= -\sum_{a_i,b_j} P(a_i,b_j) \log P(a_i) + \sum_{a_i,b_j} P(a_i,b_j) \log P(a_i|b_j) \\ &= -\sum_{a_i} P(a_i) \log P(a_i) - \left(-\sum_{a_i,b_j} P(a_i,b_j) \log P(a_i|b_j) \right) \end{aligned}$$

es decir,

$$I(A;B) = H(A) - H(A|B) \tag{2.7}$$

La ecuación (2.7) da otra interpretación de la información mutua y más utilizada que la Definición (2.5).

Según la Definición (2.3) de entropía condicional, ésta nunca a va ser mayor que la entropía de la variable A, por lo que la información mutua siempre va a ser mayor que o igual que cero. Si la entropía es una medida de incertidumbre y el conocimiento de la variable B va a reducir esa incertidumbre, entonces ese conocimiento es información que aporta sobre la variable A. Por tanto la información mutua según se definió en (2.7) es una medida de la cantidad de información que aporta la variable B sobre la variable A. Esta interpretación va a ser la que se va a manejar en la exposición del trabajo que se presenta en esta tesis.

Una característica de la información mutua es que el argumento del logaritmo en (2.6) es adimensional a diferencia de la entropía. Por tanto el valor de la información mutua es invariante frente a transformaciones invertibles y diferenciales de las variables (Battiti, 1994). A continuación se enuncian algunas propiedades de la información mutua.

i. Es una medida no negativa y acotada: $0 \leq I(A;B) \leq H(A)$

ii. Es conmutativa: $I(A;B) = I(B;A)$

iii. $I(A;A) = H(A)$ por eso se suele denominar a la entropía de una variable como "auto-información".

iv. $H(A,B) = H(A) + H(B) - I(A;B)$

La propiedad i de la información mutua indica la relación de esta medida con la dependencia entre las variables aleatorias. Cuando la información mutua es nula significa que la entropía y entropía condicional son iguales, es decir que la matriz M (2.2) que recoge la dependencia estadística entre las variables tiene todos sus elementos iguales y por tanto no existe dependencia entre las variables. Sin embargo cuando la información mutua es máxima es porque la entropía condicional es nula que ocurre cuando la matriz M es determinista, indicando que el conocimiento del valor de la variable B identifica unívocamente el valor de la variable A, o lo que es lo mismo la variable B contiene toda la información sobre la variable A.

Según las propiedades de la información mutua vistas anteriormente, esta medida es una función de similaridad (Spath, 1980), cumpliendo las siguientes propiedades:

i. $I(A;B) \leq H(A)$.

ii. Se cumple que $I(A;B) = H(A)$ si $P_A = P_B$

iii. $I(A;B) = I(B;A)$

Demostración. Las demostraciones de las anteriores son triviales a partir de las propiedades de la información mutua.

□

Un concepto muy relacionado con la información mutua es el de *distancia basada en entropía* (MacKay, 1997),

Definición 2.6 (Distancia basada en Entropía (MacKay, 1997)). *Sean A y B dos variables aleatorias con información mutua $I(A;B)$ y entropía conjunta $H(A,B)$. Se define la distancia basada en entropía entre A y B, $d(A,B)$, como,*

$$d(A,B) = H(A,B) - I(A;B) \tag{2.8}$$

La distancia basada en entropía es una medida, al igual que la información mutua, de la cantidad de información que aporta la variable aleatoria B sobre A, aunque el comportamiento es contrario al visto anteriormente para la información mutua. En este caso cuanta más información aporta la variable B menor es el valor de la distancia basada en entropía, como se demuestra en el siguiente teorema.

Teorema 2.1. *Sean A y B dos variables aleatorias discretas. La distancia basada en entropía $d(A,B)$ cumple $0 \leq d(A,B) \leq H(A) + H(B)$ siendo menor cuanto mayor información aporta B sobre A.*

Demostración. Si B no aporta ninguna información sobre A, entonces no existe ninguna relación entre ambas variables por lo que son estadísticamente independientes, $P(a,b) = P(a)P(b)$ siendo la información mutua entre ambas variables nula $I(A;B) = 0$.

Calculando la entropía conjunta,

$$\begin{aligned} H(A,B) &= -\sum_a \sum_b P(a,b) \log P(a,b) \\ &= -\sum_a \sum_b P(a)P(b) \log P(a)P(b) \\ &= -\sum_a \sum_b P(a)P(b) \log P(a) - \sum_a \sum_b P(a)P(b) \log P(b) \\ &= H(A) + H(B) \end{aligned}$$

Por lo que la distancia basada en entropía será máxima e igual a la suma de las entropías de las variables, $d(A,B) = H(A) + H(B)$.

Si B aporta toda la información sobre A entonces existe una dependencia estadística total entre ambas variables aleatorias, es decir $P(a,b) = P(a) = P(b)$, por tanto la información mutua es máxima, $I(A;B) = H(A) = H(B)$. En cuanto a la entropía conjunta $H(A,B)$,

$$\begin{aligned} H(A,B) &= -\sum_a \sum_b P(a,b) \log P(a,b) \\ &= -\sum_a \sum_b P(a) \log P(a) \\ &= H(A) \end{aligned}$$

Al ser la entropía conjunta y la información mutua iguales, la distancia basada en entropía es nula, $d(A,B) = 0$. □

A continuación se enumeran las propiedades que cumple la distancia basada en entropía y que se corresponden con las que se establecen para un funcional de distancia métrica (Anderberg, 1973).

i. Es siempre no negativa: $d(A,B) \geq 0$

Demostración.

$$d(A,B) = \sum_a \sum_b P(a,b) \log \frac{1}{P(a,b)} - \sum_a \sum_b P(a,b) \log \frac{P(a,b)}{P(a)P(b)}$$

$$
\begin{aligned}
&= \sum_a \sum_b P(a,b) \log \left(\frac{P(a)P(b)}{P(a,b)P(a,b)} \right) \\
&= \sum_a \sum_b P(a,b) \log \left(\frac{1}{P(a|b)} \frac{1}{P(b|a)} \right) \\
&= \sum_a \sum_b P(a,b) \log \frac{1}{P(a|b)} + \sum_a \sum_b P(a,b) \log \frac{1}{P(b|a)} \\
&= H(A|B) + H(B|A) \geq 0
\end{aligned}
$$

□

ii. Tiene un valor mínimo cuando las dos variables aleatorias son la misma: $d(A,A) = 0$

Demostración.

$$d(A,A) = H(A,A) - [H(A) - H(A|A)] = H(A) - H(A) = 0$$

□

iii. Es conmutativa: $d(A,B) = d(B,A)$

Demostración.

$$d(A,B) = H(A,B) - I(A,B) = H(B,A) - I(B,A) = d(B,A)$$

□

iv. Si $d(A,B) = 0$ entonces la distribuciones de probabilidad de las dos variables son iguales $P_A = P_B$

Demostración.

$$
\begin{aligned}
d(A,B) = 0 \Rightarrow H(A,B) &= I(A,B) \\
H(A) + H(B|A) &= H(A) - H(A|B) \\
H(B|A) &= -H(A|B)
\end{aligned}
$$

Como la entropía es no negativa se debe cumplir que:

$$H(B|A) = H(A|B) = 0 \Rightarrow P_A = P_B$$

□

v. Desigualdad triangular: $\forall A, B, C:\ d(A,C) \leq d(A,B) + d(B,C)$

Demostración. Reescribiendo la expresión (ec. 2.8) de la distancia $d(A,B)$ como,

$$d(A,B) = H(A|B) + H(B|A)$$

se tiene que,

$$\begin{aligned} H(A|B) + H(B|C) &\geq H(A|B,C) + H(B|C) \\ &= H(A,B|C) \\ &= H(A|C) + H(B|A,C) \\ &\geq H(A|C) \end{aligned}$$

Intercambiando en la expresión anterior A y C, se obtiene

$$H(C|B) + H(B|A) \geq H(C|A)$$

Y sumando las anteriores desigualdades,

$$H(A|B) + H(B|C) + H(C|B) + H(B|A) \geq H(A|C) + H(C|)$$

es decir,

$$d(A,B) + d(B,C) \geq d(A,C)$$

□

Una forma de visualizar la relación entre los conceptos de entropía, entropía conjunta, información mutua y distancia basada en entropía es mediante la ayuda de los diagramas mostrados en la Figura 2.1.

De la misma forma que se ha definido la información mutua para dos variables aleatorias, se puede definir la información mutua condicional entre A y B cuando se conoce C como,

Definición 2.7 (Información Mutua Condicional). *Sean tres variables aleatorias A, B y C con entropías condicionales $H(A|C)$ y $H(A|B,C)$, se define la información mutua condicional $I(A;B|C)$ como,*

$$I(A;B|C) = H(A|C) - H(A|B,C) \tag{2.9}$$

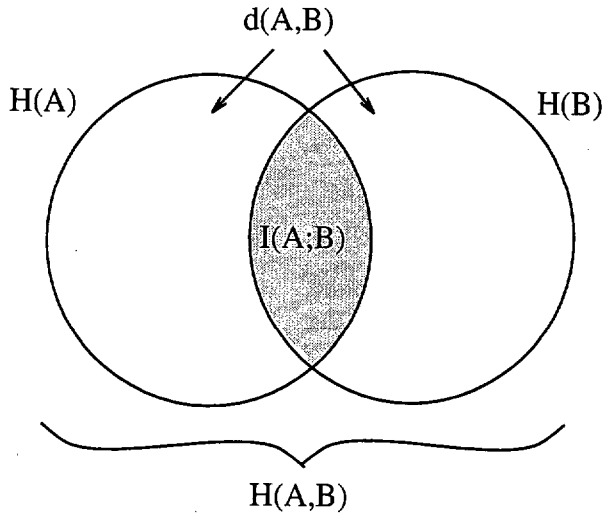


Figura 2.1: Representación gráfica de las relaciones entre entropías de dos conjuntos

La expresión de la información mutua condicional en función de las distribuciones de probabilidad es:

$$I(A;B|C) = \sum_i \sum_j \sum_k P(a_i, b_j, c_k) \log \frac{P(a_i, b_j|c_k)}{P(a_i|c_k)P(b_j|c_k)}$$

2.3 Entropía Diferencial

Los conceptos de Teoría de la Información vistos hasta ahora hacen referencia a variables aleatorias discretas y distribuciones de probabilidad, sin embargo los mismos conceptos se pueden definir para variables aleatorias continuas con funciones de densidad asociadas.

Definición 2.8 (Entropía Diferencial). *Sea A una variable aleatoria continua con función de densidad $f(a)$, se define entropía diferencial $h(A)$ como*

$$h(A) = -\int_S f(a) \log f(a) da \tag{2.10}$$

donde $S = \{a : f(a) > 0\}$ es el conjunto soporte donde está definida la función de densidad. En algunos problemas reales, la función de densidad no se conoce por lo que se debe estimar mediante la discretización del conjunto de soporte y calculándola como si de una variable aleatoria discreta se tratara.

Si se discretiza el rango de la variable continua A en intervalos de tamaño Δ siendo la función de densidad $f(a)$ continua en cada intervalo, se tiene que en cada

intervalo existe un valor a a_i que verifica:

$$f(a_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f(a)da \tag{2.11}$$

Sea la variable aleatoria discreta A^Δ definida a partir de la variable aleatoria continua A como,

$$A^\Delta = a_i, \text{ si } i\Delta \leq A \leq (i+1)\Delta \tag{2.12}$$

donde la distribución de probabilidad para la variable discreta $P_i \equiv P(A^\Delta = a_i)$ es

$$P_i = \int_{i\Delta}^{(i+1)\Delta} f(a)da = f(a_i)\Delta \tag{2.13}$$

Calculando la entropía de la variable aleatoria discreta A^Δ según la expresión (2.1) se tiene que:

$$\begin{aligned} H(A^\Delta) &= -\sum P_i \log P_i \\ &= -\sum f(a_i)\Delta \log\left(f(a_i)\Delta\right) \\ &= -\sum \Delta f(a_i) \log f(a_i) - \log \Delta \end{aligned} \tag{2.14}$$

La ecuación (2.14) muestra la entropía de una variable aleatoria obtenida a partir de la discretización de una variable continua, la cual se aproxima a la entropía diferencial a medida que el intervalo Δ tiende a cero, como se muestra en el teorema que se incluye a continuación y cuya demostración se puede encontrar en (Cover y Thomas, 1991, pág. 229).

Teorema 2.2 (Cover y Thomas, 1991). *Si la función de densidad $f(A)$ de la variable aleatoria continua A es integrable, entonces se verifica*

$$H(A^\Delta) + \log \Delta \to h(f) = h(A), \text{ a medida que } \Delta \to 0 \tag{2.15}$$

Por tanto la entropía de una discretización en 2^n intervalos de una variable aleatoria continua es aproximadamente $h(A) + n$.

Al igual que se definió la entropía diferencial para variables aleatorias continuas, se puede definir la entropía conjunta y la entropía condicional diferencial.

Definición 2.9 (Entropía Conjunta Diferencial). *Sean A y B dos variables aleato-*

rias continuas con función de densidad conjunta $f(A,B)$, se define la entropía conjunta diferencial $h(A,B)$ como

$$h(A,B) = -\iint f(a,b)\log f(a,b)dadb \tag{2.16}$$

Definición 2.10 (Entropía Condicional Diferencial). *Sean A y B dos variables aleatorias continuas con función de densidad conjunta $f(A,B)$ y función de densidad condicional $f(A/B$, se define la entropía condicional diferencial $h(A|B)$ como*

$$h(A|B) = -\iint f(a,b)\log f(a|b)dadb \tag{2.17}$$

2.4 Entropía Relativa e Información Mutua Diferencial

En esta sección se dan las definiciones equivalentes de la entropía relativa y la información mutua para variables aleatorias continuas, es decir, la entropía relativa e información mutua diferencial.

Definición 2.11 (Entropía Relativa Diferencial). *Sean dos funciones de densidad f y g, se define la entropía relativa o cruzada diferencial $D(f\|g)$ como*

$$D(f\|g) = \iint f(a)\log\frac{f(a)}{g(a)}da \tag{2.18}$$

Como se vio en la Sección 2.2, un caso particular de la entropía relativa es la información mutua, que para variables aleatorias continuas es

Definición 2.12 (Información mutua diferencial). *Sean dos variables aleatorias continuas A y B con funciones de densidad $f(A)$ y $f(B)$ y función de densidad conjunta $f(A,B)$. La información mutua entre ambas variables aleatorias $I(A;B)$ se define como la entropía relativa entre la probabilidad conjunta y el producto de probabilidades.*

$$I(A;B) = \int f(a,b)\log\frac{f(a,b)}{f(a)f(b)}dadb \tag{2.19}$$

Las propiedades que posee esta definición de la información mutua diferencial son las mismas que las de la información mutua enunciadas en la Sección 2.2.

La información mutua como se ha descrito anteriormente es una medida de de-

pendencia entre variables, como también lo son las funciones de correlación, por esta razón ha sido utilizada en algunos casos en lugar de éstas (Fraser y Swinney, 1986; Moddemeijer, 1989; Linsker, 1989; Bridle et al., 1992; Milosavljevic, 1995; Deco et al., 1995; Viola, 1995). A diferencia de las funciones de correlación, la información mutua es más general ya que recoge dependencias más complejas que las lineales, medidas por las funciones de correlación.

Para algunas funciones de probabilidad existen trabajos que establecen la relación de la información mutua con la correlación lineal. Así para variables aleatorias con función de densidad conjunta gaussiana existe una relación directa entre la información mutua y la función de correlación (Fraser, 1989). Para secuencias binarias que se encuentren desplazadas d posiciones, Li (Li, 1990) da la siguiente expresión que establece la relación entre la información mutua $I(d)$ y la función de correlación $\Gamma(d)$.

$$I(d) = \Gamma(d)\log\frac{[1+\Gamma(d)/P_1^2][1+\Gamma(d)/P_0^2]}{[1-\Gamma(d)/P_0P_1]^2} + P_1^2\log\left(1+\frac{\Gamma(d)}{P_1^2}\right) + P_0^2\log\left(1+\frac{\Gamma(d)}{P_0^2}\right) + P_0P_1\log\left(1-\frac{\Gamma(d)}{P_0P_1}\right)$$

En el cálculo de la entropía diferencial (2.10) y la información mutua diferencial (2.19) se supone conocida la función de densidad de la variable aleatoria, pero en muchos problemas reales no se dispone de esta información sino que se tiene un conjunto de muestras obtenidas mediante muestreo aleatorio. A partir de estas muestras se puede estimar la información mutua mediante métodos basados en núcleos como los propuestos en (Moon et al., 1995; Viola et al., 1995) o aproximando la función de densidad mediante histogramas (Fraser y Swinney, 1986; Moddemeijer, 1989; Moddemeijer, 1999).

La estimación por medio de núcleos consiste básicamente en estimar la función de densidad mediante funciones centradas en las muestras de forma que la información mutua se calcula según (2.19) ya que la función que se obtiene es continua. El inconveniente con estos métodos es que se deben fijar algunos parámetros de las funciones utilizadas en la estimación. Además el sesgo que introduce estos parámetros en la estimación de la información mutua no está estudiado.

En la estimación por histogramas el principal parámetro es el ancho de las celdas utilizadas en la discretización. En el trabajo de Fraser (Fraser y Swinney, 1986) se presenta un algoritmo recursivo donde el espacio se discretiza en celdas hasta que la distribución de las muestras dentro de cada celda sea plana, resultando más aproximada la aproximación (2.11). Para comprobar cuando la distribución es plana dentro de una celda los autores realizan un test de hipótesis χ^2 con un nivel de significación del 20%.

El sesgo que introduce la discretización del espacio en celdas ha sido estudiado por Moddemeijer (Moddemeijer, 1989) y se debe a los siguientes factores,

- $R - bias$, debido a la estimación de la función de densidad mediante histogramas la estimación va a ser peor a medida que el tamaño de las celdas aumenta.
- $N - bias$, debido al tamaño finito de la muestra la información mutua va a estar subestimada, siendo este sesgo mayor a medida que el número de muestras por celda disminuye.

Capítulo 3

Revisión Bibliográfica en Selección de Atributos

En este capítulo se introduce el problema de la selección de atributos, en el que se intenta obtener los atributos que mejor definen un concepto o clase en función de algún criterio de relevancia. Por tanto, algunas definiciones de relevancia de atributos existentes en la literatura son analizadas dependiendo de su capacidad para captar la relevancia de los atributos en distintos problemas. A continuación se realiza un recorrido por algunos de los trabajos que aparecen en la literatura de Aprendizaje Automático que describen algoritmos y métodos para la selección de atributos.

3.1 Introducción

Las dos grandes tareas del Aprendizaje Supervisado dentro del Aprendizaje Automático son la obtención de las reglas de clasificación (mecanismos de clasificación en general) y la selección de los atributos relevantes. Diferentes mecanismos de clasificación y métodos para su obtención se expusieron en el Capítulo 1, y en este capítulo se introducirá el problema de la selección de atributos, realizando un breve recorrido por trabajos existentes en la literatura sobre este problema. Antes de ver distintas definiciones de relevancia de atributos se expone necesidad de detectar estos atributos relevantes.

La primera razón por la que se considera necesario detectar los atributos relevantes, se basa en la preferencia por las hipótesis o modelos más sencillos frente a los más complejos. Esta preferencia ha sido utilizada con bastante frecuencia en la ciencia moderna y tiene sus orígenes en el denominado Principio de la Cuchilla de Occam (*Occam's Razor*) que originalmente recalcaba el principio Aristotélico "Entia non sunt multiplican-

da preater necessitatem" ("las entidades no debe multiplicarse más allá de lo necesario") formulándolo como "pluralitas non est ponenda sine necessitas" ("la pluralidad no debe utilizarse sin necesidad"). Este principio se ha utilizado amplia y clásicamente en el campo del Aprendizaje Automático (Blumer et al., 1987) y más recientemente en campos como Minería de Datos y Descubrimiento del Conocimiento (KDD) por Domingos (Domingos, 1999). En este último trabajo, Domingos indica que se pueden hacer dos interpretaciones del principio de la Cuchilla de Occam. Por un lado la preferencia de las hipótesis más simples por una cuestión de simplicidad de la hipótesis resultante, y por otra porque las hipótesis más simples tienen mayor probabilidad de generalizar mejor, es decir, menor error de generalización. En dicho trabajo, Domingos expone que la segunda interpretación no puede tomarse como algo general ya que si bien existen trabajos teóricos y experimentales que avalan dicha interpretación, también existen otros trabajos que demuestran su no validez.

La primera interpretación es la que, trasladada al problema de selección de atributos, ha sido utilizada como principio en esta tesis, y que se puede enunciar como,

> *Si con dos conjuntos de atributos muestran una calidad comparable en un proceso de inducción, es preferible tomar aquel conjunto que posea menor número de atributos.*

En la interpretación anterior del Principio de la Cuchilla de Occam se establece la selección de la hipótesis más simple (menor número de atributos), sin que ello suponga ninguna relación con el error del clasificador obtenido, ya que se parte de la igualdad en la información aportada por todos los conjuntos de atributos y por tanto no se realiza ninguna consideración previa en el error del clasificador. Es decir, lo que guía la selección de la hipótesis con menor número de atributos es que supone reducción del tiempo computo o en las capacidades de almacenamiento según el clasificador utilizado, así como una mayor simplicidad en la explicación del concepto inducido. Aunque esto último es bastante dependiente del contexto, por lo que no siempre la hipótesis más sencilla es más comprensible.

El otro motivo, aparte de la simplicidad de la hipótesis generada, viene dado porque la adición de nuevos atributos a un clasificador no supone un incremento en la tasa de acierto (Hamamoto et al., 1996), a excepción del clasificador bayesiano óptimo, para el cual la adición de nuevos atributos en el proceso de clasificación supone que la tasa de acierto aumente o se mantenga, nunca que disminuya. Sin embargo, como se expuso en el anterior capítulo, el clasificador bayesiano óptimo es de poca utilidad, ya

que el número de muestras necesarias para la estimación de las funciones de probabilidad crece exponencialmente con el número de atributos.

La aproximación Naive Bayes no se ve afectada tampoco por los atributos irrelevantes, pero sin embargo la tasa de acierto disminuye cuando existen atributos que son redundantes. En otros clasificadores prácticos la adición de nuevos atributos puede suponer que la tasa de acierto disminuya. Breiman (Breiman et al., 1993) encontró que la adición de nuevos atributos en la generación de los árboles de decisión no implicaba un incremento en la tasa de acierto, y sí un aumento del número de nodos. Holte (Holte, 1993) comprobó la tasa de acierto que se consigue con una sola regla en muchos de los problemas de clasificación utilizados para comprobar la eficiencia de los distintos métodos propuestos, y detectó que no es significativamente inferior a la que se consigue con reglas obtenidas con un algoritmo más elaborado como puede ser el C4.5, con un mayor número de atributos. Esto puede dar lugar a la discusión sobre la calidad de los conjuntos de datos utilizados como validación en Aprendizaje Automático, en lo referente a su relación con problemas reales, ya que de ser así los problemas reales a los que se debe enfrentar el Aprendizaje Automático son tan simples, que prácticamente se podrían resolver con una sola regla. Pero no se va a entrar en este tema ya que no es un objetivo de esta tesis entrar en esta discusión.

El problema de la selección de atributos no es algo exclusivo de Aprendizaje Automático ya que hace muchos años que se estudia en áreas como el Reconocimiento de Formas. Una recopilación de métodos de selección de atributos en este área se puede encontrar en (Devijver y Kittler, 1982). Una de las razones por la que los trabajos desarrollados en Reconocimiento de Formas no han tenido una aplicación inmediata en Aprendizaje Automático es la naturaleza, en general numérica, del dominio de los atributos tratados sin un alto contenido simbólico. Dentro del Reconocimiento de Formas las técnicas de selección de atributos se pueden dividir en dos categorías: extracción de atributos y selección de atributos.

La extracción propiamente dicha de atributos (Fig. 3.1) recoge los métodos que realizan una transformación sobre el espacio de atributos inicial, de forma que el espacio transformado resulte más adecuado en algún sentido para tareas de análisis y/o clasificación que el original. En esta categoría se pueden encontrar el Análisis de Componentes Principales (Sánchez García, 1978) y la transformada de Karhunen-Loeve (Kittler, 1986) o el Discriminante Lineal de Fisher (Duda y Hart, 1973).

En (Intrator, 1992) se utiliza una red neuronal para realizar la proyección del espacio original en uno de menor dimensión donde se intenta que la proyección sea multimodal, lo que facilita el proceso de clasificación. Torkkola y Campbell (Torkkola

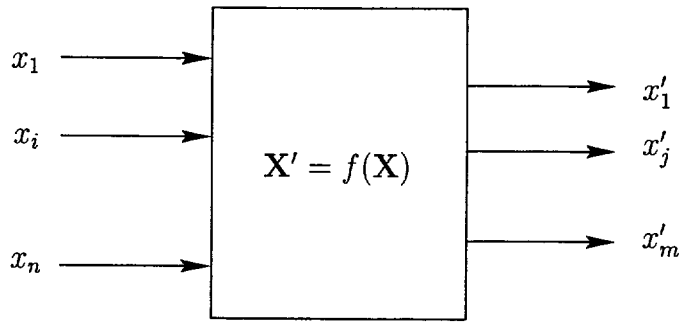


Figura 3.1: Extracción de atributos

y Campbell, 2000) proponen otra técnica diferente basada en Teoría de la Información para realizar la proyección del espacio original en uno de menor dimensión utilizando una transformación lineal que maximiza la información mutua entre las clases y las características en el subespacio proyectado. Debido a la transformación que se realiza, los atributos resultantes pierden su significado original, hecho no deseable en la mayoría de los problemas de Aprendizaje Automático. Además, debido a que la transformación se tiene que realizar cada vez que se precisa clasificar una nuevo caso, el tiempo de respuesta del clasificador se ve incrementado.

La selección de atributos en Reconocimiento de Formas, por otro lado no realiza ningún tipo de transformación por lo que el contenido semántico de cada atributo se mantiene. La selección se realiza mediante la búsqueda de los atributos que mejor definen la clase siguiendo algún tipo de medida (Fig. 3.2). Este esquema es el mismo que se ha seguido en Aprendizaje Automático, aunque las medidas que se utilizan para estimar cuales son los mejores atributos difieren en ambos campos. En Reconocimiento de Formas la selección de atributos se ha basado principalmente en medidas estadísticas como pueden ser la divergencia, la distancia de Matsusita o medidas de distancia interclase (Kittler, 1986; Jensen, 1986; Richards, 1986; Bow, 1992). El inconveniente de utilizar estas medidas en la selección de atributos en Aprendizaje Automático es que se basan en suponer una distribución específica de la clases en el espacio de atributos, como puede ser la normal, hecho que no siempre se puede asegurar en los problemas de este campo, cuando el número de muestras es reducido o de atributos es alto.

3.2 Definiciones de Relevancia

Un concepto nuclear en esta tesis y que por tanto precisa ser explícitamente tratado es la relevancia de atributos. Este concepto es necesario definirlo de forma que se puedan identificar y seleccionar qué atributos son relevantes. En (Wang, 1996) se recogen

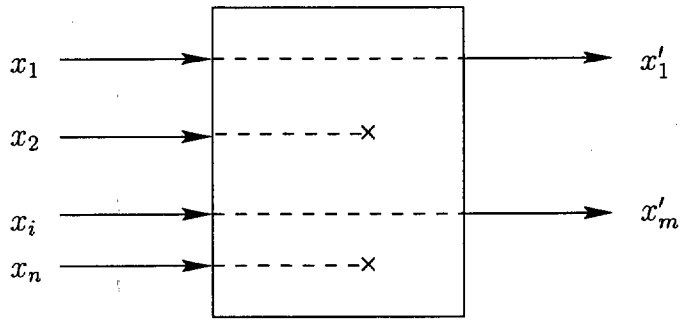


Figura 3.2: Selección de atributos

algunas formalizaciones y definiciones de relevancia debidas a diferentes autores, comenzando por una de sentido común que entronca claramente con el método de la variación concominante comentado en la Sección 1.6.

Definición 3.1. *Dadas a una evidencia existente E, una hipótesis H en consideración y la variación en la probabilidad de H debida al examen de la evidencia adicional I. Se dice que I es relevante para H en base a la evidencia E, si la probabilidad de H cambia con la adición de la nueva evidencia I a la ya existente. Si no es así se dice que es irrelevante.*

En (Wang, 1996) se recogen otras definiciones de relevancia desde un punto de vista formal en diferentes campos, aunque el interés en el presente trabajo se centra en las definiciones de relevancia en Aprendizaje Automático. En este punto nos encontramos con que en la bibliografía de este campo no existe un consenso sobre una definición única de relevancia. Ello es lógico debido a que como indica Blum (Blum y Langley, 1997) las definiciones de relevancia deben tener en cuenta la pregunta: "¿relevante para qué?". En este sentido Michalski (Michalski, 1983) clasifica los atributos que se utilizan en el proceso de aprendizaje, dependiendo de las tareas que debe realizar el proceso de aprendizaje partiendo de aquellos, en las siguientes categorías,

Relevancia Completa En el caso que los descriptores (atributos) se suponen directamente relevantes al problema, con lo que la tarea de aprendizaje consiste en formular una aserción inductiva.

Relevancia Parcial Cuando los casos observados pueden contener un gran número de descriptores irrelevantes o redundantes, con lo que la tarea de aprendizaje debe seleccionar los descriptores más relevantes y a partir de ellos construir la aserción apropiada.

Relevancia Indirecta Este es el caso más complejo, ya que si bien los descriptores no son relevantes, entre ellos algunos pueden ser utilizados para construir descriptores

derivados que sí sean relevantes. Por tanto, la tarea de aprendizaje tiene como objeto construir los nuevos descriptores, y a partir de ellos construir la aserción apropiada.

Algo que llama la atención de las anteriores definiciones es que para definir relevancia se utiliza el término relevante en el propio contenido de la definición. A continuación se dan algunas definiciones más formales de relevancia recogidas en la recopilación de Blum y Langley (Blum y Langley, 1997).

Definición 3.2. (Relevancia con respecto al concepto) *Un atributo X_i es relevante para el concepto C si existe un par de ejemplos A y B en el dominio de instancias tal que A y B difieren solo en la etiqueta $y_A \neq y_B$*

Una definición similar a la anterior es la debida a Almuallim (Almuallim y Dietterich, 1994) pero restringida a atributos booleanos. Un inconveniente que posee la anterior definición es que si en el conjunto de aprendizaje $\mathcal{D}$ no se da ninguna situación como la expresada en la Definición 3.2 no se puede establecer la relevancia de ningún atributo. Por otra parte y debido a ruido en el conjunto de aprendizaje, se pueden detectar como atributos relevantes a aquellos que no lo son.

Para tener en cuenta el conjunto de aprendizaje $\mathcal{D}$ y de esta forma evitar el inconveniente que posee la anterior definición, John (John et al., 1984) da las siguientes definiciones de relevancia.

Definición 3.3. (Fuertemente relevante con respecto al conjunto de aprendizaje) *Un atributo x_i es fuertemente relevante con respecto al conjunto de aprendizaje $\mathcal{D}$, si existen dos ejemplos A y B que difieren solo en su asignación a X_i y tienen diferentes etiquetas. De la misma forma, X_i es fuertemente relevante al concepto definido por la etiqueta Y y cuya distribución es D si existe ejemplos A y B con probabilidad distinta de cero que difieren solo en la asignación a X_i y no tienen la misma etiqueta es decir, $y_A \neq y_B$.*

Definición 3.4. (Débilmente relevante con respecto al conjunto de aprendizaje) *Un atributo X_i es débilmente relevante con respecto al conjunto de aprendizaje $\mathcal{D}$ si eliminando un subconjunto de los atributos X_i pasa a ser fuertemente relevante según la definición anterior.*

Un elemento importante que aportan las anteriores definiciones es que permiten seleccionar los atributos en función del grado de relevancia, siendo los atributos fuertemente relevantes imprescindibles porque su eliminación añade ambigüedad a las

muestras del conjunto de aprendizaje. Sin embargo los atributos débilmente relevantes pueden mantenerse o no, dependiendo de qué atributos contenga el conjunto de atributos seleccionados. Las definiciones anteriores de relevancia fuerte y débil se pueden reescribir en base a probabilidades, para ello se define $\mathbf{S}_i$ como el conjunto de todas las características menos la característica X_i, y $\mathbf{s}_i$ una instanciación de dicho conjunto.

Definición 3.5. *X_i es fuertemente relevante si existe algún valor x_i de dicha característica, una clase Y y un conjunto $\mathbf{S}_i$ con valor $\mathbf{s}_i$ para los cuales $P(X_i = x_i, \mathbf{S}_i = \mathbf{s}_i) > 0$ tal que*

$$P(Y = y|X_i = x_i, \mathbf{S}_i = \mathbf{s}_i) \neq P(Y = y|\mathbf{S}_i = \mathbf{s}_i)$$

Definición 3.6. *Una característica X_i es débilmente relevante si y solo si no es fuertemente relevante, y existe un subconjunto de características $\mathbf{S}'_i \subset \mathbf{S}_i$ para el cual existe alguna X_i y clase Y para los cuales $P(X_i = x_i, \mathbf{S}'_i = \mathbf{s}'_i) > 0$ tal que*

$$P(Y = y|X_i = x_i, \mathbf{S}'_i = \mathbf{s}'_i) \neq P(Y = y|\mathbf{S}'_i = \mathbf{s}'_i)$$

Otra definición de relevancia debida a Wang (Wang, 1996) y que hace uso de conceptos de Teoría de la Información es la relevancia variable de una atributo respecto a la clase,

Definición 3.7. (Relevancia variable) *La relevancia variable del atributo X_i con respecto a la clase Y se define como la relación existente entre la información mutua entre el atributo y la clase, y la entropía de la clase.*

$$r(X_i, Y) = \frac{I(X_i; Y)}{H(Y)}$$

La relevancia variable se aplica también al caso condicional del conocimiento de un conjunto de atributos, denominándose entonces relevancia condicional. Una diferencia que existen entre la relevancia variable y las definiciones de relevancia fuerte o débil es que se introduce un grado de relevancia, es decir, no asigna un *todo o nada* a los atributos. Así $r(X_i, Y) = 1$ indica la relevancia máxima y a medida que decrece hacia cero la relevancia va disminuyendo.

Aunque Wang demuestra que los conjunto de variables relevantes según su definición poseen una complejidad mínima de acuerdo al principio de máxima entropía, en

Blum (Blum y Langley, 1997) se recoge otra definición de relevancia como medida de complejidad, más que hacer referencia a cuáles son los atributos relevantes.

Definición 3.8. (Relevancia como medida de complejidad) *Dado un conjunto de aprendizaje $\mathcal{D}$ y un conjunto de etiquetas Y, se define la relevancia $rc(\mathcal{D}, Y)$ como el número de atributos relevantes según la Definición 3.2 que posee el menor número de atributos relevantes entre aquellos que dan el menor error sobre el conjunto de aprendizaje.*

En la anterior definición más que seleccionar los atributos más relevantes, se utiliza la relevancia como una medida de la complejidad de la función ya que el objetivo es buscar el algoritmo que dé mejor resultado con la menor complejidad, es decir, encontrar el menor número de atributos necesarios para obtener un funcionamiento óptimo en el conjunto de aprendizaje con respecto al concepto que representa.

Todas las definiciones anteriores son independientes del algoritmo de inducción utilizado, pero puede resultar que un atributo que es relevante según algunas de las definiciones anteriores, utilizado con un determinado algoritmo de inducción no intervenga en el proceso de inducción por lo que se considera irrelevante por dicho algoritmo. Una definición que si considera esta situación es la "utilidad" incremental (Caruana y Freitag, 1994) definida como,

Definición 3.9. (Utilidad incremental) *Dado un conjunto de aprendizaje $\mathcal{D}$, un algoritmo de aprendizaje L, y un conjunto de atributos $\mathbf{S}$, un atributo X_i es incrementalmente útil para L con respecto a $\mathbf{S}$ si la tasa de acierto de la hipótesis que L produce usando el conjunto de atributos $\{X_i\} \cup \mathbf{S}$ es mejor que la tasa de acierto obtenida utilizando solo el conjunto de atributos $\mathbf{S}$.*

Esta definición se adapta bastante bien al esquema de selección de atributos como un proceso de búsqueda, que es lo que se describe en la siguiente sección.

3.3 Selección de Atributos como un Proceso de Búsqueda

La selección de atributos se puede considerar como un problema de búsqueda (Siedlecki y Sklansky, 1988; Langley, 1994) en un cierto espacio de estados, donde cada estado se corresponde con un cierto subconjunto de atributos, y el espacio engloba todas los posibles subconjuntos que se pueden generar. El proceso de selección de atributos puede

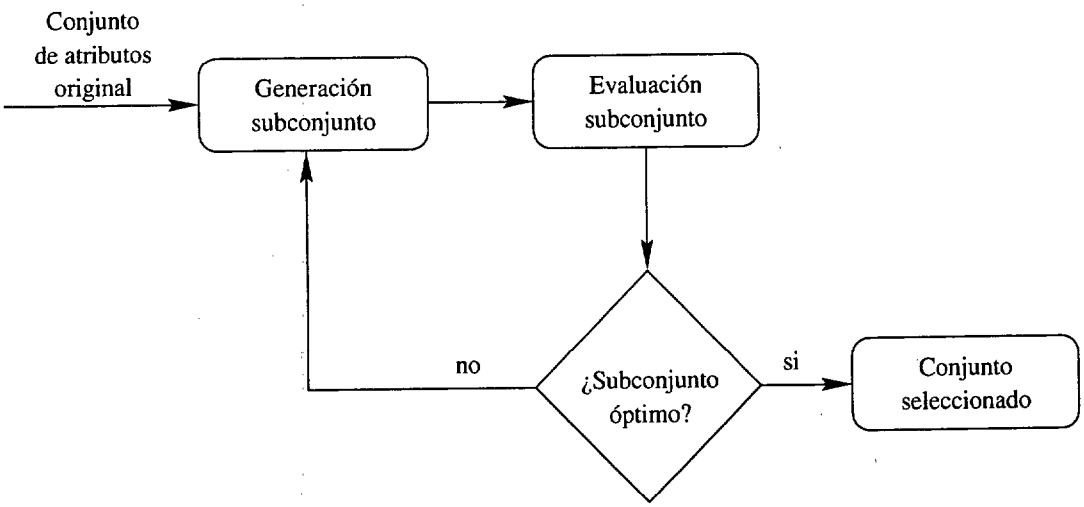


Figura 3.3: Proceso de selección de atributos

entenderse como el recorrido de dicho espacio hasta encontrar un estado (combinación de atributos) que optimice alguna función definida sobre un conjunto de atributos (Figura 3.3).

En la Figura 3.4 se muestra una representación gráfica del espacio de búsqueda correspondiente a un problema de selección de atributos, donde el conjunto inicial contiene cuatro atributos. En este caso particular cada estado se diferencia del anterior en un solo atributo, pero pueden existir otros espacios donde estados adyacentes se diferencien en más de un atributo, como se verá más adelante.

Una vez que se ha definido el espacio de búsqueda es necesario establecer un punto de inicio para empezar la búsqueda. Los dos puntos obvios, son comenzar con todos los atributos e ir eliminando a medida que avanza el proceso, o comenzar sin ningún atributo e ir añadiéndolos.

En los procesos de búsqueda es necesario establecer una estrategia para recorrer el espacio. Una posibilidad es la solución exhaustiva, que consiste en recorrer todo el espacio, pero debido a que el número de posibles combinaciones de los n atributos es $2^n - 1$, lo hace impracticable cuando el número de atributos es elevado, ya que para la selección del subconjunto de m atributos de forma exhaustiva es necesario comprobar los $\binom{m}{n} = \frac{n!}{m!(n-m)!}$ subconjuntos. Así, si se deben comprobar todos los subconjuntos desde dimensión 1 entonces el número es $\sum_{i=1}^{m} \binom{n}{i}$ que implica una complejidad $O(n^m)$, que como se muestra en (Davies y Russell, 1994) es un problema NP-completo.

Para evitar el recorrido de todo el espacio, se han definido estrategias que permiten obtener un subconjunto de atributos que no aseguran el óptimo pero que tienen un valor próximo con respecto a la función de evaluación utilizada. De entre las más

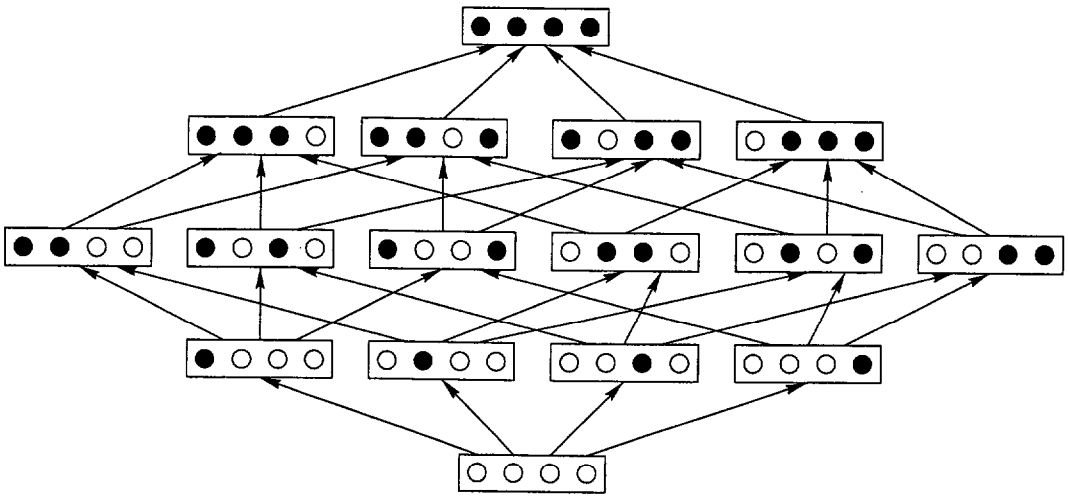


Figura 3.4: Espacio de búsqueda para cuatro características

utilizadas se encuentran las secuenciales en las que o bien se va añadiendo iterativa nuevos atributos a los ya seleccionados o bien se van eliminando del conjunto inicial. La primera estrategia se denomina Selección Secuencial hacia Adelante (SFS) y la segunda Eliminación Secuencial hacia Atrás (SBS). Un inconveniente de estos métodos es que no es posible de vuelta atrás ya que una vez se ha añadido un atributo se conserva hasta el final de la búsqueda.

Una modificación a los algoritmos secuenciales que permite la inclusión y eliminación de atributos es el algoritmo *Plus-l-Minus-r* (Devijver y Kittler, 1982) que permite añadir l atributos y eliminar r atributos en cada paso, aunque la elección de estos parámetros no es sencilla. Para evitar tener que fijar los valores de l y r, Pudil (Pudil et al., 1994) propone los métodos de Búsqueda Flotante (SFFS y SBFS), que se corresponde con el Plus-l-Minus-r dejando los parámetros l y r flotantes. La diferencia entre el SFFS y SBFS es que en el primero comienza sin atributos mientras que el segundo comienza con todos los atributos. El procedimiento para el SFFS (para el SBFS es simétrico) consiste en añadir en cada paso el atributo que provoque un mayor incremento de la función de evaluación y luego comenzar un proceso de eliminación condicional. Esta eliminación supone ir extrayendo atributos de forma que la cardinalidad del conjunto se vaya reduciendo, siempre que la función de evaluación para cada dimensión obtenida por eliminación de un atributo sea mayor que la que se obtuvo por adición de un atributo.

Las estrategias secuenciales SFS y SBS son las más sencillas de implementar pero la literatura documenta muchas más, aunque generalmente se pueden agrupar en unas pocas categorías atendiendo a dos criterios: primero a la región del espacio que recorren para encontrar el subconjunto de atributos óptimos y segundo a cómo se realiza este recorrido. Doak (Doak, 1994) establece tres categorías para las estrategias: exponencia-

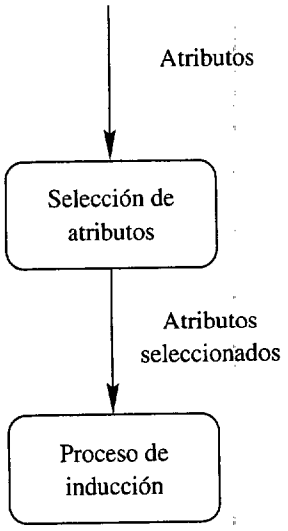


Figura 3.5: Estrategia Filtro

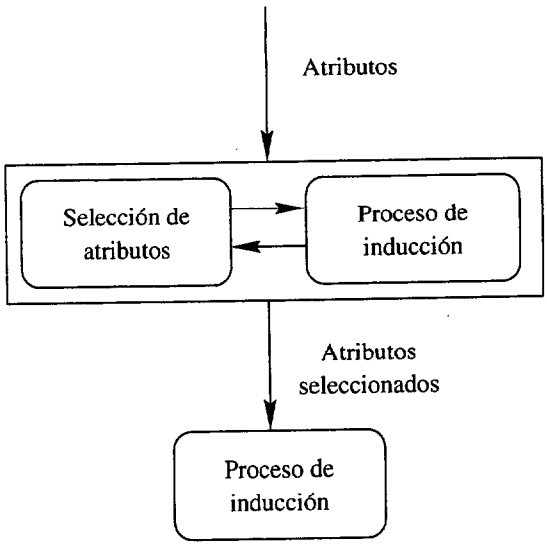


Figura 3.6: Estrategia Envolvente o *wrapper*

les, secuenciales y aleatorias. Dentro de las primeras se encuentran aquellas estrategias que tienen complejidad $O(n^m)$ pero aseguran la obtención del subconjunto óptimo. Las estrategias secuenciales a diferencia de las exponenciales recorren solo una porción del espacio de búsqueda y por tanto no aseguran la obtención del óptimo, aunque el coste computacional es polinomial. Las estrategias aleatorias se basan en visitar diferentes de regiones del espacio de búsqueda sin un orden predefinido, evitando de esta forma que se pueda obtener un óptimo local de la función de evaluación para un determinado subconjunto de atributos. Dash (Dash y Liu, 1997) propone una clasificación muy similar a la realizada por Doak, ya que también establece las siguientes tres categorías: completa, heurística y aleatoria, que equivalen a las realizadas por Doak. Jain y Zongker (Jain y Zongker, 1997) realizan una comparativa entre 15 métodos de búsqueda diferentes, incluyendo heurísticos, estocásticos y óptimos para un problema artificial de dos clases con funciones de densidad gausianas, por lo que el error es conocido e igual a la distancia de Mahalanobis, y encuentran que de los métodos probados los de Búsqueda Flotante tienen resultados bastante similares al Branch&Bound, pero de forma más rápida.

Como se ha comentado anteriormente, la estrategia de búsqueda intenta encontrar el subconjunto que optimice una determinada función de evaluación. Esta medida de evaluación estará definida para un conjunto de atributos y deberá medir la capacidad discriminante del conjunto de atributos para distinguir entre las diferentes clases definidas en el problema. Las funciones de evaluación utilizadas en los distintos trabajos de selección de atributos se han basado en diferentes criterios para medir la relevancia o capacidad de discriminación de los atributos. En la literatura existen varias taxonomías de estas medidas de evaluación atendiendo a diferentes criterios, dependiendo de los

autores.

Langley (Langley, 1994) agrupa las funciones de evaluación en dos categorías: Filtro y Envolvente (*wrapper*). En la primera categoría (Fig. 3.5), se incluyen los esquemas en los que la selección de atributos se realiza como un preproceso al proceso de inducción y por tanto de manera independiente, por lo que puede entenderse como un filtrado de los atributos irrelevantes. Por otro lado, en los esquemas de tipo envolvente (John et al., 1984)(Fig. 3.6), la selección de los atributos y la inducción de las reglas de clasificación no son elementos independientes, ya que la selección de los atributos hace uso del proceso de inducción para evaluar la calidad de cada conjunto de atributos seleccionados en cada momento.

Blum y Langley (Blum y Langley, 1997) establecen una clasificación de las funciones de evaluación utilizando también como criterio la dependencia que existe entre el proceso de selección y el de inducción, agrupándolas en cuatro categorías. La primera, que denomina de Esquemas Empotrados (*embedded*), recoge aquellas soluciones de selección de atributos donde no existe dicho proceso como tal, si no que viene dado por el propio esquema del algoritmo de inducción, como ocurre en los algoritmos que genera árboles de decisión o de descripciones lógicas, que utilizan solo aquellos atributos necesarios para obtener una descripción consistente con el conjunto de aprendizaje. La segunda y tercera categoría son la Filtro y la Envolvente comentadas anteriormente. La última categoría se encuadra los esquemas de selección basados en la ponderación de los distintos atributos. Un ejemplo de este esquema de selección puede observase en las redes neuronales donde un peso muy bajo asociado a ciertas entradas implica de facto poca relevancia en el proceso de clasificación, ya que su influencia en la salida es mínima.

A diferencia de las anteriores clasificaciones, Doak (Doak, 1994) establece una clasificación de las medidas de evaluación basada en la naturaleza de éstas, más que de su interrelación con el proceso de inducción, estableciendo tres categorías para las funciones de evaluación. A pesar de que el criterio es diferente las categorías, resultantes son similares a las anteriores. Por un lado se encuentran las medidas que utilizan propiedades intrínsecas a los datos. Estas se corresponden a los esquemas de tipo filtro, porque se basan en la información que obtienen del conjunto de aprendizaje. Otra categoría se refiere a las medidas basadas en la tasa de acierto de un clasificador, que son las que los otros autores han encuadrado dentro de las denominadas de tipo Envolvente. La última categoría incluye los métodos que calculan o estiman de forma incremental la tasa de acierto de un clasificador. Esta última categoría viene a ser un caso especial de la segunda, cuando la estrategia de búsqueda es secuencial.

Dash (Dash y Liu, 1997) realiza una taxonomía similar a la realizada por Doak,

pero dividiendo aquellas que se basan en propiedades intrínsecas de los datos, es decir las tipo filtro, en varias categorías dependiendo de que propiedades se extraen de los mismos. De esta forma la clasificación que establece de las funciones de evaluación es la siguiente: medidas de distancia, medidas de información, medidas de dependencia, medidas de consistencia y por otro lado las medidas basadas en la tasa de error de un clasificador que corresponderían a las de tipo envoltura.

Como en todo proceso de búsqueda, en la selección de atributos es necesario establecer un criterio de parada que permita determinar cuando se ha encontrado el subconjunto de atributos para los que la función de evaluación da el valor óptimo. Una opción para fijar este criterio consiste en considerar que la función utilizada para medir la calidad de los atributos seleccionados sufre un proceso de saturación cuando encuentra el mejor conjunto de atributos. En este caso se detiene la búsqueda cuando a partir de un determinado estado, el valor de la función utilizada no mejora sustancialmente para los estados sucesores. Otra posibilidad es no establecer un criterio de parada y dejar evolucionar la estrategia de búsqueda utilizada hasta el final, para luego tomar el subconjunto de atributos para el cual la función de evaluación fue máximo. No fijar un criterio de parada supone que para la búsqueda exhaustiva se recorra todo el espacio de búsqueda, con el consiguiente coste computacional que ello supone, aunque con ello se puede obtener el subconjunto para el cual la función de evaluación utilizada da el valor óptimo. En el caso de funciones de evaluación que asignan un valor a cada atributo, un criterio de parada consiste en establecer un umbral, de forma que la búsqueda termina cuando la medida asociada a cada uno de los atributos no seleccionados en un determinado momento supera el umbral fijado.

3.4 Revisión de Trabajos en Selección de Atributos

En esta sección se realiza un recorrido por diferentes algoritmos propuestos en la bibliografía para realizar la selección de atributos. Los distintos algoritmos y métodos se clasifican en función de dos parámetros: estrategia para recorrer el espacio de búsqueda y función de evaluación de cada subconjunto de atributos. Con respecto al primer parámetro se establecen las mismas categorías que establece Doak: búsqueda exhaustiva, búsqueda heurística y búsqueda aleatoria. En lo que respecta a la función de evaluación se establecen las siguientes categorías: basada en medidas de distancia, basada en Teoría de la Información, basada en medidas de consistencia y basada en el error de un clasificador. En la Tabla 3.1 aparecen los trabajos más representativos recogidos en esta sección. En esta tabla se muestra el primer autor y el nombre del sistema, el tipo de

Autor	Evaluación	Búsqueda	Inicio	Parada
Aha (BEAM)	error clasificador	exhaustiva	aleatorio	nº iteraciones
Almuallim (FOCUS)	consistencia	exhaustiva	ninguno	consistencia
Bradley (FSV)	distancia	heurística	todos	saturación
Battiti (MIFS)	medida información	heurística	ninguno	nº atributos
Caruana (BSE-SLASH)	error clasificador	heuristica	todos	saturación
Cardie (IG-CBL)	medida información	heurística	todos	consistencia
Domingos (RC)	error clasificador	heurística	todos	saturación
Foroutan (AMB&B)	error clasificador	exhaustiva	todos	saturación
Fukunaga (B&B)	distancia	exhaustiva	todos	umbral
González (SLAVE)	medida información	aleatoria	aleatorio	consistencia
Holte (1-R)	error clasificador	heurística	ninguno	nº atributos
Inza (FSS-EBNA)	error clasificador	aleatoria	n/2	saturación
Kira (Relief)	distancia	heurística	todos	umbral
Koller	medida información	heurística	todos	nº atributos
Langley (OBLIVION)	error clasificador	heurística	todos	saturación
Liu (LVF)	consistencia	aleatoria	ninguno	umbral
Moore (RACE)	error clasificador	heurística	ninguno	saturación
Scherf (EUBAFES)	distancia	heurística	todos	umbral
Skalak (RMHC)	error clasificador	heurística	aleatorio	nº iteraciones
Vafaie (IS)	error clasificador	heurística	$\lceil\sqrt{n}\rceil$	saturación
Wang (CR)	medida información	heurística	ninguno	saturación
Wang (VCC)	consistencia	heurística	ninguno	umbral

Tabla 3.1: Clasificación de los métodos de selección de atributos

función de evaluación utilizada, la estrategia de búsqueda, el punto de comienzo de la búsqueda y por último el criterio de parada si existe.

3.4.1 Medidas de Distancia en el Espacio de Atributos

En el espacio referido cada muestra se considera un punto y los conceptos en este espacio se suponen que forman una región compacta. Por tanto el conjunto de atributos a seleccionar es aquel que proyecta el conjunto de entrenamiento de forma que la región que ocupa cada clase sea lo más compacta posible y la separación entre éstas sea máxima. Al igual que para el resto de medidas se describirán los métodos agrupados según la estrategia de búsqueda que utilizan (exhaustiva, heurística o aleatoria)

Para atributos continuos, si existen dos clases se puede realizar la separación de las muestras mediante un hiperplano en el espacio de características,

$$P = \{\mathbf{x}|\mathbf{x} \in \mathcal{R}^n, \mathbf{x}^T\mathbf{w} = \lambda\} \tag{3.1}$$

El vector $\mathbf{w}$ se le denomina vector de pesos y a γ peso umbral. El vector de pesos es normal al hiperplano y se encuentra a una distancia del origen de $\frac{|\gamma|}{\|\mathbf{w}\|_2}$. En muchos casos las muestras no forman dos conjuntos linealmente separados, por lo que la obtención del hiperplano de separación no es factible. Bradley (Bradley et al., 1998) propone para estos casos el algoritmo FSV que realiza una transformación del espacio de atributos en uno de menor dimensión siempre que las clases puedan ser linealmente separables o casi linealmente separables, mediante la obtención del hiperplano que minimice el promedio de muestras mal clasificadas por el mismo. Para ello define dos hiperplanos fronteras correspondiente cada uno a una clase de forma que el hiperplano buscado se encuentre en medio y sea paralelo a ellos. Las expresiones de estos planos son equivalentes a una formulación de programación lineal robusta (RLP) y por tanto se pueden aplicar técnicas de RLP para la obtención estos hiperplanos.

La selección de atributos con el algoritmo FSV se realiza mediante la supresión de tantos elementos del vector $\mathbf{w}$ en (3.1) como sea posible mientras se mantenga una capacidad aceptable de separación del hiperplano resultante. La eliminación de los elementos del vector $\mathbf{w}$ se realiza por la multiplicación de los elementos del vector por una función escalón, que debido a la discontinuidad que posee en el método FSV se sustituye por una función cóncava exponencial para realizar la minimización sujeta a restricciones con la técnica de programación lineal. Evidentemente este algoritmo sólo se puede utilizar en problemas biclásicos donde los atributos sean continuos, y los resultados serán mejores en cuanto las clases sean más linealmente separables. El algoritmo SVM (Bradley y Mangasarian, 1998) se basa en el mismo concepto que el FSV pero el problema se resuelve utilizando el concepto de las Máquinas de Soporte Vectorial (*Support Vector Machine*)(Schölkopf et al., 1999; Cristianini y Shawe-Taylor, 2000).

Exploración Exhaustiva

El método Branch-and-Bound (B&B) propuesto por Narendra y Fukunaga (Narendra y Fukunaga, 1977), no realiza siempre un recorrido completo por todo el espacio de búsqueda, pero en el peor caso presenta una complejidad exponencial igual que los métodos de búsqueda exhaustiva aunque como éstos siempre asegura encontrar el conjunto de atributos que tiene valor óptimo para la función de evaluación. La función de evaluación usada en el método B&B se basa en la distancia de Mahalanobis (Duda y Hart, 1973) entre los atributos seleccionados y la clase. La ventaja del método se encuentra precisamente en no precisar una búsqueda exhaustiva para obtener el conjunto de atributos para el cual la función de evaluación óptima (máxima o mínima). Para ello, la función debe ser monótona frente a la adición de atributos, es decir que la adición de un

Algoritmo 2 Algoritmo Relief

```
Separar las muestras en positivas S+ y negativas S-
Inicializar vector de pesos W = (0,... 0)
for i=0 to m do
  Seleccionar aleatoriamente una muestra X ∈ S
  Seleccionar aleatoriamente una de las muestras positivas más cercanas a X, Z+ ∈ S
  Seleccionar aleatoriamente una de las muestras negativas más cercanas a X, Z- ∈ S
  if X es positiva then
    Near-Hit=Z+; Near-Miss=Z-
  else
    Near-Hit=Z-; Near-Miss=Z+
  end if
  for i=1 to n do
    W_i = W_i - diff(x_i, near_hit_i)^2 + diff(x_i, near_hit_i)^2
  end for
end for
Relevance = (1/m)W
for i=1 to n do
  if Relevance_i mayor que nivel de relevancia then
    Seleccionar atributo i
  end if
end for
```

nuevo atributo al conjunto existente no decremente el valor de la misma. El criterio de parada en este algoritmo viene definido por el número de atributos que debe contener el conjunto de atributos seleccionados. Una vez establecido este número, el método parte de todos los atributos y se van eliminando uno a uno conservándose en cada momento conjunto de dimensión igual a la indicada y que posea el mayor valor de la función de evaluación ya que este valor va a ser utilizado como un umbral podar el espacio de búsqueda. Esta reducción se basa en podar la búsqueda a partir de todos los conjuntos en los que la función de evaluación sea menor que el umbral. Este proceso de poda se basa en la propiedad de monotonía de la función de evaluación.

Exploración Heurística

Relief (Kira y Rendell, 1992) es un algoritmo inspirado en el aprendizaje basado en casos que intenta obtener los atributos estadísticamente más relevantes para un determinado concepto. El algoritmo se basa en asignar un peso a cada atributo y seleccionar los atributos cuyo peso supera un umbral prefijado (*nivel de relevancia*). El peso asociado a cada atributo se calcula a partir de la distancia euclídea entre el valor del atributo de una muestra y las muestras denominadas *Near-Hit* y *Near-Miss* (Algoritmo 2), que

se corresponden con la muestras positivas y negativas más cercanas, respectivamente, según la distancia euclídea. El concepto de Near-Miss ya aparece en (Winston, 1980) definido como "el ejemplo negativo que difiere de uno positivo en un número pequeño de atributos", que trasladado a un enfoque de distancia se corresponde con el concepto de proximidad, como lo utiliza Kira y Rendell. El algoritmo permite utilizar atributos continuos y discretos. Para los primeros se utiliza la distancia euclídea y para los segundos se asigna una distancia cero si los valores del atributo son iguales y uno si son diferentes. Aunque el algoritmo original solo admite la selección de atributos para un problema donde existen ejemplos positivos y negativos de un concepto, Kononenko (Kononenko, 1994) propone varias extensiones para problemas multiclásicos así como para permitir valores perdidos en el conjunto de aprendizaje. Como comentan los autores, un inconveniente del método es la incapacidad de detectar atributos redundantes, ya que si existen varios, los seleccionará todos aunque con solo una parte de los mismos se pueda resolver el problema.

EUBAFES (Scherf y Brauer, 1997), al igual que Relief, es un algoritmo basado en distancia, que modificando el peso de cada atributos refuerza, la similaridad entre muestras pertenecientes a la misma clase y la disimilitud entre muestras de diferentes clases. A diferencia de Relief, la modificación de los pesos de cada una de los atributos se realiza de forma que se minimice el valor de la siguiente función

$$J = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \delta_{ij}^{kNN} \left(\delta_{ij} \frac{d_{ij}(\mathbf{W})}{N_s} - (1 - \delta_{ij}) \frac{d_{ij}(\mathbf{W})}{N_v} \right) \tag{3.2}$$

donde

$$d_{ij}(\mathbf{W}) = \sqrt{\sum_k w_k \rho(X_k^{(i}, X_k^{(j}) + \tau} \tag{3.3}$$

$$\rho(X_k^{(i}, X_k^{(j}) = \begin{cases} |X_k^{(i}, X_k^{(j}| & \text{si atributo } k \text{ es continuo} \\ 0 & \text{si atributo } k \text{ es nominal y los valores son diferentes} \\ 1 & \text{si atributo } k \text{ es nominal y los valores son iguales} \end{cases}$$

y δ_{ij} vale uno si la muestra *i-ésima* es de la misma clase que la muestra *j-ésima* y cero en caso contrario; y δ_{ij}^{kNN} es uno si la muestra *j-ésima* está entre los k vecinos más cercanos de la muestra *i-ésima*. Por tanto, la función objetivo J es la suma de las interdistancias ponderadas de las muestras que en un entorno de los k vecinos más

cercanos ($\delta_{ij}^{kNN} = 1$) pertenecen a la misma clase ($\delta_{ij} = 1$) menos las pertenecientes a diferente clase ($1 - \delta_{ij} = 1$). La obtención de los pesos w_k en (3.3) se realiza mediante un proceso de optimización utilizando el descenso según el gradiente, restringiendo los valores de los pesos al intervalo $[0, 1]$ de forma que se seleccionarán solo los atributos cuyo peso sea 1. El número de atributos viene dado por el parámetro τ en (3.3) ya que el gradiente de (3.2) será más negativo (aumento de los pesos) cuanto mayor sea τ.

Fayyad e Irani (Fayyad y Irani, 1992) proponen un tipo de medidas denominada C-SEP para la inducción de árboles de decisión. Estas medidas intentan maximizar la distancia entre clases y la coherencia en cada clase. Un ejemplo de esas medidas que recogen en su trabajo es el coseno del ángulo que forman los vectores de clase. Estos vectores recogen la frecuencia de ocurrencia de muestras de cada clase en el conjunto de aprendizaje. El método se basa en realizar una partición por un atributo en dos conjuntos, y obtener los vectores de clases de cada partición. Si la partición es perfecta (en cada partición solo existen muestras de una clase) los vectores de clases son ortogonales y su coseno es mínimo, sin embargo si el atributo es irrelevante las dos particiones tendrán aproximadamente igual número de muestra de cada clase y los vectores de clase son prácticamente paralelos y por tanto la medida será máxima.

Exploración Aleatoria

Brill (Brill et al., 1992) utiliza un algoritmo genético para seleccionar los atributos que son utilizados como entrada a una red neuronal del tipo CounterPropagation. La función de ajuste del algoritmo genético se basa en la tasa de error del clasificador del vecino más cercano, y para acelerar la obtención del algoritmo genético se utiliza un muestreo del conjunto de aprendizaje. Otro elemento que se incluye en este trabajo es la introducción del algoritmo genético con Equilibrio Puntuado, donde varios algoritmos genéticos evolucionan por separado durante varias generaciones y luego intercambian los mejores individuos de cada población con el resto, lo cual hace el proceso de selección más rápido. Otra medida que utilizan para ir seleccionando los subconjuntos es la separación entre las clases a reconocer. La primera de las medidas se muestra superior a la segunda, ya que esta última no guarda una relación directa con el rendimiento del clasificador utilizado y que está basado en reglas.

3.4.2 Medidas basadas en Teoría de la Información

Las medidas basadas en Teoría de la Información han sido utilizadas ampliamente en selección de atributos. Yao (Yao et al., 1999) muestra un resumen de la utilización de medidas de este tipo (entropía, entropía condicional, información mutua, ...) en la detección de dependencias de atributos en el ámbito del KDD.

Exploración Heurística

Dentro de este apartado de búsqueda heurística, se incluye un conjunto de métodos que Blum (Blum y Langley, 1997) denomina Esquemas Empotrados (*embedded*) ya que no tienen como objetivo la selección de atributos sino la generación de árboles de decisión. Debido a que sólo se utilizan aquellos atributos necesarios para obtener una descripción consistente con el conjunto de aprendizaje, los seleccionan en función de medidas basadas en Teoría de la Información. Un algoritmo bastante utilizado para la inducción de árboles de decisión es el ID3 (Sec. 1.7) así como sus variantes. En ID3, en cada nodo se busca el atributo que divide el conjunto de aprendizaje dando lugar a una mayor Ganancia de Información. La Ganancia de Información como la define Quinlan (Quinlan, 1986), se corresponde con la información mutua (Def. 2.5) entre la clase y el atributo. Una modificación a la información mutua es lo que denomina Quinlan (Quinlan, 1993) como Relación de Ganancia (*Gain Ratio*) que es la relación entre la información mutua y la entropía del atributo X_i.

$$GR(X_i) = \frac{I(X_i;Y)}{H(X_i)}$$

Un inconveniente que aparece en general en todas las medidas basadas en Teoría de la Información es el sesgo hacia los atributos con un mayor número de valores. Jun (Jun et al., 1997) realiza una modificación de la Ganancia de Información que denomina Ganancia Normalizada (*normalized gain*) dividiendo por el logaritmo del número de valores que puede tomar el atributo.

$$normalized\ gain(X_i, n_i) = \frac{gain(X_i)}{\log_2 n_i}, \quad n_i \geq 2$$

siendo n_i el número de valores que toma el atributo X_i. López de Mántaras (López de Mántaras, 1991) propone una medida para la inducción de árboles de decisión que no se ve tan influenciada por el número de atributos. La expresión de esta medida para el

atributo X_i, $d(X_i)$, es:

$$d(X_i) = H(X_i|Y) + H(Y|X_i) \tag{3.4}$$

En este trabajo se demuestra que la medida $d(X_i)$ es una medida de distancia. Además haciendo uso de las propiedades de la entropía vistas en el Capítulo 2 se obtiene que esta medida es equivalente a la propuesta por MacKay (Def. 2.6), como se demuestra a continuación para dos variables aleatorias discretas A y B,

$$\begin{aligned} d(A,B) &= H(A,B) - I(A;B) \text{ (ec. 2.8)} \\ &= H(A,B) - H(A) + H(A|B) \\ &= H(B|A) + H(A|B) \text{ (ec. 3.4)} \\ &= d(A) \end{aligned}$$

Una aplicación de los métodos de inducción de árboles como algoritmo de selección de atributos aparece en (Cardie, 1993), donde se propone la utilización de árboles de decisión generados con el algoritmo C4.5 para seleccionar atributos en un problema de Procesamiento del Lenguaje Natural. Esta aproximación se basa en considerar que todos los atributos no utilizados en el árbol generado son irrelevantes. El problema consiste en obtener tres datos (parte del discurso, semántica general y semántica específica) asociados a cada palabra, utilizando para ello 35 atributos. Con estos 35 atributos se obtienen tres árboles para cada uno de los datos y se consideran atributos relevantes para cada dato, los utilizados en el árbol correspondiente. Utilizando estos atributos para la clasificación utilizando los k vecinos más cercanos, los autores obtienen mejores resultados que los que se obtienen con los árboles de decisión o con los k vecinos más cercanos con todos los atributos. Una modificación al anterior esquema de selección (Cardie y Howe, 1997) es el IG-CBL que es un algoritmo de ponderación por la información mutua de aquellos atributos utilizados en la inducción del árbol de decisión.

El algoritmo MIFS (Battiti, 1994) intenta aportar una solución aproximada al problema del elevado coste computacional que supone el cálculo de la información mutua entre un vector de atributos y una clase, ya que para un vector de atributos de una elevada cardinalidad es necesario disponer de millones de muestras. El Algoritmo (Algoritmo 3) se basa en calcular la información mutua de cada atributo con la clase y entre cada par de atributos. En el algoritmo se selecciona inicialmente el atributo con mayor información mutua con la clase y luego se van añadiendo atributos al conjunto de los ya seleccionados. Los atributos se van añadiendo de forma que aporten la mayor información sobre la clase y que no sean redundantes con la información ya aportada por

Algoritmo 3 Algoritmo MIFS

Inicializar: **F**=conjunto inicial de atributos y **S**=conjunto vacío
Para cada atributo X_i calcular $I(Y;X_i)$
Encontrar $X_0 = \max\{I(Y;X_i)\}$
$\mathbf{F} = \mathbf{F} - X_0$, $\mathbf{S} = \{X_0\}$
while $|S| \neq m$ **do**
  Para todos los pares de atributos (X_i, X_j) tal que $X_i \in \mathbf{F}$ y $X_j \in \mathbf{S}$, calcular $I(X_i;X_j)$
  Seleccionar el atributo $X_s = \max\{I(Y;X_s) - \beta \sum_{X_j \in \mathbf{S}} I(X_s;X_j)\}$
  $\mathbf{F} = \mathbf{F} - X_s$, $\mathbf{S} = \mathbf{S} \cup \{X_s\}$
end while

los atributos previamente seleccionados, por lo que se intenta maximizar la información mutua con la clase menos una cantidad proporcional a la suma la información mutua del atributo candidato con los ya seleccionados. De esta forma si dos atributos están altamente correlacionados, una vez seleccionado uno de ellos la selección del segundo se encuentra penalizada. El algoritmo termina cuando se ha seleccionado un número predeterminado m de atributos.

Un esquema bastante similar al anterior es el debido a Setiono y Liu (Setiono y Liu, 1996) donde proponen un método de selección de características para reducir el número de entradas a una red neuronal. Al igual que el algoritmo MIFS se utiliza una heurística para evitar el cómputo de la información mutua entre la variable aleatoria multivariante formada por todos los atributos y la clase. En este trabajo se calcula la información mutua entre cada atributo y la clase y entre cada par de atributos y la clase, es decir, se considera la información que aporta cada atributo de forma independiente y en unión de otro. De esta forma se realiza una división de los atributos en: seleccionables de primer orden y seleccionables de segundo orden. Los primeros son aquellos cuya información mutua con la clase es mayor que el promedio de las información mutua de todos los atributos. Todos los atributos que no son seleccionables de primer orden se comprueba si lo pueden ser de segundo orden. Para ellos se forman pares de atributos y se comprueba si la información mutua del par de atributos con la clase normalizada y sin normalizar es mayor que el promedio. Aquellos atributos que no son seleccionables de primer ni de segundo orden se denominan atributos eliminables de segundo orden y no se incluyen en el conjunto de atributos relevantes. Yang (Yang y Moody, 1999) propone un esquema muy similar al anterior pero para selección de características para visualización (selecciona dos características) basándose en la información mutua conjunta de pares de atributos, que es lo que Setiono denomina como atributos seleccionables de segundo orden.

La información mutua es un caso particular de la entropía relativa (Def. 2.4) que mide la similaridad de dos distribuciones. Koller y Sahami (Koller y Sahami, 1996) presentan un método basado en medir la similaridad de la distribución condicional de la clase con un conjunto de atributos $P(Y|\mathbf{X})$ con la distribución resultante después de eliminar un atributo $P(Y|\mathbf{G})$ ($\mathbf{G} = \mathbf{X}\backslash\{X_i\}$). A partir de la entropía relativa definen la medida Δ_G de la siguiente forma:

$$\Delta_G = \sum_{X_i} P(X_i) D(P(Y|\mathbf{X}), P(Y|\mathbf{G})) \tag{3.5}$$

donde $P(X_i)$ es la probabilidad del atributo X_i y $D(P(Y|\mathbf{X}), P(Y|\mathbf{G}))$ es la entropía relativa de las distribuciones antes mencionadas. El método en cada paso busca el subconjunto de atributos G que minimice Δ_G, es decir, el subconjunto que después de eliminado el atributo X_i tenga una distribución más similar a la que se obtiene con el conjunto de atributos completo.

Wang (Wang et al., 1998; Wang et al., 1999) propone el algoritmo CR para la selección de atributos que utiliza una medida de relevancia, $r(\mathbf{X}, Y)$, entre el conjunto de atributos y la clase basada en conceptos de Teoría de la Información. Esta medida de relevancia cumple con los dos axiomas que establece Wang para que un subconjunto sea óptimo. El axioma de suficiencia establece que el mejor subconjunto seleccionado debe preservar la información del aprendizaje, que la define como la información que aporta todo el conjunto de atributos inicial y que se mide como la información mutua entre los atributos y la clase. Por tanto para cualquier subconjunto de atributos cuya información mutua con la clase sea igual a la del conjunto inicial de atributos se dice que preserva la información del aprendizaje. El axioma de necesidad establece el criterio para seleccionar el mejor de todos los subconjuntos que cumplen el axioma de suficiencia. Este axioma hace referencia a la capacidad de generalización del subconjunto a seleccionar y que está relacionada con el principio de la Cuchilla de Occam. La medida que utiliza Wang es la entropía conjunta entre el conjunto seleccionado y la clase. Así el subconjunto óptimo será el que conserve la información de aprendizaje (axioma de suficiencia) y posea menor entropía conjunta con la clase (axioma de necesidad). Wang incluye una medida de relevancia que recoge los dos axiomas anteriores para la selección del subconjunto óptimo:

$$r(\mathbf{X}, Y) = \frac{I(\mathbf{X}; Y)}{H(Y)} \tag{3.6}$$

siendo el subconjunto óptimo el que maximiza la anterior medida de relevancia. También

Algoritmo 4 Algoritmo CR

Calcular $r(X_i, \mathcal{Y})\ \forall X_i \in X$
$X_0 =$ atributo con mayor valor de relevancia $r(X_i, \mathcal{Y})$
BSFS $= \{X_0\}$
repeat
  Encontrar $X_i \notin$ **BSFS** y $r(X_i, Y|\mathbf{BSFS})$ sea máxima
  BSFS $=$ **BSFS** $\cup \{X_i\}$
until $r(\mathbf{BSFS}, Y) = 1$
Devolver **BSFS**

define la relevancia para dos conjuntos conocido un tercero como:

$$r(\mathbf{X}, Y|\mathbf{Z}) = \frac{I(\mathbf{X}; Y|\mathbf{Z})}{H(Y|\mathbf{Z})} \tag{3.7}$$

La utilización de esta medida con una estrategia de búsqueda secuencial se muestra en el algoritmo CR (Algoritmo 4)

CFS (*Correlation-based Feature Selection*) (Hall, 2000) es un algoritmo basado en Teoría de la Información que utiliza una estrategia de búsqueda heurística para llevar a cabo la selección de atributos. En CFS se intenta obtener el conjunto de atributos más correlacionados con la clase y menos correlados entre si, utilizando la siguiente medida de mérito para el conjunto de atributos **S**,

$$Merit_{\mathbf{S}} = \frac{m\overline{r_{cf}}}{\sqrt{m + k(m-1)\overline{r_{ff}}}}$$

siendo m el número de atributos del conjunto **S** y $\overline{r_{cf}}$ y $\overline{r_{ff}}$ medidas de correlación con la clase y entre atributos respectivamente. En el caso de problemas con clases discretas la medida de correlación es la información mutua normalizada mientras que para problemas donde las clases sean continuas (problemas de regresión), Hall propone la correlación lineal modificada cuando uno de los atributos es discreto y el otro continuo o bien los dos discretos.

Algunos autores (Daelemans y van den Bosch, 1992; Wettschereck y Dietterich, 1995) han utilizado la información mutua como un factor de ponderación de los elementos que forman los conjuntos de atributos continuos, cuando se utilizan para el cálculo de distancias, de forma que aquellos atributos con menor información mutua contribuyan menos al valor de la medida que otros con mayor información mutua.

Exploración Aleatoria

SLAVE (González y Pérez, 1997) es un algoritmo de aprendizaje genético para generación de reglas que realiza la selección de atributos en dos fases diferentes. En la primera utiliza la información mutua normalizada ([0,1]) entre cada atributo y la clase y se fija aleatoriamente un umbral, de forma que aquellos atributos con un valor menor a ese umbral no son utilizados en el proceso de inducción de reglas. Una vez el algoritmo genético comienza el proceso de generación de reglas utilizando los atributos cuyo valor de información mutua supera el umbral, este valor va modificándose mediante operadores de cruce y mutación para valores reales. La segunda fase en la selección de los atributos se lleva a cabo cuando se han obtenido las reglas que son consistentes con el conjunto de aprendizaje. Aquellos atributos del consecuente que toman todos los valores posibles se eliminan ya que son irrelevantes.

3.4.3 Consistencia con el Conjunto de Aprendizaje

Los métodos de selección encuadrados dentro de esta categoría se basan en medidas de consistencia en las que básicamente se busca el conjunto de atributos menor que siga siendo consistente con el conjunto de aprendizaje o sea consistente con el mayor número de muestras existentes. Al igual que otros métodos basados en la consistencia, la existencia de ruido en el conjunto de aprendizaje degrada su rendimiento, en este caso la calidad de los atributos seleccionados.

Exploración Exhaustiva

El algoritmo FOCUS propuesto por Almuallim (Almuallim y Dietterich, 1992; Almuallim y Dietterich, 1994) se basa una implementación del *Espacio de Versiones* (Sec. 1.7.1) que denomina MIN-FEATURES y que consiste en obtener una de las hipótesis que contenga el menor número de atributos relevantes, según la definición de relevancia (Definición 3.2). El espacio de hipótesis considerado es el de funciones lógicas expresables mediante productos lógicos, y se demuestra que el número de ejemplos suficiente para el aprendizaje de este tipo de funciones crece logarítmicamente con el número de atributos irrelevantes en un marco de aprendizaje PAC. Para realizar esta búsqueda se intenta encontrar el conjunto de atributos de menor tamaño que recubre todos los conflictos, definiendo un conflicto como dos muestra con el mismo valor para el atributo pero pertenecientes a clases diferentes. La búsqueda del conjunto suficiente se realiza primero sobre todos los conjuntos de atributos de tamaño 1, luego de tamaño 2 y así

sucesivamente. Lo que da como resultado un coste computacional de orden exponencial, que coincide con la demostración de Davies (Davies y Russell, 1994) de que este es un problema NP completo.

Para reducir este coste el algoritmo FOCUS-2 utiliza una cola donde se almacena la parte del espacio de búsqueda que contiene los conjuntos de atributos que pueden contener el subconjunto de atributos relevantes, aunque en el caso peor, este algoritmo se comporta como FOCUS. Para dominios donde el número de atributos es elevado, la utilización de los algoritmos anteriores puede ser inviable, es por ello que los autores proponen tres heurísticas basadas en la búsqueda secuencial hacia adelante y que se diferencian en el criterio utilizado para ir seleccionando el siguiente atributo. La denominada MIG se basa en ir añadiendo el atributo que entre los no seleccionados posea menor de entropía. Otra heurística que proponen es la SG que se basa comenzar con todos los conflictos existente e ir seleccionando en cada momento el atributo que cubra el mayor número de conflictos no cubiertos. El último algoritmo que proponen los autores es el WG que asigna un peso a cada atributo. Este peso se obtiene en función del número de atributos que cubren un determinado conflicto, de forma que el peso asignado a un atributo por cubrir un conflicto es inversamente proporcional al número de otros atributos que también lo cubren. Un inconveniente que presenta este método es que está diseñado para problemas booleanos expresables como productos lógicos, algo que limita bastante su utilización en dominios con atributos continuos, ya que los nominales siempre se pueden convertir en atributos booleanos. Otro elemento bastante importante es la influencia del ruido en el conjunto de aprendizaje, ya que al estar basado en la resolución de conflictos si estos conflictos son debidos a ruido, el conjunto de atributos seleccionados no corresponde con los atributos relevantes al problema.

Exploración Heurística

Un algoritmo que realiza un recorrido heurístico del espacio de búsqueda y utiliza la consistencia como medida de la calidad de los atributos seleccionados es el propuesto por Wang (Wang y Sundaresh, 1998). Este algoritmo se basa en el criterio *Vertical Compactness* que realiza la compactación vertical del conjunto de aprendizaje es decir, calcula el promedio de inconsistencia para cada subconjunto de atributos. Cuando varias muestras poseen el mismo vector de atributos y difieren en la clase existe un conjunto de inconsistencias. La medida *inconsistency count* la definen como la cardinalidad de un conjunto de muestras con iguales valores para un subconjunto de atributos **S** menos el número de muestras pertenecientes a la clase mayoritaria, es decir, se considera la clasificación correcta para todas las muestras la clase mayoritaria y el resto de muestras

Algoritmo 5 Algoritmo LVF

```
Inicializar C_best = n; número de atributos del problema
for i=0 to MAX_TRIES do
  Seleccionar aleatoriamente un subconjunto de atributos S
  C = número atributos de S
  if C < C_best then
    if InconCheck(S, D) < γ then
      C_best = C; S_best = S
      Dar como mejor conjunto S_best
    end if
  else if C = C_best y InconCheck(S, D) < γ then
    Dar como mejor conjunto S_best
  end if
end for
```

perteneciente a otras clases se consideran inconsistencias. El promedio de inconsistencia se define como la suma de todas las cuentas de inconsistencias dividido por en número total de muestras. Este promedio de inconsistencia se corresponde con la tasa de error cuando se resuelven las inconsistencias asignándolas a la clase mayoritaria. El subconjunto que se selecciona es el de menor dimensionalidad que no supere un umbral fijado por el usuario. El procedimiento de búsqueda que emplean es un híbrido entre la búsqueda en profundidad y en anchura.

Exploración Aleatoria

Liu y Setiono (Liu y Setiono, 1996*c*) proponen el algoritmo LVF basado a su vez en el algoritmo de búsqueda aleatorio Las Vegas (Brassard y Bratley, 1996). Este algoritmo realiza saltos aleatorios en espacio de búsqueda para permitir la localización del conjunto de atributos más rápidamente. El uso de la aleatoriedad en este algoritmo es tal que la solución siempre se encuentra aunque se realicen elecciones erróneas a costa de más tiempo de cómputo. El algoritmo LVF (Algoritmo 5) consiste en generar aleatoriamente conjuntos de atributos **S** e ir seleccionando en cada momento aquel que tenga el menor número de atributos C_{best} y cuyo promedio de inconsistencia sea menor que el umbral γ fijado por el usuario.

En el algoritmo LVF los dos elementos que definen la calidad del conjunto de atributos seleccionados son el parámetro MAX_TRIES y el criterio de consistencia $InconCheck(S, D)$. El primero define el número de subconjuntos que se comprueban. Debido a que es un proceso aleatorio, si este número es muy bajo es improbable obtener el mejor subconjunto mientras que si es muy alto se realizarán muchas comprobaciones

sobre subconjuntos después de encontrar el mejor. En el trabajo mencionado, este valor se ha establecido de forma experimental en $77 \times n^5$ (n número inicial de atributos). El otro elemento que define la calidad del subconjunto seleccionado es el promedio de inconsistencias sobre el número total de muestras. Los autores indican que utilizando mecanismos de indexado se puede calcular el promedio de inconsistencia con un coste computacional de $O(n)$. Una propiedad de esta medida de inconsistencia es que es monótona y por tanto se puede obtener el conjunto de atributos con valor óptimo de la medida utilizando el algoritmo B&B.

En (Liu et al., 1998) se presenta el algoritmo ABB que es una implementación del algoritmo B&B utilizando la anterior medida. Del algoritmo LVF básico (Algoritmo 5) existen algunas variantes. LVS (Liu y Setiono, 1998*b*) o LVI (Liu y Setiono, 1998*a*), son versiones que permiten seleccionar los atributos sin utilizar inicialmente todos las muestras, ya que en problemas con un gran número de muestras el coste del cálculo del promedio de inconsistencia puede ser alto. Para ello divide el conjunto inicial de muestras en dos, D_0 y D_1, realiza el proceso de selección como en LVF en D_0 y luego comprueba en D_1. Si existen inconsistencias, las muestras que las producen se mueven a D_0 y se repite el proceso hasta que con los atributos seleccionados no se produzcan más inconsistencias.

El algoritmo LVW (Liu y Setiono, 1996*b*) es una modificación al algoritmo LVF donde la medida de evaluación es la tasa de error del árbol inducido por el C4.5. QBB (Dash et al., 2000) es un algoritmo híbrido ya que comienza utilizando el LVF para generar un conjunto de atributos que se utiliza como conjunto inicial en el algoritmo ABB que refina la búsqueda realizada por LVF.

3.4.4 Tasa de Error del Clasificador

Una diferencia que existe en la utilización del error de clasificación con respecto a las medidas anteriores es que éste se obtiene como una estimación del error (Capítulo 2) que no es una medida determinista (Kohavi, 1994) y por tanto al proceso de búsqueda se le añade incertidumbre. Este hecho normalmente no se ha tenido en cuenta por parte de los autores que utilizan el error del clasificador como medida de calidad del subconjunto seleccionado. Como se verá más adelante, Moore (Moore y Lee, 1994) sí hace uso de esta naturaleza estadística de la estimación en el proceso de búsqueda.

Los métodos basados en el error del clasificador obtienen el mejor conjunto de atributos para un determinado clasificador, pero presentan algunos inconvenientes en problemas donde el número de atributos es muy elevado o cuando se combinan con

técnicas de búsqueda como los algoritmos genéticos. Algunos autores (Martín Bautista y Villa, 1999) indican que la utilización del error del clasificador resta generalidad al resultado, ya que depende directamente de la bondad del clasificador elegido.

Exploración Exhaustiva

En algunos de los métodos que se exponen en este apartado se obtiene el subconjunto óptimo (con relación al error del clasificador) sin tener que realizar el recorrido completo del espacio de búsqueda (Fig. 3.4), aunque todos ellos en el caso peor sí tienen que recorrerlo completamente.

La utilización del algoritmo B&B de búsqueda utilizando como medida el error del clasificador, la proponen Ichino y Sklansky (Ichino y Sklansky, 1984). Una propiedad que debe cumplir el clasificador utilizado es que posea comportamiento monótono con el número de atributos (igual que la medida de divergencia utilizada originalmente en el B&B). En el trabajo de Ichino se utiliza el error de clasificador por hiperparalepípedos (*Box classifier*) que es monótono cuando es consistente con el conjunto de aprendizaje. Para asegurar esta propiedad los autores eliminan del conjunto de aprendizaje todas aquellas muestras incorrectamente clasificadas y por tanto inconsistentes con el clasificador. En un trabajo posterior (Foroutan, 1987) se incluye una modificación que permite que el clasificador no sea monótono. El clasificador utilizado es el lineal a intervalos que, sujeto a ciertas condiciones es monótono pero que en la aproximación utilizada no lo es. La modificación que se propone sobre el algoritmo B&B es que el umbral se incrementa en un porcentaje para poder evaluar combinaciones de atributos que de otra forma se podarían, pero que debido a la no monotonía del clasificador pueden dar lugar a combinaciones con valor menores que el umbral.

BEAM (Aha y Bankert, 1994; Aha y Bankert, 1995) es otro algoritmo de búsqueda que en el caso peor realiza una búsqueda exhaustiva y se basa en una modificación de la búsqueda de primero el mejor con una cola de tamaño predeterminado. En esta cola se almacenan en orden decreciente los subconjuntos que mejor tasa de acierto han obtenido con el algoritmo IB1, ya que la selección del siguiente subconjunto a explorar se hace de forma aleatoria con una probabilidad que es función de la posición que ocupan en la cola. La inicialización de la cola se realiza en una primera fase donde se genera aleatoriamente un número determinado de subconjuntos y se selecciona inicialmente el de mejor tasa de acierto, y a partir de éste comienza el proceso de exploración del espacio durante un número de iteraciones (generación de nuevos subconjuntos) definido por el usuario. Si el número de iteraciones y el tamaño de la cola se hacen suficientemente

grandes, este método se convierte en el método de búsqueda Mejor el Primero recorriendo todo el espacio; sin embargo si el tamaño de la cola se hace igual a 1 entonces se convierte en la Búsqueda Secuencial.

Davies y Russell (Davies y Russell, 1994) proponen un algoritmo basado FOCUS que mantiene una lista de todos los conjuntos de atributos de una determinada dimensionalidad, y en la cual se van eliminando aquellos conjuntos que dan lugar a un árbol de decisión que no clasifica correctamente las muestras utilizadas. La búsqueda es exhaustiva en el sentido, que cuando la lista está vacía se vuelve a inicializar con todos los conjuntos de dimensionalidad superior. Por tanto en el caso peor se llega a la lista que contiene todos los conjuntos (un único conjunto) con todos los atributos.

Exploración Heurística

El error de clasificación en una red neuronal es utilizado por Setiono (Setiono y Liu, 1997) para realizar la selección de atributos. La red neuronal es de tres capas completamente interconectada y una vez entrenada con todos los atributos, se procede a calcular el resultado de la red eliminando todas las conexiones de un determinado atributo con el nivel oculto. El atributo con el que la red decrementó menos su rendimiento es seleccionado y el proceso se repite sucesivamente mientras el rendimiento de la red no disminuya más de un cierto umbral respecto al rendimiento con todos los atributos. Para el entrenamiento de la red se utiliza como medida de error la entropía cruzada, añadiendo un término que incluye el tamaño de la red. Además, en lugar de utilizar la retropropagación del error (backpropagation) estándar se utiliza el algoritmo BFGS (Broyden-Fletcher-Shanno-Goldfarb)(Watrous, 1987) que es una variante del método quasi-Newton, ya que es más rápido que la propagación hacia atrás.

El algoritmo IS (*Importance Score*) (Vafaie y Imam, 1994; Imam y Vafaie, 1994) consiste en la ordenación de los atributos en base a la medida de dependencia *IS*. A partir de esta ordenación se obtienen aquellos atributos que inducen el conjunto reglas con menor error. El cálculo de la medida IS para cada atributo se realiza después de generar un conjunto de reglas de decisión con el algoritmo AQ (Sec. 1.7) utilizando todos los atributos. A partir de este conjunto inicial de reglas, el valor IS de un atributo se obtiene en función del número de reglas que contienen a dicho atributo, de forma que para un atributo que no aparece en ninguna regla el valor IS será nulo mientras que será uno si está en todas las reglas. Una vez se computa el valor IS para cada atributo, se seleccionan inicialmente aquellos que no superen un umbral[1] para el valor IS. Con estos

[1]los autores establecen un número que corresponde con el redondeo superior de la raíz cuadrada del

atributos se genera un conjunto de reglas cuyo error fijará el umbral para la adición de otros nuevos atributos, ya que se irán añadiendo sucesivamente y generando conjuntos de reglas mientras la tasa de error de las reglas generadas no incremente la tasa de error inicial.

Holte (Holte, 1993) también propone un método, denominado 1-R, basado en la tasa de error de las reglas generadas a partir del conjunto de atributos. A diferencia del algoritmo IS las reglas tienen un solo atributo y selecciona aquel cuya regla asociada produce un menor error. Los resultados que obtiene en las bases de datos con el algoritmo 1-R no son mucho peores en tasa de error a los que se obtienen con más atributos y reglas generadas con un algoritmo más sofisticado como el C4.5. Un inconveniente del método 1-R aparece cuando un atributo se comporta como identificador de cada muestra, tomando un valor diferente para cada una, en cuyo caso el método tiende a elegir este atributo ya que posee el menor error sobre el conjunto de aprendizaje, aunque evidentemente su capacidad de generalización es bastante mala. En este caso es necesario eliminar previamente este tipo de atributos.

Kohavi (Kohavi y Frasca, 1994) introduce el clasificador Holte-II de selección de atributos basado en la propuesta de Holte. La clasificación se realiza mediante la búsqueda de la muestra a clasificar en el conjunto de aprendizaje asignándole si se encuentra, la misma clase; si no se encuentra se le asigna la clase que es mayoritaria en dicho conjunto. En lugar de utilizar un solo atributo en el clasificador Holte-II se realiza una selección de atributos y los resultados son comparados por los obtenidos con el uso del C4.5, llegando a similares conclusiones que Holte.

Caruana y Freitag (Caruana y Freitag, 1994) realizan una comparativa de diferentes métodos heurísticos de selección de atributos en dos problemas obtenidos del sistema CAP (*Calendar Apprentice*) (Mitchell et al., 1994). En este trabajo la calidad de los atributos seleccionados se mide como la estimación del error de los árboles de decisión generados con ID3/C4.5 mediante la técnica *holdout*. Los métodos comparados son búsquedas secuenciales hacia adelante y atrás, incluyendo vuelta atrás. Una modificación de la búsqueda secuencial hacia atrás es la denominada BSE-SLASH donde en cada paso se eliminan todos los atributos no utilizados en el árbol generado por el algoritmo ID3. De los métodos comparados, los que incluyen búsqueda bidireccional dan los mejores resultados, aunque sin mucha diferencia con los otros dos.

Otro método basado también en la búsqueda secuencial hacia atrás es OBLIVION (Langley y Sage, 1994) en el que se utiliza como clasificador el árbol *oblivious*. Éste es

número de atributos.

un árbol de decisión donde todos los nodos en un determinado nivel hacen referencia al mismo atributo, por lo que se obtienen todas las posibles combinaciones de valores de los atributos. Una vez generado el árbol con todos los atributos, se procede a un proceso de poda de aquellos atributos que no influyen en la clasificación. Los autores de OBLIVION encontraron que un árbol *oblivious* es equivalente a un clasificador del vecino más cercano en el que se eliminan ciertos atributos en el cálculo de la distancia. En OBLIVION se prefiere la búsqueda hacia atrás ya que la eliminación de un atributo que se encuentra interrelacionado con otros supone un aumento en la tasa de error, razón por la cual en el trabajo de Caruana y Freitag el método BSE-SLASH da resultados similares a los de búsqueda secuencial hacia atrás o los que incluyen vuelta atrás.

El vecino más cercano y la búsqueda secuencial hacia atrás se emplea también en el método RC (Domingos, 1997), aunque a diferencia de OBLIVION la decisión para la eliminación se realiza utilizando información local a cada muestra del conjunto de entrenamiento. Así para cada muestra se eliminan aquellos atributos que son diferentes a los de la muestra más cercana de la misma clase. Si el error del clasificador utilizando la muestra sin estos atributos disminuye entonces se eliminan definitivamente y se continúa el proceso. Si el error no disminuye no se eliminan los atributos y se repite el proceso con otra muestra no utilizada previamente, hasta que se han utilizado todos las muestras.

Una aproximación basada también en el clasificador del vecino más cercano es el algoritmo RACE (Moore y Lee, 1994) que utiliza una heurística de búsqueda basada en la competición de diferentes conjuntos de atributos (*race*). Para cada conjunto de atributos se estima el error por validación cruzada dejando uno fuera, pero reduciendo el coste computacional mediante la modificación de los límites de Hoeffding por una aproximación bayesiana. En esta aproximación se va calculando la estimación del error comenzando con pocas muestras y luego añadiendo cada vez nuevas muestras, y se van eliminando ("en la carrera") aquellos conjuntos de atributos que tienen poca probabilidad de dar la menor tasa de error. Este trabajo sí tiene en cuenta la naturaleza no determinista de la estimación del error del clasificador como medida de la calidad en la selección de los atributos.

Kohavi y John (Kohavi y John, 1997) incluyen la utilización de los clasificadores ID3 y el clasificador bayesiano con dos estrategias de búsqueda heurística: primero el mejor y secuencial hacia adelante, en un estudio en diferentes tipos de bases de datos tanto artificiales como reales. En el trabajo incluyen una modificación del esquema de búsqueda hacia adelante con lo que denominan Operadores Compuestos (*Compound Operators*) que permiten explorar el espacio de forma más rápida y que tiene una relación con las diferentes categorías de relevancia (fuerte, débil e irrelevante). Los operadores

compuestos se generan dinámicamente después de crear el conjunto de hijos de un nodo del espacio y se utilizan para una expansión y luego se descartan. La generación de estos operadores se hace como composición de los operadores que han dado lugar a los mejores hijos en la expansión anterior del espacio de búsqueda. Por ejemplo si del nodo $(0,0,0,0,0)$ los mejores hijos que se obtienen son $(0,1,0,0,0)$ y $(0,0,0,0,1)$, es decir la adición de un nuevo atributo, el segundo y el quinto respectivamente, entonces el operador compuesto será la adición de esos dos atributos a la vez para generar la siguiente expansión.

Exploración Aleatoria

Siedlecki y Slansky (Siedlecki y Sklansky, 1989) proponen la utilización de un algoritmo genético para el recorrido del espacio de búsqueda. La función de ajuste usada en el algoritmo genético propuesto, se basa en obtener el subconjunto de atributos que tenga menor cardinalidad y con una tasa de error inferior a un umbral (*feasible factor*) fijado por el usuario. Este objetivo se consigue haciendo intervenir en la función de ajuste la tasa de error del clasificador 5-NN, el umbral y la cardinalidad del subconjunto, de forma que los subconjuntos con una tasa de error superior al umbral son penalizados independientemente del número de atributos.

Otro trabajo donde se utiliza un algoritmo genético para recorrer el espacio de búsqueda es debido a Vafaie y DeJong (Vafaie y De Jong, 1993; Vafaie y De Jong, 1994). El clasificador utilizado en este caso se basa en el conjunto de reglas generadas por el algoritmo AQ15, a partir del cual se obtiene la función de ajuste para cada conjunto de atributos. La función de ajuste se basa en la suma ponderada de las muestras clasificadas correctamente menos la suma ponderada de las clasificadas incorrectamente con el conjunto de reglas generadas. El peso que pondera la clasificación de cada muestra es la proporción en la que los atributos de dicha muestra aparecen en los antecedentes del conjunto de reglas generados con AQ15.

Guerra-Salcedo (Guerra-Salcedo et al., 1999) hace uso de Tablas de Decisión Euclídeas (EDT) como clasificador para estimar la calidad del conjunto de atributos seleccionados mediante un algoritmo genético. Las EDT se basan en las Tablas de Decisión por Mayoría (DTM) (Kohavi, 1995*b*), diferenciándose en que la asignación a una clase de las muestras que no aparecen en la tabla; en la DTM se asignan a la clase mayoritaria en la tabla, mientras que en la EDT se devuelve la clase de la muestra almacenada más cercana. Como estrategia de búsqueda se utilizan dos tipos de algoritmo genéticos denominado CHC y CF/RSC.

GADistAI (Yang y Honavar, 1998) es otro algoritmo de selección de atributos que utiliza un algoritmo genético que incluye en la función de ajuste la tasa de error de una red neuronal y el coste de la clasificación. La red neuronal utilizada (Yang et al., 1998), denominada DistAI, posee tres capas, estando la capa oculta compuesta por unidades esféricas, similares a las utilizadas en las Redes Neuronales basadas en Funciones de Base Radial y el cómputo de los pesos entre la capa oculta y la salida no se realiza de forma iterativa lo que acelera el proceso de aprendizaje y permite su utilización en el algoritmo de selección de atributos.

Aparte de los algoritmos genéticos utilizados en los trabajos anteriores, existe un tipo de algoritmos denominados EDA (*Estimation Distribution Algoritms*) con un funcionamiento similar a los algoritmos genéticos pero sin la utilización de los operadores de cruce o mutación, ya que las sucesivas generaciones se obtienen a partir de individuos de la población actual, con cada individuo de la nueva generación representado en función de su distribución de probabilidad en la generación actual. Los autores, para estimar la distribución de probabilidad en cada población utilizan una Red Bayesiana que tiene en cuenta la interdependencia entre los atributos utilizados dando lugar a lo que los autores denominan EBNA. La combinación del algoritmo EBNA (utilizando como población las combinaciones de características de la misma forma que se utiliza en los algoritmos genéticos) con un clasificador da lugar al algoritmo FSS-EBNA que proponen los autores. En el artículo mencionado se utilizan varios clasificadores como un árbol de decisión o el clasificador bayesiano, considerando que se ha encontrado el subconjunto de atributos relevantes cuando no hay un incremento en la tasa de acierto del clasificador utilizado en una determinada generación.

Otro método basado en el recorrido aleatorio del espacio de búsqueda, aunque no haciendo uso de los algoritmos genéticos es el algoritmo RMHC-PF (Skalak, 1994). Este algoritmo se basa en la búsqueda *Random Mutation Hill Climbing* (RMHC), que consiste en partir de un conjunto aleatorio de atributos y de forma aleatoria añadir uno nuevo o eliminar alguno de los contenidos en el conjunto e ir conservando el subconjunto que menor error produce con el clasificador del vecino más cercano.

3.4.5 Métodos no Encuadrados en la Clasificación Anterior

Existen algunos métodos y algoritmos en la literatura sobre Aprendizaje Automático que no se basan en las medidas vistas anteriormente o son diseñados para problemas de aprendizaje muy específicos. En esta sección se presentarán algunos de estos métodos.

Liu y Setiono (Liu y Setiono, 1995; Liu y Setiono, 1996*a*; Liu y Setiono, 1997)

proponen un método de selección de atributos basado en la discretización de los atributos numéricos. Antes de la selección se procede a discretizar los atributos numéricos mediante una variación del método ChiMerge (Kerber, 1992). Al final del proceso de discretización se eliminan los atributos que han dado lugar a un solo intervalo. El proceso de discretización se divide en dos fases. En la primera se discretizan todos los atributos partiendo de un valor alto del nivel de significación para el estadístico χ^2, que se decrementa en sucesivas iteraciones. En la segunda fase se discretiza cada atributo por separado, comenzando con el nivel de significación con el que se concluyó la primera fase. La condición de parada de esta segunda fase se fija por un promedio de inconsistencias con los valores discretizados en cada momento. Al concluir esta segunda fase, se eliminan aquellos atributos que han dado lugar a un solo intervalo. Este método solo es válido en la selección de atributos numéricos (ordinales) en problemas de aprendizaje supervisado, ya que la información sobre la clase en necesario para el estadístico χ^2.

Un método de ponderación de atributos específico para Procesamiento de Lenguaje Natural es el que propone Cardie (Cardie, 2000) para resolver el problema de obtención de la parte de la frase a la que hace referencia un pronombre relativo mediante un clasificador del vecino más cercano. En el trabajo se proponen tres esquemas de ponderación heurísticos basados en teorías cognitivas. El primer esquema, *Subject Accessibility Bias*, se basa en dar mayor peso al atributo que representa el sujeto de la frase ya que puede ser el referido por el pronombre. Otro esquema, *Recency Bias*, da mayor peso a aquellos atributos que corresponden con las partes de la frase más cercanas al pronombre. El tercer esquema que propone, *Restricted Memory Based*, se basa en seleccionar aleatoriamente un número de partes de la oración y descartar el resto ya que esto guarda relación con estudios que indican que, cuando leemos, solo se mantiene en memoria unos pocos elementos. En los experimentos descritos, ninguno de los tres esquemas de pesado de atributos mejora al clasificador sin ponderación. Por tanto, incluyen un algoritmo que busca combinaciones, de forma incremental y exhaustiva, de los anteriores esquemas con diferentes parámetros, obteniendo en este caso mejores resultados que con cada esquema por separado.

Rauber (Rauber y Steiger-Garção, 1993) presenta un método para selección de atributos nominales basado en tablas de contingencia que indican el número de muestras de cada clase que contiene cada uno de los valores del atributo. Si el atributo es relevante para la clase, la partición resultante por el atributo dará lugar a una tabla donde en cada fila existirá una celda con la mayoría de las muestras. Sin embargo, en el caso de los atributos irrelevantes las diferentes celdas de la tabla tendrán un número aproximadamente igual de muestras. Para detectar las dos situaciones, los autores uti-

lizan una medida R_0 basada en el test de hipótesis χ^2. Como limitación se encuentra que trata a los diferentes atributos de forma individual por lo que no detecta la posible correlación entre los diferentes atributos.

DIET (Kohavi et al., 1997) es un algoritmo que pondera los atributos para el cálculo de la distancia utilizada en el clasificador del vecino más cercano. Los pesos, a diferencia de otros trabajos comentados anteriormente, no utilizan ningún tipo de medida sino que el usuario define el número k de pesos que pueden tener los atributos y se genera un conjunto de pesos discretos $(0, 1/k, 2/k, \ldots, (k-1)/k, 1)$. El algoritmo comienza asignando el peso mediana ó 0 si $k = 1$, y la estrategia para ir modificando los pesos es Mejor el Primero, donde cada estado en el espacio se genera asignando a cada atributo el peso mayor y menor. La calidad de los pesos y por tanto de los atributos seleccionados, se evalúa siguiendo un modelo *wrapper* y se detiene la búsqueda cuando después de cinco expansiones no se produce un incremento en la tasa de acierto.

Mladenić (Mladenić, 1998; Mladenić y Grobelnik, 1999) en un problema clasificación de documentos utiliza para ponderar los atributos (palabras) la medida *OddsRatio*, que es utilizada en problemas biclásicos y con atributos booleanos. La expresión de la medida es:

$$OddsRatio = \log \frac{P(W|C_1)(1 - P(W|C_2))}{(1 - P(W|C_1)P(W|C_2)} \tag{3.8}$$

donde $P(W|C_i)$ es la probabilidad condicional de aparición de la palabra W dado que el documento pertenece a la clase C_i. Como se ha comentado anteriormente, la medida se plantea para problemas biclásico. En este caso la pertenencia o no del documento a la categoría de interés; al igual que los atributos, se fija en función de la aparición o no de la palabra en el documento. En los experimentos que llevan a cabo los autores, se demuestra que esta medida tiene mejor comportamiento que otras como la información mutua, en problemas donde el número de muestras de la clase positiva o de interés es pequeño (1%-10%) respecto al total de muestras, ya que la información mutua por ejemplo selecciona las muestras que dividen mejor las dos clases, pero seleccionando las características que definen la clase de no interés pero no las que definen la clase de interés. La utilización del clasificador bayesiano en los experimentos, se ve perjudicada frente a otros como el k-vecinos más cercanos, por este hecho, ya que utiliza para la clasificación las palabras (atributos) que aparecen en los documentos y no las que no aparecen, que serían las que seleccionarían otras medidas.

Capítulo 4

Modelo para la Selección de Atributos

En este capítulo se presenta la propuesta para la selección de atributos basada en conceptos de Teoría de la información. En la introducción se comenta que es posible establecer una analogía conceptual entre un clasificador y un canal de información, planteándose así la utilización de conceptos provenientes de la Teoría de la Información para abordar el proceso de selección de atributos. A continuación se modela el proceso de selección de atributos utilizando dichos conceptos. Se introduce el concepto de matriz de transinformación como un elemento que permite medir la interdependencia de los atributos dos a dos, obteniendo de esta forma una estimación de la dependencia de grupos de atributos sin necesidad de computar funciones de distribución multivariables. Por último se presenta la medida GD, útil para medir la dependencia de la clase con el conjunto de atributos así como algunos algoritmos que permiten su utilización en la selección de atributos. Por último se propone un esquema de sustitución para resolver la selección en problemas con conjuntos de atributos con valores perdidos.

4.1 Introducción

Una de las definiciones de Aprendizaje Automático recogidas en el Capítulo 1 debida a Mitchel (Mitchell, 1980) era la siguiente,

> "la habilidad para generalizar a partir de la experiencia pasada de forma que se puedan abordar nuevas situaciones que se encuentran relacionadas con la experiencia acumulada"

La anterior definición incluye una orientación que entiende el aprendizaje como algo diferente de la simple memorización de conceptos y es la capacidad de generalización del conocimiento adquirido para así clasificar (abordar el tratamiento de) nuevas muestras no utilizadas en el proceso de aprendizaje. La capacidad de generalización es posible ya que el conocimiento se estructura como un conjunto de mecanismos de clasificación que se obtienen como resultado del aprendizaje, utilizando para ello un conjunto de ejemplos del concepto o clase[1]. Lo importante del Aprendizaje Supervisado para que se posea capacidad de generalización es que debe existir un conocimiento previo del tipo de mecanismo de clasificación que pueden resolver el problema e introducirlo en el proceso de aprendizaje. Este sesgo que se introduce es imprescindible ya que de lo contrario la capacidad de generalización no iría más allá del conjunto de aprendizaje (Mitchell, 1997).

En el Capítulo 1 (Sec. 1.7) se introdujeron diferentes tipos de mecanismos de clasificación (clasificadores a partir de ahora) como árboles de decisión, redes neuronales o clasificadores basados en ejemplos, así como diferentes algoritmos de inducción que a partir de un conjunto de ejemplos intentan generar el clasificador que mejor abstrae (según algún criterio definido a priori) el concepto representado en el conjunto de muestras poseyendo a la vez capacidad de generalización sobre nuevos ejemplos no utilizado en el proceso de aprendizaje.

El proceso de obtención del clasificador puede caracterizarse como una búsqueda en el espacio de parámetros del clasificador, como es el caso en redes neuronales, o el espacio de posibles clasificadores, como ocurre con los árboles de decisión. El objetivo de la mayoría de los algoritmos de inducción es obtener un clasificador con una tasa de error baja. Debido a que no se dispone de todas las posibles muestras que definen el problema a tratar sino de un conjunto limitado, es necesario estimar la tasa de error que cometerá el clasificador sobre las muestras no utilizadas en el proceso de aprendizaje y que toma como una medida de la capacidad de generalización. Algunas de estas técnicas de estimación del error a partir del conjunto de muestras utilizadas también en el proceso de aprendizaje se vieron en la Sección 1.8.

El elemento quizás más importante en el Aprendizaje Automático son las muestras utilizadas en el proceso de aprendizaje. Éstas junto al tipo de clasificador a obtener establecen el sesgo de partida y son la únicas fuentes de información disponible por el algoritmo de inducción para generar el clasificador. Así el resultado del proceso de inducción, va a depender básicamente por un lado de la capacidad del clasificador seleccionado para separar las diferentes clases y por otro de la *calidad* de las muestras

[1]En este capítulo se utilizará clase para referirse al concepto objeto del proceso de inducción

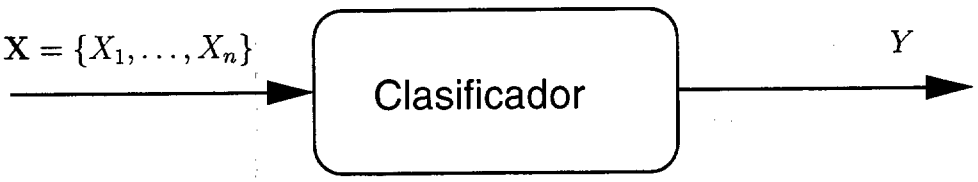


Figura 4.1: Representación esquemática de un clasificador

del conjunto de aprendizaje. Las muestras utilizadas en aprendizaje supervisado constan de n-tuplas donde cada elemento se corresponde a un atributo o característica[2] del individuo; junto con un indicativo de la clase a la que pertenece dicha muestra. Una definición formal de muestra se dio en la Sección 1.1.

Centrándonos en las muestras de aprendizaje, su calidad viene dada en principio por la de sus características. Una forma de medir la calidad de las características es la cantidad de información que aportan sobre la clase. Así cuanta más información aporten más interesantes serán a efectos del proceso de inducción. El proceso de obtención de las características de mayor calidad se denomina de *selección de atributos* y al igual que el proceso de inducción se puede modelar como un proceso de búsqueda en el espacio de todas las combinaciones posibles de características (Sec. 3.1) y responde básicamente a dos motivos: simplicidad en la hipótesis generadas e incremento del rendimiento de algunos clasificadores medido como tasa de acierto.

El objeto de estudio de esta tesis se centra en el problema de la selección de atributos en Aprendizaje Automático y no en el estudio de algoritmos de inducción o clasificadores, por lo que éstos se considerarán como una caja negra (Fig. 4.1) que recibe como entrada una muestra no etiquetada $\mathbf{X} = \{X_1, \ldots, X_n\}$ definida por un vector de características, y como salida, la clasificación o asignación a una clase $Y \in \mathcal{Y}$ de la muestra de entrada. Si consideramos esta visión del problema de clasificación con el clasificador aceptando muestras sin etiquetar y asignando una clase a dicha muestra, como muchos sistemas de información se puede estudiar su funcionamiento considerando la relación existente entre las entradas a dicho clasificador y la salida del mismo.

Tal y como se ha dicho, la calidad del proceso de inducción viene influenciado entre otros aspectos por la calidad de las características utilizadas en el mismo, siendo el vector de características la entrada al clasificador. Por tanto conviene que éstas aporten la mayor cantidad de información sobre la clase. La consideración del clasificador como una caja negra tiene una gran similitud con el estudio de los canales de información en

[2]Se utilizará atributo o característica indistintamente a lo largo del capítulo

Figura 4.2: Canal de información

Teoría de la Información (Sec. 2.1) donde se establece el modelado a partir del alfabeto de entrada y el alfabeto de salida y la relación que existe entre la transmisión de los símbolos del alfabeto de entrada y la recepción de un símbolo perteneciente al alfabeto de salida, sin considerar el medio, tecnología o proceso empleados en la transmisión. La similitud entre el clasificador y el canal de información está en la consideración de todas las posibles instancias del conjunto de atributos $\mathcal{X}$ como alfabeto de entrada del canal de información asociado al clasificador, y el concepto o clase $\mathcal{Y}$ como alfabeto de salida (Figura 4.2). Además existe otro elemento común en ambos casos y es la relación entre la entrada y la salida en el clasificador y el canal de información, ya que en este último esta relación viene dada por la matriz de canal $M = [P(b_i|a_j)]$ que recoge la probabilidad de obtener un determinado símbolo b_j a la salida cuando se ha enviado a la entrada a_i.

Desde el punto de vista del proceso de clasificación, el clasificador óptimo es el clasificador bayesiano que se basa en la probabilidades a posteriori y que para el caso de un conjunto de características nominal tiene la misma forma que la matriz del canal sustituyendo el símbolo de entrada por una instancia del conjunto de atributos y el símbolo de salida por la clase, $P(Y|\mathbf{X})$. Por tanto si se supone válida la anterior consideración, puede plantearse la utilización de conceptos de Teoría de la Información para la obtención del mejor conjunto de atributos como aquellos que más información aportan a la clase. En la siguiente sección se introduce la utilización de la información mutua como medida de la importancia o relevancia de un conjunto de atributos.

4.2 Relevancia como Información Mutua

Un concepto muy utilizado en Teoría de la Información es la cantidad de información que proporciona una variable aleatoria sobre otra, que trasladado al modelo del clasificador es la cantidad de información que proporciona el conjunto de atributos sobre la clase. Así el objetivo de la selección de atributos como la obtención de aquellos más relevantes,

resulta ser la obtención de los que más información aportan sobre la clase. Por tanto el problema de la selección de atributos se traduce en medir la cantidad de información que aportan los atributos sobre la clase y encontrar el conjunto de atributos que aportan más información, siendo necesario disponer de una medida de la cantidad de información que suministra un conjunto de atributos.

Una medida que recoge la cantidad de información entre dos variables aleatorias es la información mutua (Sec. 2.2). En el clasificador las dos variables aleatorias se corresponden con el vector de atributos y la clase, por tanto se puede obtener la información que un conjunto de atributos $\mathbf{X}$ aporta a la clase Y calculando la información mutua entre ambos $I(\mathbf{X};Y)$ y cuya expresión es,

$$I(\mathbf{X};Y) = \sum\sum P(\mathbf{x},y)\log\frac{P(\mathbf{x},y)}{P(\mathbf{x})P(y)} \tag{4.1}$$

La expresión 4.1 de la información mutua se puede expresar de forma incremental a partir de los atributos que forman el conjunto $\mathbf{X}$ haciendo uso de la propiedad asociativa de la misma (Cover y Thomas, 1991, pág. 22), de lo que además se deducirá que es una medida creciente con la dimensionalidad. Así, denominando $\mathbf{X}^{(n-1}$ al conjunto de atributos $\{X_1, X_2, \ldots, X_{n-1}\}$, entonces la información mutua del conjunto $\mathbf{X} = \mathbf{X}^{(n} = \{\mathbf{X}^{(n-1}, X_n\}$ se puede calcular como,

$$I(\mathbf{X}^{(n};Y) = I(\{\mathbf{X}^{(n-1}, X_n\};Y) = I(\mathbf{X}^{(n-1};Y) + \Delta I_{n-1} \tag{4.2}$$

donde:

$$\Delta I_{n-1} = I(X_n;Y|\mathbf{X}^{(n-1})$$

Si en la ecuación (4.2) se sustituye $I(\mathbf{X}^{(n-1};Y)$ por su expresión, y en esta última se sustituye $I(\mathbf{X}^{(n-2};Y)$, y así sucesivamente se llega a la siguiente expresión de la información mutua,

$$I(\mathbf{X}^{(n};Y) = I(X_1;Y) + I(X_2;Y|X_1) + \cdots + I(X_n;Y|\mathbf{X}^{(n-1})$$

de forma que se puede calcular recursivamente según:

$$I(\mathbf{X}^{(n};Y) = I(X_1;Y) + \sum_{j=2}^{n} I(X_j;Y|\mathbf{X}^{(j-1}) \tag{4.3}$$

A continuación se hace uso de la información mutua en la definición de relevancia de atributos y conjuntos de atributos.

Definición 4.1. *Un atributo X_i se dice* ***no relevante*** *respecto a la clase Y si se cumple que $I(X_i;Y) = 0$*

Definición 4.2. *Un atributo X_i se dice* ***relevante*** *respecto a la clase Y si se cumple que $I(X_i;Y) > 0$*

De las anteriores definiciones se puede obtener la relación de atributo más relevante como el atributo X_{mr} que cumple $I(X_{mr};Y) = \max_{\forall i=1\ldots n}\{I(X_i;Y)\}$; o la de relevancia entre atributos, siendo X_i más relevante que X_j si se cumple que $I(X_i;Y) > I(X_j;Y)$.

Las anteriores definiciones hacen referencia únicamente a atributos aislados, considerados de forma independiente, pero en muchos casos la dependencia con la clase es de un conjunto de atributos considerado como un todo y no como elementos aislados. Por tanto las definiciones anteriores se modifican para conjuntos de atributos. Así sea $\mathbf{Z}$ un conjunto de atributos tal que $\mathbf{Z} \subseteq \mathbf{X}$, se tiene que

Definición 4.3. *Un atributo $X_i \notin \mathbf{Z}$ es* ***relevante conjuntamente con*** $\mathbf{Z}$ *en la definición de Y si siendo $I(Y;X_i) > 0$, se cumple que:*

$$I(\mathbf{Z} \cup X_i;Y) > I(\mathbf{Z};Y)$$

Definición 4.4. *Un atributo $X_i \notin \mathbf{Z}$ es* ***no relevante respecto al conjunto*** $\mathbf{Z}$ *en la definición de Y si siendo $I(X_i;Y) > 0$, se cumple que:*

$$I(\mathbf{Z} \cup X_i;Y) = I(\mathbf{Z};Y)$$

La utilización de la información mutua en la selección de atributos ha sido propuesta por diferentes autores (Sec. 3.4.2). Wang (Wang, 1996) define la información mutua entre el conjunto de atributos y la clase como la *información del aprendizaje.*

De la expresión incremental de la información mutua (ec. 4.2) se deduce fácilmente que la mayor cantidad de información del aprendizaje que puede obtenerse es la que proporciona el conjunto de atributos inicial.

Proposición 4.1. *Sea un conjunto de atributos $\mathbf{X} = \{X_1,\ldots,X_n\}$ y una clase Y, no existe ningún subconjunto $\mathbf{Z} \subseteq \mathbf{X}$ que aporte más información que el conjunto completo de atributos, es decir, $I(\mathbf{X};Y) \geq I(\mathbf{Z};Y)\ \forall\, \mathbf{Z} \subseteq \mathbf{X}$*

Demostración. Se deduce de forma trivial de la expresión (4.2). □

Haciendo uso de la Proposición 4.1, Wang define el Subconjunto de Atributos Suficiente (SAS) para el aprendizaje como:

Definición 4.5. *Sea* $\mathbf{Z} \subseteq \mathbf{X}$, *se dice que* $\mathbf{Z}$ *es un* ***subconjunto de atributos suficiente (SAS)*** *para el aprendizaje si aporta la misma información que el conjunto completo de atributos,* $I(\mathbf{Z};Y) = I(\mathbf{X};Y)$.

De la definición anterior se deduce que para cada subconjunto de atributos suficiente existe un subconjunto de atributos no informativos $\mathbf{W}$, compuesto por los atributos pertenecientes al conjunto inicial de atributos y no incluidos en el subconjunto de atributos suficiente ($\mathbf{W} = \mathbf{X} \backslash \mathbf{Z}$). Este conjunto de atributos puede ser eliminado del proceso de clasificación ya que no aporta ninguna información al proceso de aprendizaje.

Proposición 4.2. (Wang, 1996) *Sea, sin perdida de generalidad,* $\mathbf{Z} = \{X_1, \ldots, X_i\}$ *y* $\mathbf{W} = \{X_{i+1}, \ldots, X_n\}$, *si se cumple que* $I(\mathbf{Z};Y) = I(\mathbf{X};Y)$ *entonces la clase* Y *es condicionalmente independiente de* $\mathbf{W}$ *conocido* $\mathbf{Z}$.

Demostración. Por la propiedad asociativa de la información mutua,

$$I(\mathbf{X};Y) = I(\mathbf{Z},\mathbf{W};Y) = I(\mathbf{Z};Y) + I(\mathbf{W};Y|\mathbf{Z}) \tag{4.4}$$

Como $\mathbf{Z}$ es un SAS se verifica que $I(\mathbf{Z};Y) = I(\mathbf{X};Y)$, lo que implica que el segundo sumando de la expresión (4.4) debe ser nulo, siendo por tanto la información mutua entre la clase y el subconjunto $\mathbf{W}$ nula cuando $\mathbf{Z}$ es conocido, lo cual solo se verifica cuando son condicionalmente independientes. □

El subconjunto de atributos no informativos, se puede ver que coincide con el conjunto de atributos no relevantes según la Definición 4.4. Este subconjunto de atributos no informativos $\mathbf{W}$ puede contener dos tipos de atributos que no aportan información al proceso de aprendizaje:

a) los atributos irrelevantes que son aquellos que no aportan información.

b) los atributos redundantes que son aquellos que aportando información, son completamente dependientes de un subconjunto del subconjunto de atributos suficiente, por lo que toda la información que aportan al proceso de información ya se encuentra contenida en el SAS.

Definición 4.6. *Sea* $\mathbf{I} \subseteq \mathbf{W}$, *se dice que* I *es un conjunto de* ***atributos irrelevantes*** *si la sustitución de cualquier subconjunto de atributos del SAS por* $\mathbf{I}$ *supone una reducción de la cantidad de información al proceso de aprendizaje.*

Definición 4.7. *Dos conjuntos de atributos disjuntos* $\mathbf{R}$, $\mathbf{T}$ *tales que* $\mathbf{R} \subset \mathbf{X}$ *y* $\mathbf{T} \subset \mathbf{X}$ *(*$\mathbf{R} \cap \mathbf{T} = \emptyset$*), se dicen que son* ***redundantes*** *si se cumple que* $I(\mathbf{R};\mathbf{T})$ *es máxima.*

Teorema 4.1. *Si* $\mathbf{R}$ *y* $\mathbf{T}$ *son dos subconjuntos redundantes, toda la información que aporta uno de ellos se encuentra completamente contenida en el otro por lo que el SAS no incluye a uno de los dos.*

Demostración. Sea sin perdida de generalidad $\mathbf{X} = \mathbf{R} \cup \mathbf{T}$, entonces se tiene que:

$$I(\mathbf{X};Y) = I(\mathbf{T},\mathbf{R};Y) = I(\mathbf{T};Y) + I(\mathbf{R};Y|\mathbf{T}) \tag{4.5}$$

Como $\mathbf{T}$ y $\mathbf{R}$ son redundantes se debe cumplir que la información mutua entre ambos sea máxima y por las propiedades de la información mutua,

$$0 \leq I(\mathbf{R};\mathbf{T}) \leq H(\mathbf{R}) \Rightarrow I(\mathbf{R};\mathbf{T}) = H(\mathbf{R})$$

Verificándose entonces que $H(\mathbf{R}|\mathbf{T}) = 0$. Desarrollando la expresión de la información mutua condicional (4.5),

$$I(\mathbf{R};Y|\mathbf{T}) = H(\mathbf{R}|\mathbf{T}) - H(\mathbf{R}|Y,\mathbf{T}) \tag{4.6}$$

Al ser la información mutua condicional siempre positiva $I(\mathbf{R};Y|\mathbf{T}) \geq 0$, al igual que la entropía condicional $H(\mathbf{R}|Y,\mathbf{T}) \geq 0$. Se debe cumplir que para que (4.6) sea nula, la entropía $H(\mathbf{R}|Y,\mathbf{T})$ también lo debe ser, por lo que $I(\mathbf{R};Y|\mathbf{T}) = 0$. Es decir, $I(\mathbf{X};Y) = I(\mathbf{T};Y)$, condición que indica que $\mathbf{T}$ es un SAS y por tanto que solo uno de los dos conjuntos de atributos redundantes forma el subconjunto de atributos suficiente. □

Proposición 4.3. *Si* $\mathbf{R}$ *y* $\mathbf{T}$ *son dos subconjuntos redundantes, se pueden intercambiar en el Subconjunto de Atributos Suficiente, sin que la cantidad de información aportada al proceso de aprendizaje se vea afectada.*

Demostración. Descomponiendo la cantidad de información que aporta el conjunto inicial de atributos $\mathbf{X} = \mathbf{T} \cup \mathbf{R}$,

$$I(\mathbf{X};Y) = I(\mathbf{T},\mathbf{R};Y) = I(\mathbf{T};Y) + I(\mathbf{R};Y|\mathbf{T})$$
$$I(\mathbf{X};Y) = I(\mathbf{R},\mathbf{T};Y) = I(\mathbf{R};Y) + I(\mathbf{T};Y|\mathbf{R})$$

Por lo que siguiendo el mismo desarrollo que en el Teorema 4.1 se obtiene que $I(\mathbf{R};Y) = I(\mathbf{T};Y)$, siendo además $I(\mathbf{R};Y|\mathbf{T}) = I(\mathbf{T};Y|\mathbf{R}) = 0$. Es decir cualquiera

de los dos conjuntos de atributos relevantes puede formar parte del subconjunto de atributos suficiente ya que aportan la misma información. □

En el Capítulo 3 se recogen algunas definiciones de relevancia entre las que se encuentra la de atributo fuertemente relevante (Def. 3.3). A continuación se demuestra que los atributos que componen el subconjunto de atributos suficiente son atributos fuertemente relevantes según la Definición 3.5.

Proposición 4.4. *Los elementos de un subconjunto de atributos suficiente son atributos fuertemente relevantes según la definición 3.5.*

Demostración. La demostración se realizará considerando que un atributo fuertemente relevante no pertenece al SAS y llegando a una contradicción. Sea $\mathbf{Z}$ un SAS del conjunto de atributos $\mathbf{X}$ y $X_i \notin \mathbf{Z}$. Por la definición (Def. 3.3), si X_i es un atributo fuertemente relevante para la clase Y, se cumple que $P(y|\mathbf{z}, x_i) \neq P(y|\mathbf{z})$.

Las entropías condicionales definidas a partir de las anteriores distribuciones de probabilidad condicionales son,

$$H(Y|\mathbf{Z}, X_i) = -\sum\sum\sum P(y, \mathbf{z}, x_i) \log P(y|\mathbf{z}, x_i)$$
$$H(Y|\mathbf{Z}) = -\sum\sum P(y, \mathbf{z}) \log P(y|\mathbf{z})$$

Por las propiedades de la entropía se tiene que $H(Y|\mathbf{Z}) \geq H(Y|\mathbf{Z}, X_i)$ y además por ser X_i fuertemente relevante ($P(y|\mathbf{z}, x_i) \neq P(y|\mathbf{z})$) se debe cumplir que, $H(Y|\mathbf{Z}) > H(Y|\mathbf{Z}, X_i)$. Con lo que la información mutua condicional,

$$I(Y; X_i|\mathbf{Z}) = H(Y|\mathbf{Z}) - H(Y|\mathbf{Z}, X_i) > 0 \tag{4.7}$$

Sin embargo por la definición de SAS se debe cumplir que $I(Y; X_i|\mathbf{Z}) = 0$, que no se cumple por (4.7), por lo que si X_i es fuertemente relevante debe pertenecer al subconjunto de atributos suficiente. □

La definición de subconjunto de atributos suficiente (Def. 4.5) no establece la unicidad del mismo para un conjunto de atributos $\mathbf{X}$, por lo que pueden existir más de un subconjunto de atributos suficiente, aportando todos la misma cantidad de información al proceso de aprendizaje. Resulta pues necesario establecer un criterio para seleccionar el subconjunto de atributos relevantes. En esta tesis se utilizará como criterio de selección, el número de atributos que posee el subconjunto de atributos suficiente, así se considerará como subconjunto de atributos relevantes el subconjunto de atributos

suficiente con menor número de atributos. La utilización de este criterio responde a la preferencia establecida en el Principio de la Cuchilla de Occam por hipótesis o modelos más sencillos. Un elemento importante en la adopción de este criterio es que no establece ninguna relación en cuanto a capacidad de generalización del conjunto de atributos relevantes como establecen algunos autores (Sec. 3.1) ya que se parte de subconjunto de atributos relevantes que desde el punto de vista de Teoría de la Información aportan la misma cantidad de información al proceso de aprendizaje que el conjunto inicial. De acuerdo con el criterio adoptado se establece la siguiente definición,

Definición 4.8. *Se define el* ***Subconjunto de Atributos Relevantes*** *respecto a la clase Y al Subconjunto de Atributos Suficiente con menor número de atributos.*

Por tanto el problema de selección de atributos se plantea como el de la búsqueda del subconjunto de atributos suficiente que posea menor número de atributos, por lo que es necesario computar la información que aportan los posibles subconjuntos como la información mutua entre el subconjunto y la clase.

La utilización de la información mutua como se define en (ec. 4.1) tiene el inconveniente de que las funciones de distribución de probabilidad (f.d.p.) multivariantes que intervienen no siempre se conocen y se deben estimar. Para la estimación de las f.d.p. existen distintas técnicas (Hand, 1986) entre ellas cabe comentar las basadas en histogramas, los métodos basados en núcleos, el método de los vecinos más próximos o los basados en desarrollos en series. El método más sencillo es el basado en histogramas, que viene a ser una extensión del histograma unidimensional, y consiste en dividir el espacio en celdas y estimar la f.d.p. como la proporción de muestras que caen en cada celda. Un problema que aparece con el uso de esta técnica es que cuando aumenta la dimensionalidad, el número de celdas aumenta de forma exponencial. Así la complejidad computacional de esta estimación es de $O(v^n)$ donde v es el número de valores que toman las características (suponiendo todas con igual número de valores) y n el número de características. Este problema es debido a la conocida como Maldición de la Dimensionalidad (*Belman's Curse of Dimensionality*). Aunque hasta ahora se ha comentado el problema de la dimensionalidad para la estimación basada en histograma, este problema es genérico cuando se estiman funciones de probabilidad multivariantes. Así, para el caso de una distribución gaussiana de dimensión d, Comon (Comon, 1995) establece que el número mínimo de muestras N para realizar la estimación basándose en núcleos y con un error menor del 10% es,

$$\log_{10} N \approx 0.6 \left(d - \frac{1}{4} \right)$$

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

Figura 4.3: Problema del or-exclusivo

La aproximación más utilizada para evitar la estimación de f.d.p. multivariables en el cálculo de la información mutua se basa en considerar los atributos como independientes y en consecuencia, computar la información mutua o cualquier otra medida basada en Teoría de la Información como la aportación de cada atributo por separado. El inconveniente que presenta esta aproximación es que en ciertos problemas los atributos tomados independientemente tienen información mutua nula con la clase mientras que en unión de otros sí aportan suficiente información para resolver el problema. Un ejemplo es el conjunto de datos de la Figura 4.3. Si se calcula por separado la información mutua de cada atributo con la clase, se tiene que $I(X_1;Y) = 0$ y $I(X_2;Y) = 0$ con lo que se podría concluir que son atributos irrelevantes para el problema. Sin embargo, calculando la información mutua de los dos atributos conjuntamente con la clase $I(X_1, X_2; Y) = 2$ se puede ver que la información mutua es mayor que cero. Este mismo problema fue identificado por Caruana (Caruana y Freitag, 1994) en la utilización de los algoritmos ID3/C4.5 para la selección de árboles que prefieren atributos que parecen buenos de forma aislada pero que son subóptimos combinados con otros.

Como ya se ha comentado, el cálculo de la información mutua según la expresión (4.1) es computacionalmente costoso. La propuesta que se hace en esta tesis es una medida que permite estimar la información que aporta un conjunto de atributos a la clase teniendo en cuenta las interdependencias entre atributos sin necesidad de estimar funciones de probabilidad multivariables. Antes de exponer esta aproximación se introduce el concepto de matriz de transinformación, necesario en el cálculo de la medida propuesta.

4.3 Matriz de Transinformación

En la sección anterior se vio que la consideración de los atributos como independientes en el cálculo de la información mutua no siempre es correcta, por tanto puede resultar útil disponer de un elemento que recoja las posibles interdependencias que existen entre atributos sin necesidad de estimar funciones de probabilidad multivariables. En esta

sección se introduce el concepto de Matriz de Transinformación cuyo objetivo es precisamente obtener la dependencia de las características dos a dos para luego integrarlas y obtener de esa forma una estimación de las posibles interdependencias que pueden darse en un conjunto de características.

Definición 4.9. *La **Matriz de Transinformación (MdT)** $\mathbf{T} = [t_{i,j}]_{i,j=1...n}$ de un conjunto de n atributos* $\mathbf{X}$ *es una matriz cuadrada de dimensión n donde cada elemento* $t_{i,j}$ *es la información mutua entre los atributos* X_i *y* X_j*.*

$$t_{ij} = I(X_i; X_j) = \sum\sum P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)P(x_j)}$$

Algunas de las propiedades que cumplen los elementos de la MdT $\mathbf{T}$ son las siguientes:

1. La matriz es simétrica: $t_{i,j} = t_{j,i},\ \forall i, j$
2. Todos los elementos de la matriz son no negativos: $t_{i,j} \geq 0,\ \forall i, j$
3. Los elementos de la diagonal principal son mayores o iguales que el resto de los elementos de la fila: $t_{i,i} \geq t_{i,j},\ \forall i, j$

Demostración. La primera y segunda propiedad se demuestran directamente a partir de las propiedades de la información mutua (Sec. 2.2). La demostración de la tercera se puede obtener a partir de las definición de los elementos de la matriz de transinformación,

$$t_{i,i} = I(X_i; X_i) = H(X_i)$$

y por la propiedades de la información mutua $0 \leq t_{i,j} = I(X_i; X_j) \leq H(X_i)$, se demuestra que los elementos de la diagonal principal son siempre mayores o iguales que cualquier elemento de la misma fila o columna. □

El subconjunto de atributos no informativos $\mathbf{W}$ asociado puede estar compuesto por atributos irrelevantes y atributos redundantes. La matriz de transinformación permite detectar los atributos que son redundantes dos a dos de una forma sencilla. La detección de los atributos redundantes se basa en la tercera propiedad de la matriz de transinformación y haciendo uso del siguiente teorema.

Teorema 4.2. *Sea el conjunto de atributos* $\mathbf{X}$ *y su matriz de transinformación asociada* $\mathbf{T}$*, los atributos* X_i *y* X_j *son redundantes si se cumple que* $t_{ii} = t_{ij}$*.*

Demostración. Si $t_{ii} = t_{ij}$ se tiene que $I(X_i; X_i) = H(X_i) = t_{i,i} = t_{i,j} = I(X_i; X_j) = H(X_i) - H(X_i|X_j)$ verificándose que $H(X_i|X_j) = 0$, es decir, las dos características son completamente dependientes. □

Una vez que se han encontrado los atributos que son redundantes y se extraen del conjunto de atributos inicial, en la matriz de transinformación resultante la tercera propiedad se modificaría como,

3. El elemento de la diagonal principal es mayor que el resto de elementos de la fila y columna $t_{i,i} > t_{i,j}$, $\forall j$

Fraser y Swinney (Fraser y Swinney, 1986) indican que la matriz de transinformación MdT puede considerarse como una matriz de correlación generalizada. Esto se basa en la propia definición de los elementos de la matriz de transinformación como la información mutua entre dos atributos, que como se vio en la Sección 2.4 se corresponde con una medida de correlación más general que la lineal, aunque como también se describió en la misma sección, Fraser y Li en diferentes trabajos han establecido la relación entre la correlación lineal y la información mutua para algunos tipos de variables y distribuciones de probabilidad. Una diferencia que existe entre la matriz de correlación y la de transinformación es que todos los valores de la primera se encuentran en el intervalo $[-1, 1]$ a diferencia de la matriz de transinformación. Sin embargo como se expone a continuación, se puede obtener una matriz normalizada donde todos los elementos se encuentren acotados en el intervalo $[0, 1]$.

Definición 4.10. *Se define* ***Información Mutua Normalizada,*** $I^n(X, Y)$, *entre las variables aleatorias* $\mathbf{X}$ *e* $\mathbf{Y}$ *a la relación entre la información mutua y la raíz cuadrada del producto de las entropías de cada variable,*

$$I^n(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}}$$

Proposición 4.5. *La información mutua normalizada* $I^n(X, Y)$ *se encuentran acotada en el intervalo* $[0, 1]$.

Demostración. La demostración de la anterior proposición se realiza partiendo de la expresión de la información mutua normalizada,

$$I^n(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}}$$

Estudiando los casos de dependencia extrema entre las variables X y Y se presentan dos posibles situaciones:

1. Si X y Y son completamente independientes,

$$P(x,y) = P(x)P(y)$$

siendo por tanto la información mutua entre ambos atributos nula,

$$\log \frac{P(x,y)}{P(x)(y)} = 0 \Rightarrow I(X,Y) = 0 \tag{4.8}$$

y haciendo uso de la definición de la información mutua normalizada, se obtiene que $I^n(X,Y) = 0$.

2. Si X y Y son completamente dependientes,

$$P(x|y) = 1 \Rightarrow \left\{ \begin{array}{l} P(x,y) = P(x) \\ P(x,y) = P(y) \end{array} \right\} \Rightarrow P(x) = P(y)$$

y por tanto

$$\begin{aligned} I(X,Y) = H(X) - H(X|Y) &= \sum\sum P(x,y) \log \frac{P(x,y)}{P(x)P(y)} = \\ &= \sum P(x) \log \frac{1}{P(x)} = \\ &= H(X) = H(Y) \end{aligned} \tag{4.9}$$

y sustituyendo lo anterior en la definición de la transinformación normalizada $I^n(X,Y) = 1$.

□

Definición 4.11. *Se define* ***Matriz de Transinformación Normalizada*** *a la matriz* $\mathbf{T^n} = [t^n_{ij}]_{i,j=1...n}$, *cuyos elementos son la información mutua normalizada entre dos atributos* $X_i \in \mathbf{X}$ *y* $X_j \in \mathbf{X}$, $t^n_{i,j} = I^n(X_i, X_j)$.

Se hace notar que la expresión de la matriz de transinformación normalizada guarda una gran similitud con la expresión de la correlación lineal, siendo además magnitudes acotadas que se obtienen a partir de matrices que no tienen sus elementos acotados, como son la de covarianza y la de transinformación.

La utilización de la matriz de transinformación en la medida propuesta para la selección de atributos supone que ésta es definida positiva. A continuación se demuestra que la matriz de transinformación tal y como se definió en la Definición 4.9 cumple con las condiciones necesarias para ser definida positiva y para dimensión 2 cumple también las condiciones suficientes. Primero se demostrará que la matriz de transinformación es una matriz hermítica para luego demostrar que la matriz de transinformación cumple las condiciones necesarias de una matriz hermítica para ser definida positiva y para dimensión 2 también las suficientes.

Proposición 4.6. *La matriz de transinformación es una matriz hermítica.*

Demostración. Por la definición de matriz hermítica (Lancaster y Tismenetsky, 1985; Bronson, 1991) se tiene que una matriz $\mathbf{A}$ es hermítica si $\mathbf{A} = \mathbf{A}^H$, es decir, si es igual a su compleja traspuesta conjugada. Al ser los elementos de la matriz de transinformación $\mathbf{T}$ reales y simétrica, se cumple que $\mathbf{T} = \mathbf{T}^H$ y por tanto es hermítica. □

Las condiciones necesarias que debe cumplir una matriz hermítica, como la matriz de transinformación $\mathbf{T}$ para que sea definida positiva son:

1. Si una matriz hermítica es definida positiva, entonces los elementos de la diagonal principal son todos positivos.
2. Si una matriz hermítica A es definida positiva, entonces para cualquier $i, j = 1 \ldots n$ se verifica que:

$$a_{ii}a_{jj} > |a_{ij}|^2$$

3. Si una matriz hermítica es definida positiva entonces su mayor elemento en valor absoluto debe estar en la diagonal principal

Demostración. La primera condición la cumple la matriz de transinformación ya que los elementos de la diagonal principal se corresponden con la entropía de la característica y ésta es siempre positiva.

La segunda condición se demuestra partiendo de la tercera propiedad enunciada anteriormente para la matriz de transinformación donde se establecía que $t_{ii} > t_{ij}$ de la misma forma que $t_{jj} > t_{ji}$ para todo $i, j = 1 \ldots n$. Al ser $\mathbf{T}$ simétrica, se tiene que,

$$\left.\begin{array}{l} t_{ii} > t_{ij} \\ t_{jj} > t_{ij} \end{array}\right\} \Rightarrow t_{ii}t_{jj} \geq t_{ij}^2$$

Por la propia definición de la matriz de transinformación los elementos de la diagonal principal son mayores que cualquier otro elemento de la misma fila o columna, y además son todos positivos entonces se deduce que el mayor elemento en valor absoluto va a estar en la diagonal principal, por lo que la tercera propiedad queda de esta forma demostrada. □

Las anteriores condiciones son necesarias pero no suficientes. Una condición necesaria y suficiente para que una matriz hermítica, como la de transinformación, sea definida positiva es la siguiente,

4. Una matriz hermítica es definida positiva si y solo si puede ser reducida a una triangular superior con todos los valores positivos, usando la tercera operación elemental sobre fila (añadir una fila multiplicada por un escalar a otra).

Demostración. La demostración de la propiedad anterior se tiene solo para matrices de dimensión 2. Partiendo de la matriz de transinformación,

$$\begin{bmatrix} t11 & t12 \\ t21 & t22 \end{bmatrix}$$

y obteniendo la triangular superior D mediante la suma de la segunda fila a la primera, multiplicada por $\frac{-t_{21}}{t_{11}}$

$$D = \begin{bmatrix} t11 & t12 \\ 0 & t22 + \frac{-t_{21}}{t_{11}} t12 \end{bmatrix}$$

cuyos elementos son todos positivos ya que $\frac{-t_{21}}{t_{11}} < 0$ y $t_{12} < t_{22}$. □

La demostración analítica de la condición necesaria y suficiente no se ha encontrado todavía para dimensiones mayores, aunque en los varios miles de ejecuciones para el desarrollo experimental (Cap. 5) que se han realizado hasta el momento, no se ha encontrado ninguna matriz de transinformación singular.

4.4 Medida GD

En la Sección 2.2 se introdujo el concepto de distancia basada en entropía (Def. 2.6) entre dos variables aleatorias, por tanto, se puede obtener la distancia entre el conjunto

de características $\mathbf{X}$ y la clase Y,

$$d(\mathbf{X}, Y) = H(\mathbf{X}, Y) - I(\mathbf{X}, Y) \tag{4.10}$$

como la diferencia entre la entropía conjunta y la información mutua entre el conjunto de características y la clase. Esta medida, al igual que la información mutua, refleja la dependencia entre el conjunto de características y la clase pero a diferencia de la información mutua, esta distancia[3] es decreciente a medida que la dependencia aumenta (Teorema 2.1). Por ello puede sustituir a la información mutua en la selección de los atributos, con la ventaja añadida de que la máxima dependencia es fácilmente detectable ya que en este caso la distancia es cero, mientras que esta dependencia completa maximiza la información mutua y se corresponde con la entropía, que es diferente para cada conjunto de atributos. Debido a esta dualidad con la información mutua, la distancia basada en entropía ya ha sido utilizada en selección de atributos anteriormente por López de Mántaras (López de Mántaras, 1991) aunque sin tener en cuenta las posibles dependencias entre atributos, ya que su trabajo se encuadraba en la inducción de árboles de decisión donde en cada nodo solo se considera un solo atributo para realizar la partición, por lo que se calcula la distancia entre un atributo y la clase.

La distancia entre el conjunto de atributos y la clase como se define en (ec. 4.10) se puede expresar incrementalmente de forma análoga a la información mutua (ec. 4.3). Definiendo $\mathbf{X}^{(n} = \{X_1, \ldots, X_n\}$ y $\mathbf{X}^{(n-1} = \{X_1, \ldots, X_{n-1}\}$, se puede poner $D(\mathbf{X}^{(n}, Y)$ como,

$$D(\mathbf{X}^{(n}, Y) = D(\mathbf{X}^{(n-1}, Y) + D(X_n, Y|\mathbf{X}^{(n-1})$$

siendo $D(X_n, Y|\mathbf{X}^{(n-1})$,

$$D(X_n, Y|\mathbf{X}^{(n-1}) = H(X_n, Y|\mathbf{X}^{(n-1}) - I(X_n, Y|\mathbf{X}^{(n-1})$$

Expresando $D(X_n, Y|\mathbf{X}^{(n-1})$ en función de $D(X_{n-1}, Y|\mathbf{X}^{(n-2})$ y así sucesivamente se obtiene,

$$D(\mathbf{X}^{(n}, Y) = D(X_1, Y) + \sum_{j=2}^{n} D(X_j; Y|\mathbf{X}^{(j-1}) \tag{4.11}$$

siendo $\mathbf{X}^{(1} = \{X_1\}$, $\mathbf{X}^{(2} = \{X_1, X_2\}$, ...

Una consecuencia que se extrae directamente de la expresión recursiva (ec. 4.11)

[3]En este capítulo cuando se hace referencia a distancia se refiere a la distancia basada en entropía

es la naturaleza creciente con la dimensionalidad de la distancia, porque como se muestra a continuación $D(X_j; Y|\mathbf{X}^{(j-1)})$ es siempre no negativa.

$$\begin{aligned} D(X_j; Y|\mathbf{X}^{(j-1)}) &= H(X_j; Y|\mathbf{X}^{(j-1)}) - I(X_j; Y|\mathbf{X}^{(j-1)}) \\ &= H(X_j; Y|\mathbf{X}^{(j-1)}) - H(X_j|\mathbf{X}^{(j-1)}) + H(X_j|Y, \mathbf{X}^{(j-1)}) \\ &= 2H(X_j, \mathbf{X}^{(j-1)}, Y) - H(X_j, \mathbf{X}^{(j-1)}) - H(\mathbf{X}^{(j-1)}, Y) \geq 0 \end{aligned}$$

ya que

$$\begin{aligned} H(X_j, \mathbf{X}^{(j-1)}, Y) &\geq H(X_j, \mathbf{X}^{(j-1)}) \\ H(X_j, \mathbf{X}^{(j-1)}, Y) &\geq H(\mathbf{X}^{(j-1)}, Y) \end{aligned}$$

Al existir una relación entre la información mutua y la distancia basada en entropía se puede obtener un conjunto dual de definiciones con las basadas en información mutua. Así se tiene que,

Definición 4.12. *Un atributo X_i se dice* ***no relevante*** *respecto a la clase Y si se cumple que $D(X_i, Y) = H(X_i) + H(Y)$*

Definición 4.13. *Un atributo X_i se dice* ***relevante*** *respecto a la clase Y si se cumple que $D(X_i, Y) < H(X_i) + H(Y)$*

Dos atributos X_i y X_j relevantes con respecto a la clase Y, se tiene que X_i es más relevante que X_j si $D(X_i, Y) < D(X_j, Y)$.

Si en lugar de considerar los atributos de forma aislada, se consideran formando parte de un conjunto de atributos, se pueden establecer las siguientes definiciones de relevancia. Así sea $\mathbf{Z}$ un conjunto de atributos tal que $\mathbf{Z} \subseteq \mathbf{X}$.

Definición 4.14. *Un atributo $X_i \notin \mathbf{Z}$ es* ***no relevante respecto al conjunto*** $\mathbf{Z}$ *en la definición de Y si se cumple que:*

$$D(\mathbf{Z} \cup X_i, Y) = H(X_i, Y|\mathbf{Z})$$

Definición 4.15. *Un atributo $X_i \notin \mathbf{Z}$ es* ***relevante conjuntamente con*** $\mathbf{Z}$ *en la definición de Y si se cumple que:*

$$D(\mathbf{Z} \cup X_i, Y) < H(X_i, Y|\mathbf{Z})$$

Es decir, si un atributo es relevante conjuntamente con un conjunto $\mathbf{Z}$ entonces la incertidumbre (entropía) sobre dicho atributo y la clase cuando se conoce el conjunto $\mathbf{Z}$

se reducirá en la misma cantidad que la información (información mutua) que aporta a la clase. Así el atributo X_i será más relevante que el atributo X_j cuanto más decremente la incertidumbre, que equivale a $D(\mathbf{Z} \cup X_i, Y) < D(\mathbf{Z} \cup X_j, Y)$.

Por tanto haciendo uso de la Definición 4.15, *el proceso de búsqueda del subconjunto de atributos más relevantes con cardinalidad l se traduce en obtener el subconjunto de atributos de cardinalidad l con menor distancia con la clase.*

Un inconveniente de la utilización de las expresiones de la distancia (4.10) o (4.11) en la búsqueda del subconjunto de atributos relevantes, es que implican la estimación de funciones de probabilidad multivariantes tanto para el cálculo de la información mutua como de la entropía conjunta, lo que la hace poco práctica debido al alto costo computacional que esto supone. Una primera aproximación para estimar el valor de la distancia (4.10), es la utilizada en varios trabajos (Cap. 3) cuando aparecen problemas similares, y que consiste en considerar las diferentes características por separado y computar la distancia del conjunto de características $\mathbf{X}$ como una combinación de las distancias de cada una de ellas. Como regla de combinación de las diferentes distancias se utiliza alguna norma vectorial como la de orden 2 de Hölder, L_2, es decir,

$$\begin{aligned} \tilde{d}(\mathbf{X}, Y) = \|D_{\mathbf{X}}\|_2 &= \sqrt{\mathbf{D}_{\mathbf{X}}^t \mathbf{D}_{\mathbf{X}}} \\ &= \sqrt{d(X_1, Y)^2 + d(X_2, Y)^2 + \cdots + d(X_n, Y)^2} \end{aligned} \tag{4.12}$$

donde $\mathbf{D}_{\mathbf{X}}$ es un vector cuyos elementos se corresponden con la distancia entre cada característica y la clase,

$$\mathbf{D}_{\mathbf{X}} = \begin{pmatrix} d(X_1, Y) \\ d(X_2, Y) \\ \vdots \\ d(X_n, Y) \end{pmatrix}$$

La estimación de la distancia usada en (4.12) será mejor cuantas menos interdependencias existan entre las características ya que las dependencias no se tomarían en cuenta dicha estimación.

Una posibilidad para recoger las interdependencias de las características es la utilización de la matriz de transinformación definida en la sección anterior, como un elemento que recoge las dependencias de los atributos dos a dos, de forma que sin estimar las distribuciones de probabilidad multivariantes se obtiene una medida de las posibles interdependencias existentes en el conjunto de características. Por tanto es necesario

modificar la expresión (ec. 4.12) para la estimación de la distancia basada en entropía de forma que incluya la matriz de transinformación del conjunto de características. Esta expresión debe ser tal que la aparición de dependencias entre los atributos suponga un decremento en la estimación de la distancia y por tanto se favorezca aquellas combinaciones de atributos interdependientes permitiendo de esta forma incluir bloques de atributos interrelacionados.

Un problema similar al anterior de la inclusión de dependencias para ponderar distancia euclídea entre elementos ya ha sido tratado en Análisis Multivariante y Reconocimiento de Formas en los métodos de clasificación basados en distancia. En Reconocimiento de Formas la distancia utilizada es la distancia euclídea y la ponderación se realiza por medio de la matriz de covarianza, dando lugar a la denominada distancia generalizada de Mahalanobis (Duda y Hart, 1973), cuya expresión para dos puntos $\mathbf{A}$ y $\mathbf{B}$ y una matriz de covarianza Σ es la siguiente,

$$d_{\Sigma}(A,B) = \sqrt{\mathbf{d}(\mathbf{A},\mathbf{B})^t \Sigma^{-1} \mathbf{d}(\mathbf{A},\mathbf{B})} \tag{4.13}$$

La distancia euclídea y la matriz de covarianza en la expresión anterior guardan gran similitud conceptual con la distancia basada en entropía y la matriz de transinformación. La utilización de las medidas de distancia, como la euclídea, en la clasificación por distancia en Reconocimiento de Formas se basan en la consideración de estas medidas de distancia como medidas de disimilitud entre muestras, así muestras pertenecientes a la misma forma son similares y por tanto se encuentran a poca distancia en el espacio de características utilizado, mientras que muestras de formas diferentes se encontraran alejadas. Esta interpretación de la medida de distancia euclídea en Reconocimiento de Formas es similar al concepto de distancia basado en entropía para variables aleatorias, donde una gran dependencia entre ambas supone una distancia pequeña mientras que una independencia supone un valor mayor.

En cuanto a la matriz de covarianza, viene a recoger una información análoga a la de la matriz de transinformación, es decir, dependencia de los atributos dos a dos, siendo la distancia de Mahalanobis más pequeña cuanto mayor es la correlación lineal de las características que definen las formas, aunque en el caso de la matriz de transinformación la interrelación que se recoge por las variables puede ser de mayor complejidad que la lineal como se indicó en la Sección 4.3.

Por tanto una forma de introducir la matriz de transinformación en la expresión (ec. 4.12) es en similitud a como se realiza con la matriz de covarianza en la expresión de la distancia de Mahalanobis. De esta forma definimos la **Medida GD** como,

Definición 4.16. *Se define la* ***Medida GD***, $d_{GD}(\mathbf{X}, Y)$, *entre el conjunto de atributos* $\mathbf{X}$ *con matriz de Transinformación asociada* $\mathbf{T}$, *y la clase* Y *como,*

$$d_{GD}(\mathbf{X}, Y) = \mathbf{D}_{\mathbf{X}}^{t}\mathbf{T}^{-1}\mathbf{D}_{\mathbf{X}} \tag{4.14}$$

La expresión anterior (ec. 4.14) para conjuntos de atributos de dimensión 2 se comporta como un funcional de distancia cuadrático y por tanto cumple las propiedades de una medida de distancia entre las que se encuentra que es creciente con la dimensionalidad (Cuadras Avellana, 1981). Esto es así porque si $\mathbf{f}(\mathbf{a}, \mathbf{b})$ es un vector de las distancias entre elementos de los vectores $\mathbf{a}$ y $\mathbf{b}$ y $\mathbf{M}$ es una matriz definida positiva, entonces la expresión $\mathbf{f}(\mathbf{a}, \mathbf{b})^{t} M \mathbf{f}(\mathbf{a}, \mathbf{b})$ es un funcional de distancia cuadrático (Spath, 1980, pág. 18). Para conjuntos de atributos de dimensión mayor, no se puede afirmar que la medida GD sea un funcional de distancia ya que la demostración de la no singularidad de la matriz de no se ha encontrado para dimensión mayor de 2 como se comentó anteriormente. Sin embargo en todos los experimentos realizados la medida GD ha presentado siempre un comportamiento creciente con la dimensionalidad.

Dicho comportamiento supone que una vez eliminados los atributos completamente redundantes haciendo uso de la matriz de transinformación (Teorema 4.2), el subconjunto con menor valor de la medida GD va a ser siempre el que contiene un solo atributo. Por tanto en el proceso de selección es necesario indicar el número de atributos m a seleccionar de entre el conjunto inicial $\mathbf{X}$. Una vez especificada la cardinalidad, aquel subconjunto con menor valor de la medida GD será el más relevante en la definición de la clase ya que, por la propia definición de la medida, el subconjunto seleccionado incluirá las características más relevantes (menor distancia basada en entropía) y que formen unidades en el sentido de encontrarse interrelacionados entre sí.

La búsqueda del subconjunto de dimensión m puede realizarse mediante el algoritmo "branch and bound" que asegura la obtención del subconjunto con menor valor de la medida GD debido al comportamiento observado de crecimiento con la dimensionalidad. En el Algoritmo 6 se puede ver el procedimiento GD-BB propuesto que realiza la búsqueda empleando la estrategia "branch and bound" implementada con recursividad para la selección del subconjunto de atributos de dimensión m.

Aunque el algoritmo GD-BB asegura la obtención del subconjunto con menor valor de la medida GD para una dimensión dada, en el peor de los casos realiza una búsqueda exhaustiva por lo que para un conjunto de atributos $\mathbf{X}$ con una dimensionalidad alta puede ser bastante costoso. Por ello se introduce el algoritmo GD-SFS (Algoritmo 7) que utiliza una estrategia de búsqueda secuencial hacia adelante comenzando con el

Algoritmo 6 Algoritmo GD-BB

Entrada: m Dimensionalidad del subconjunto resultado.
Entrada: **X** Conjunto de atributos inicial.
Entrada: T Matriz de transinformación asociada a **X**.
Salida: **Z** Subconjunto de atributos seleccionados.

root.num_child=n-m-1
root.feature=X_0 {X_0 no existe, es solo a efectos de inicialización}
$\mathbf{Z}_{temp} = \emptyset$
$\alpha = -\inf$
search_tree(root)

search_tree(node)
if node no es el nodo raíz **then**
  Seleccionar característica asociada a node, $\mathbf{Z}_{temp} = \mathbf{Z}_{temp} \cup node.feature$
  if $d(\mathbf{Z}_{temp}, Y) \geq \alpha$ **then** {Podar todos los nodos por debajo de *node* porque no pueden ser solución}
    retornar
  end if
  if Cardinalidad($\mathbf{Z}_{temp}$) == m **then** {Guardar $\mathbf{Z}_{temp}$ como mejor subconjunto}
    $\mathbf{Z} = \mathbf{Z}_{temp}$
    $alpha = d(\mathbf{Z}, Y)$
    retornar
  end if
  for i=1 to node.num_child **do** {Generar los nodos hijos de node}
    Crear nodo, new_node
    new_node.feature=$X_{node.feature+i}$
    new_node.num_child=node.num_child-i
    Incluir new_node como nodo hijo de node
  end for
  for Todos los nodos hijo de node **do** {Realizar la llamada recursiva para todos los nodos hijos de node}
    search_node(nodo hijo de node)
  end for
end if

Algoritmo 7 Algoritmo GD-SFS

Entrada: m Dimensionalidad del subconjunto resultado.
Entrada: $\mathbf{X}$ Conjunto de atributos inicial.
Entrada: T Matriz de transinformación asociada a $\mathbf{X}$.
Salida: $\mathbf{Z}$ Subconjunto de atributos seleccionados.

```
{Búsqueda de los atributos redundantes}
for i=1 to n-1 do
  for j=i+1 to n do
    if t_ii = t_ij then
      Marcar atributo X_j como atributo redundante
    end if
  end for
end for
Eliminar atributos redundantes de X
{Búsqueda secuencial hacia adelante}
Z = ∅
i=1
while i < m do
  Buscar X_k tal que d_GD(Z ∪ X_k, Y) = min_{X_j ∈ X} {d_GD(Z ∪ X_j, Y)}
  Z = Z ∪ X_k
  X = X\{X_k}
  i=i+1
end while
```

conjunto de atributos vacío hasta alcanzar la dimensionalidad indicada. En los experimentos realizados comparando ambos algoritmos se ha encontrado que los resultados son bastante similares, por lo que debido al menor costo computacional del algoritmo GD-SFS, será este último el empleado en el trabajo experimental que se muestra en el Capítulo 5. Un análisis del coste computacional del algoritmo GD-SFS demuestra que el proceso de búsqueda del subconjunto de atributos más relevantes de dimensión m tiene un coste que es $O(m^2)$, y debido a que el cálculo de la medida GD supone la inversión de la matriz de transinformación que para un subconjunto de dimensión k tiene un coste $O(k^3)$ y el cálculo de la medida que es cuadrática y por tanto $O(k^2)$, se obtiene que el coste del algoritmo GD-SFS es $O(m^5)$, que para conjuntos con un número de atributos grandes ($n > 20$) es bastante menor que la estimación de las funciones de distribución multivariantes que suponen orden exponencial en n.

4.5 Valores Perdidos

Una situación que aparece en muchas bases de datos es la existencia de valores perdidos en algunas muestras. Esto se refiere a que en ellas ciertas características no poseen un valor perteneciente al dominio, bien porque no se pudo obtener o bien porque es erróneo. El problema que existe con estas muestras es cómo hacerlas intervenir en el proceso de selección de atributos.

Si el método de selección de atributos considera las características como independientes, por ejemplo en la aproximación propuesta en (4.12), las muestras con valores perdidos no se hacen intervenir en el proceso de selección cuando se considera la característica con dichos valores perdidos. En el caso antes mencionado esto supone no hacerlas intervenir en la estimaciones de las funciones de probabilidad. Sin embargo, si se están considerando dependencias entre características en el proceso de selección de atributos, entonces la solución anterior no es válida porque ello supone eliminar completamente todas las muestras del proceso de estimación y en consecuencia la perdida de información que aportan esas muestras para las características que si poseen un valor conocido.

Además de la eliminación de las muestras otra propuesta que se ha realizado para la problemática de los valores perdidos es la sustitución del valor perdido por un nuevo valor, incrementando de esta forma el dominio del atributo con valores perdidos. El inconveniente que posee esta aproximación es que la distribución de los valores conocidos se ve modificada y por tanto, la cantidad de información que aporta a la clase o a otro atributo para medir su dependencia entre atributos.

Otra posibilidad, es dar a los valores perdidos el valor promedio o mediana, o el valor más repetido, según se trate de atributos continuos o nominales. Esta aproximación es válida cuando la distribución se aproxime a una distribución normal, pero en casos más generales como se pueden dar en muchas bases de datos de las utilizadas en Aprendizaje Automático, esta aproximación modifica la distribución de probabilidad de los valores que si son conocidos, al introducir un sesgo hacia el atributo promedio o que más se repite.

Una solución más compleja es la inducción del valor perdido de la característica a partir de un árbol de decisión generado considerando la característica con valores perdidos como el concepto a aprender y como entrada el resto de los características. Otra aproximación que solo se puede aplicar sobre el conjunto de aprendizaje es la asignación a los valores perdidos el valor más frecuente que toma dicho atributo para aquellas muestras que pertenecen a la misma clase que la muestra que contiene el valor

perdido.

Desde el punto de vista del clasificador considerado como un canal de información, los atributos con valores perdidos no deben aportar ninguna información ni a la clase ni sobre otros atributos. En el caso extremo de un atributo que no posea ningún valor conocido, será un atributo completamente irrelevante ya que no aporta información. Así pues la sustitución de los valores perdidos tiene que ser tal que cumpla las siguiente condiciones,

1. $I(X_i;Y) \to 0$ cuando el número de valores perdidos de X_i aumenta.
2. $I(X_i;X_j) \to 0 \ \forall X_j \neq X_i$ cuando el número de valores perdidos de X_i aumenta.

De las aproximaciones antes mencionadas, una que permite que las anteriores condiciones se cumpla es la sustitución de los valores perdidos por un mismo valor, por ejemplo la media para atributos continuos o el valor más frecuente en caso de los nominales. Si desarrollamos la expresión de la información mutua para ver como influye la sustitución de los valores perdidos por el mismo valor, se tiene que,

$$I(X_i;Y) = H(X_i) - H(X_i|Y) = \sum P(x_i)\log P(x_i) - \sum\sum P(x_i,y)\log P(x_i|y)$$

Por las propiedades de la entropía (Sec. 2.1) se puede comprobar que a medida que aumenta el número de muestras con valores perdidos para la característica X_i,

$$P(x_i) \to \begin{cases} 1 & \text{si } x_i = \text{valor promedio o más frecuente} \\ 0 & \text{en otro caso} \end{cases}$$

Lo mismo para la distribución condicional $P(x_i|y)$, por lo que se tiene que.

$$\left.\begin{array}{c} H(X_i) \to 0 \\ H(X_i|Y) \to 0 \end{array}\right\} \Rightarrow I(X_i;Y) \to 0$$

El desarrollo anterior es el mismo para $I(X_i;X_j)$ sustituyendo Y por X_j, llegando a las mismas conclusiones por lo que se verifican las condiciones expuestas anteriormente para la sustitución de los valores perdidos, de forma que la información que aportan los atributos es inversamente proporcional al número de valores perdidos de éste. Para comprobarlo se ha realizado un simple experimento que consiste en sustituir diferentes porcentajes de un atributo por la media en una base de datos y observar el compor-

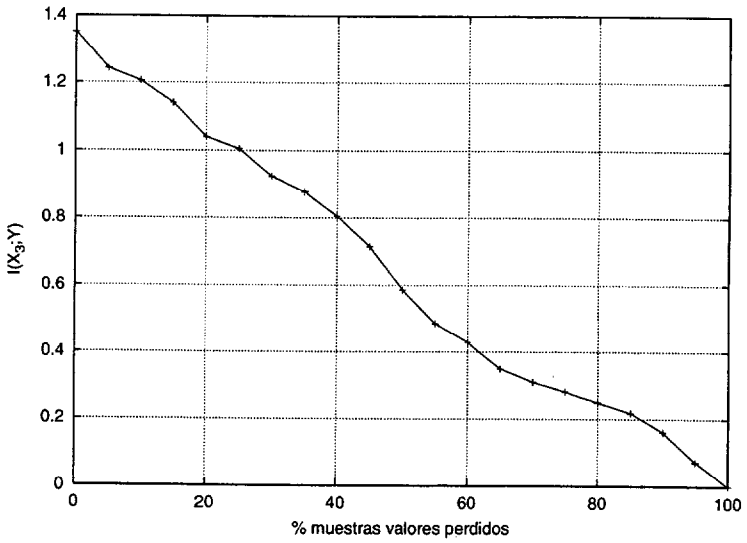


Figura 4.4: Variación de $I(X_3;Y)$ en función del número de muestras con valores perdidos

tamiento de $I(X_i;Y)$ y $I(X_i;X_j)$. La base de datos utilizada es la *Iris*[4] y el atributo utilizado es X_3 (longitud del pétalo). En la Figura 4.4 se muestra la variación de la información mutua entre el atributo X_3 y la clase a medida que aumenta el número de valores perdidos, observándose como $I(X_3;Y)$ decrece a medida que aumenta el número de muestras con valores perdidos. El mismo comportamiento aparece cuando se estudia $I(X_3;X_j)$ excepto para $X_j = X_3$ ya que en este caso la información mutua de un atributo con el mismo es la entropía, y esta entropía aumenta muy ligeramente al principio para luego comenzar a decrecer (Figura 4.5).

[4]Para una descripción de la base de datos ver Sección 5.3

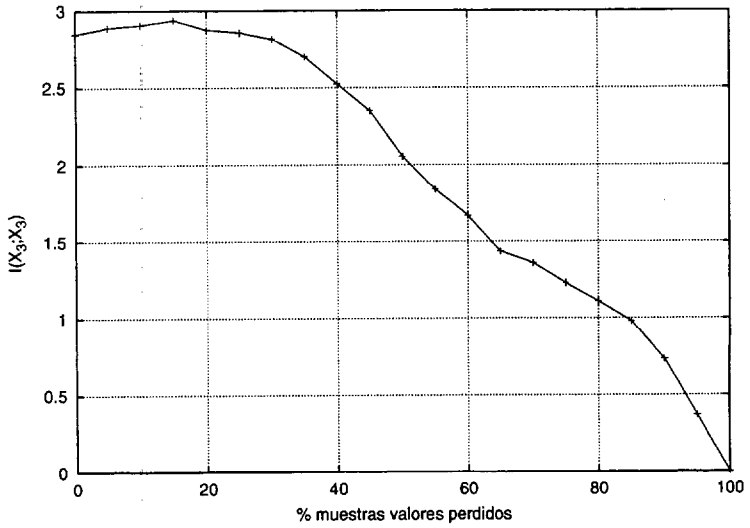


Figura 4.5: Variación de $H(X_3)$ en función del número de muestras con valores perdidos.

Capítulo 5

Evaluaciones Experimentales

Este capítulo recoge los resultados de los experimentos realizados para comprobar el comportamiento de la medida GD en la selección de atributos. Debido a que esta medida está basada en Teoría de la Información, existe la posibilidad que al igual que otras medidas basadas en la misma teoría presente un sesgo dependiente del número de valores de los atributos. Ello hace que el resultado se vea afectado por este factor y no solo por la relevancia del atributo, por lo cual se mostrarán una serie de experimentos para estudiar este sesgo en la medida GD. A continuación se comprobará la bondad de esta medida en la selección de atributos relevantes. Esto se realizará con varios conjuntos de datos, donde la dependencia de la clase con los atributos es conocida a priori y por tanto el resultado correcto también es conocido. Debido a que estos conjuntos de datos no suelen ser reales sino generados sintéticamente, también se estudiará el comportamiento en conjuntos de datos reales con diferentes tipos de atributos y donde la dependencia con la clase no es conocida a priori. Por ello la evaluación de la calidad de los atributos seleccionados se realiza por comparación con los resultados de selección obtenidos con otros métodos, validando estadísticamente los resultados.

5.1 Evaluación del Comportamiento con la Cardinalidad del Conjunto de Atributos

Las medidas basadas en Teoría de la Información suelen presentar un sesgo hacia los atributos que poseen un mayor número de valores. En (White y Liu, 1994) se realiza una comparativa entre las medidas *information gain*, *gain ratio* y la distancia de Mántaras, y medidas basadas en el estadístico de la χ^2 para la selección de atributos en la

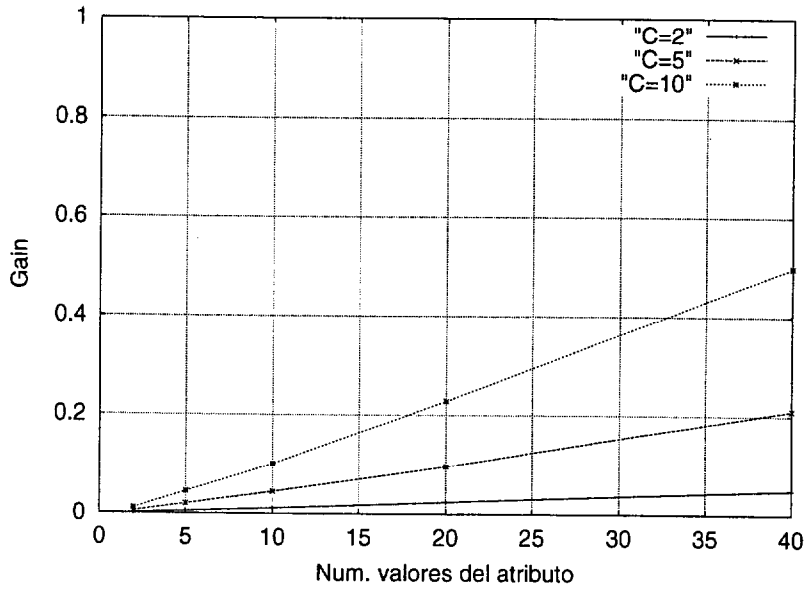


Figura 5.1: Medida Gain para atributos no informativos

inducción de árboles de decisión, demostrándose de forma empírica que las basadas en la χ^2 no presentan un sesgo favorable hacia atributos con un mayor número de valores. Kononenko (Kononenko, 1995) extiende el estudio realizado por White utilizando un mayor número de medidas y otros casos de estudio. La comparativa se basa en una técnica de simulación Monte Carlo donde se utilizan diferentes atributos con distinto número de valores y relación con la clase. Los experimentos se repiten 1000 veces sobre bases de datos con 600 muestras que contiene cada una de ellas 2, 5 y 10 clases con igual número de muestras. Los conjuntos generados se dividen en dos grupos. Un grupo donde los atributos no poseen relación con la clase, ya que la distribución de los valores es uniforme e independiente del valor de la clase. En otro grupo de conjuntos de datos, los valores de los atributos tiene una relación con la clase siguiendo la distribución:

$$P(j \in \{1,\ldots,\lfloor\frac{V}{2}\rfloor\}|i) = \begin{cases} \frac{1}{i+kC} & i \mod 2 = 0 \\ 1-\frac{1}{i+kC} & i \mod 2 \neq 0 \end{cases}$$

donde V es el número de valores del atributo, C el número de clases, i la clase a la que pertenece la muestra y k un parámetro que determina como de informativo es el atributo, en los ejemplos utilizó $k = 1$. Dentro del conjunto $\{1,\ldots,\lfloor\frac{V}{2}\rfloor\}$ el valor del atributo sigue una distribución uniforme.

Hemos reproducido en esta fase esos experimentos para comparar otras medidas con la propuesta GD. A continuación se muestran los resultados para Gain, la distancia de Mántaras y la medida GD. Se han elegido estas medidas para la comparación con la

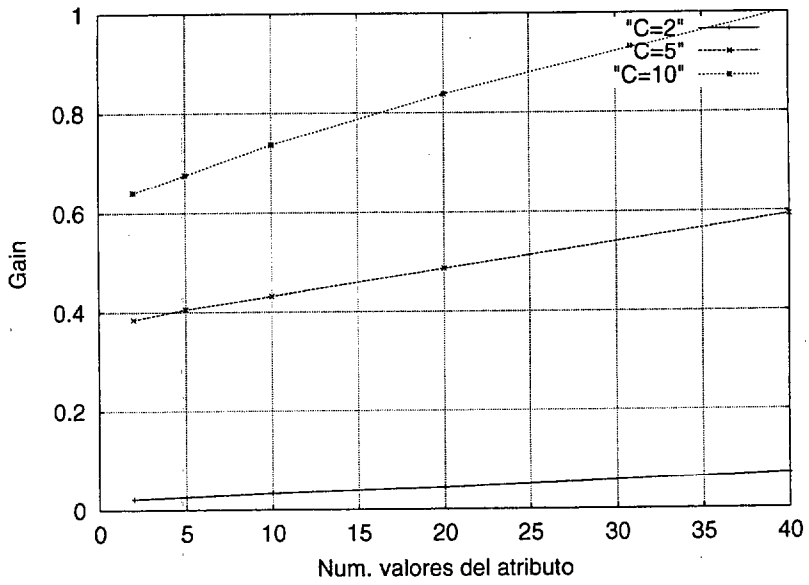


Figura 5.2: Medida Gain para atributos informativos

medida GD porque son aquellas, que al estar basadas en Teoría de la Información, se pueden ver más afectadas por la dimensionalidad del atributo.

En las Figuras 5.1 y 5.2 se muestran los resultados para la medida Gain propuesta por Quinlan, que como se puede apreciar, tiene un comportamiento lineal con el número de valores que puede tomar el atributo. El comportamiento es similar tanto en atributos relevantes como en irrelevantes. La medida Gain Ratio que es una modificación de la medida Gain para evitar el sesgo hacia atributos con un número de valores alto también presenta el mismo comportamiento. La distancia de Mántaras no presenta una dependencia tan lineal como la Gain pero también va aumentando con el número de valores del atributo como se puede ver en la Figuras 5.3 y 5.4.

Los resultados que se obtienen para la medida GD se muestran en las Figuras 5.5 y 5.6. En las gráficas se puede observar que para la medida GD disminuye inicialmente de forma lineal para luego mantenerse casi constante para atributos con un número elevado de valores. El comportamiento descendente de la medida se debe a la propia definición de la misma, ya que cuanto más relevante es un conjunto de atributos menor es el valor de la medida. Por tanto el sesgo hacia atributos con mayor número de valores solo se da para los tres primero atributos, manteniéndose casi constante para situaciones con un número elevado de atributos. Además para los atributos relevantes, el valor de la medida GD es menor indicando ello que son más relevantes, recogiendo por tanto la relevancia de los mismos.

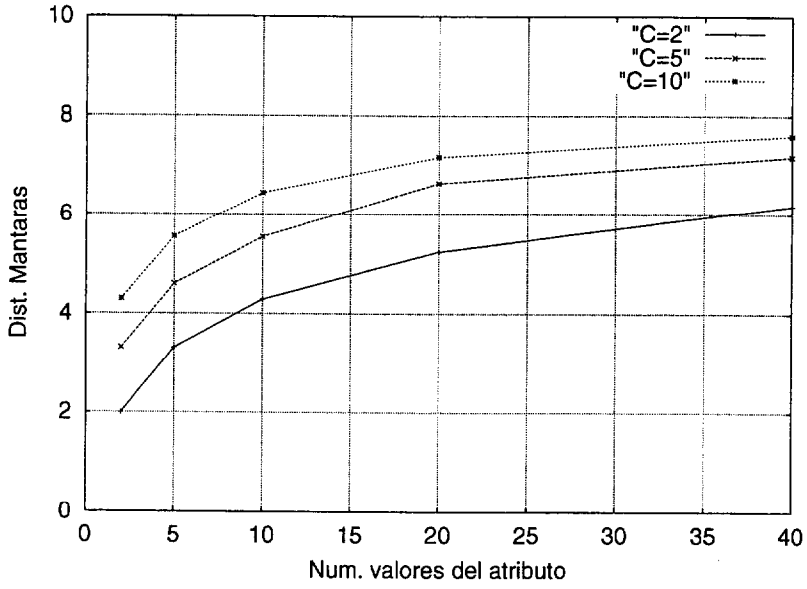


Figura 5.3: Distancia de Mántaras para atributos no informativos

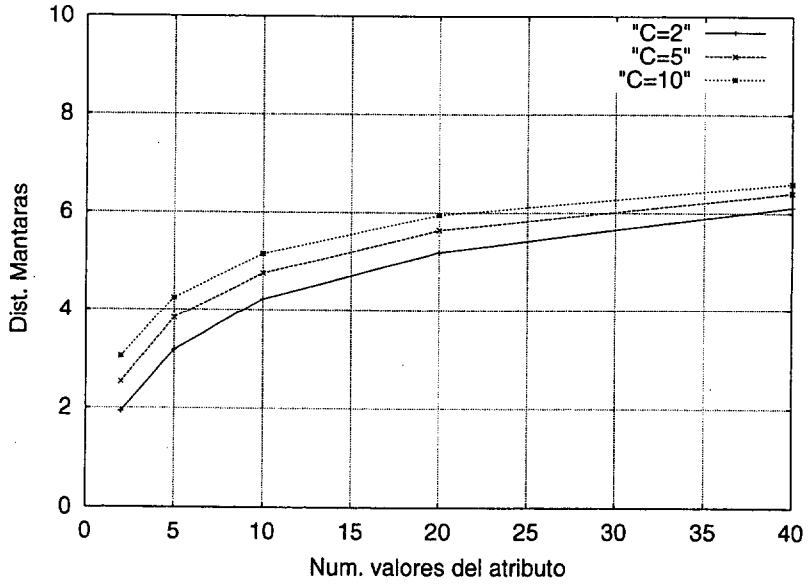


Figura 5.4: Distancia de Mántaras para atributos informativos

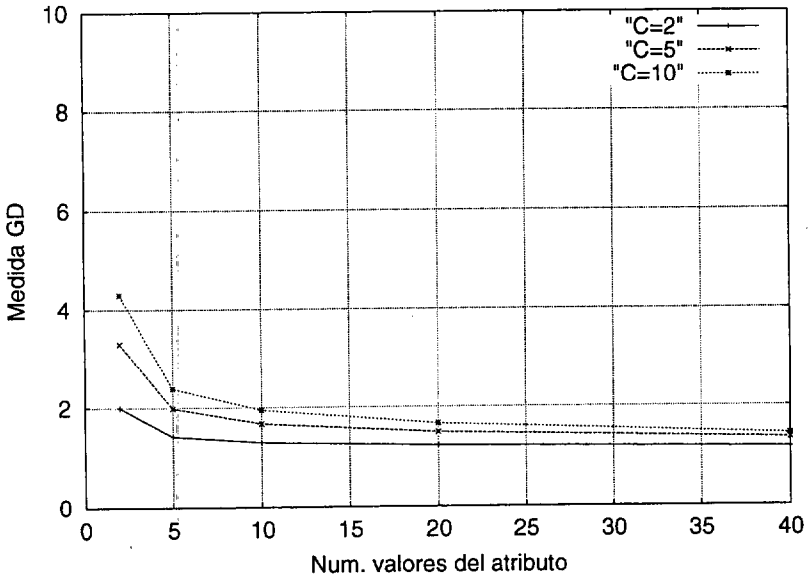


Figura 5.5: Medida GD para atributos no informativos

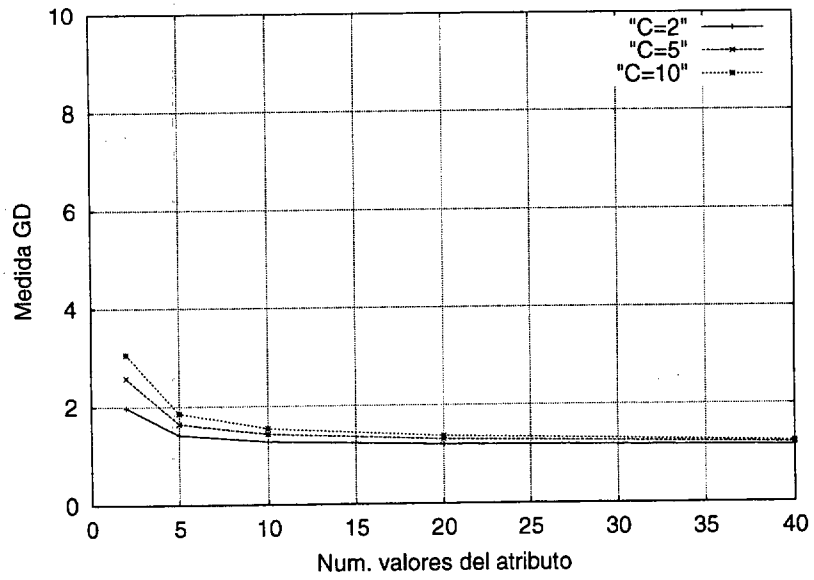


Figura 5.6: Medida GD para atributos informativos

5.2 Estudio Empírico con Bases de Datos Sintéticas

Una vez comprobado el mejor comportamiento de la medida con respecto a la cardinalidad de los conjuntos de atributos en comparación con otras medidas de naturaleza similar también basadas en Teoría de la Información, se estudia el comportamiento de la medida en la selección de atributos en bases de datos de acceso público y utilizadas para estudios comparativos, donde el conjunto de atributos relevantes es conocido a priori. Estas bases de datos son generadas sintéticamente, ya que no pertenecen a ningún problema real donde la dependencia de la clase con los atributos es desconocida.

Las bases de datos sintéticas utilizadas son las siguientes.

CorrAL Esta base de datos fue creada por John (John et al., 1984) y corresponde a un problema biclásico con 6 atributos booleanos $(A0, A1, B1, B0, I, C)$. La clase está definida como ($A0$ AND $A1$) OR ($B0$ AND $B1$), I es un atributo aleatorio y C es un atributo que es igual al valor de la clase en el 75% de los casos.

Led24 Como indica el nombre, es una base de datos donde se codifican con 7 atributos booleanos $(x1, \ldots, x7$, los correspondiente a un display de 7 segmentos) los 10 dígitos numéricos, con 900 muestras. Además de estos 7 atributos que definen la clase, existen otros 17 atributos booleanos $(x8, \ldots, x24)$ que están uniformemente distribuidos y que son completamente irrelevantes.

Monk1, Monk2, Monk3 Estos tres conjuntos se utilizaron como banco de prueba para diferentes algoritmos de aprendizaje (Thrun, 1991). Los tres conjuntos están definidos por 6 atributos distintos, cada uno con un número diferente de valores. Todos los conjuntos contienen 432 muestras.

Atributo	Valores
head shape	round, square, octagon
body shape	round, square, octagon
is smiling	yes, no
holding	sword, balloon, flag
jacket colour	red, yellow, green, blue
has tie	yes, no

Todos los problemas son biclásicos, siendo la definición de las clases las siguientes para cada conjunto.

1. Monk1: (*head shape* = *body shape*) OR (*jacket colour* = *red*)

Base de datos	SFS	SBS
Led24	x1, x2, x3, x4, x5, x6, x7	x1, x2, x3, x4, x5, x6, x7
Monk1	jacket color, head shape, body shape	jacket color, head shape, body shape
Monk2	head shape, body shape, is smiling, holding, jacket color, has tie	head shape, body shape, is smiling, holding, jacket color, has tie
Monk3	body shape, jacket color,	body shape, jacket color,
	holding	holding

Tabla 5.1: Resultados con bases de datos artificiales

2. Monk2: La clase viene definida por todas las muestras donde dos de los atributos toman el primer valor.
3. Monk3: (*jacket colour* = *green* AND *holding* = *sword*) OR (*jacket colour* $\neq$ *blue* AND *body shape* $\neq$ *octogon*), con un 5% de las muestras mal clasificadas.

Parity5+5 Problema de paridad de cinco bits con cinco bits adicionales de ruido. Los bits que definen la clase son *b1, b2, b3, b4, b5* siendo el resto (*b6, b7, b8, b9, b10*) irrelevantes.

Parity5+2 Es una modificación de la base de datos anterior donde los bits *b9* y *b10* son iguales a los bits *b1* y *b2*.

Las pruebas para las bases de datos anteriores se han dividido en dos categorías:

a) En una primera categoría se incluyen *Led24*, *Monk1*, *Monk2* y *Monk3* ya que el conjunto completo de datos se encuentra disponible (Blake y Merz, 1998) y por tanto los resultados son comparables con otras aproximaciones en selección de atributos. Para estas bases de datos se realiza el proceso de selección utilizando las estrategias SFS y SBS y se toma en cada caso los k primeros atributos, siendo k el número de atributos relevantes y que es conocido a priori.

De la Tabla 5.1 se puede concluir que el comportamiento de la medida GD es bueno en el caso de las bases de datos analizadas, ya que en todos los casos selecciona en primer lugar los atributos que definen la clase. Para la base de datos *Monk2* el resultado no es muy significativo ya que todos los atributos son relevantes, sin embargo en *Monk3* que posee un error de clasificación del 5%, éste no afecta al resultado mostrando de esta forma que la medida recoge la dependencia subyacente

Estrategia	A0	A1	B0	B1	I	C
SFS	725	742	757	741	43	1000
SBS	727	737	759	742	43	1000

Tabla 5.2: Resultados para la base de datos CorrAL

en la base de datos sin verse afectada por cierta cantidad de ruido, como ocurre con medidas basadas en la consistencia con el conjunto de aprendizaje.

b) En la segunda categoría se agrupan las bases de datos para las que se dispone de la descripción de la clase en función de los atributos, como son *CorrAL*, *Parity5+5* y *Parity5+2*. Para que los resultados sobre estas bases de datos no se encuentren sesgados por el conjunto utilizado, se generan 1000 conjuntos y para cada uno de ellos se realiza el proceso de selección de atributos y se contabiliza el número de veces en los que un atributo se encuentra entre los k primeros de los seleccionados siendo k, al igual que antes, el número de atributos relevantes conocido por la descripción de la base de datos.

En la Tabla 5.2 se muestran los resultados para la bases de datos CorrAL. Como indica Kohavi (Kohavi y John, 1997, pág. 283) para otros métodos, la medida GD como método Filtro de selección de atributos incluye siempre el atributo C que tiene el mismo valor que la clase en el 75% de las muestras. Sin embargo se observa que en general el atributo irrelevante no es seleccionado y el resto se seleccionan indistintamente, por lo que el conjunto de los cuatro atributos más relevantes incluiría a C y luego tres de los siguiente *A0,A1,B0,B1*. Los resultados que se presentan en (John et al., 1984; Kohavi y John, 1997) sobre esta base de datos utilizando un método Envolvente (Wrapper) y un árbol de decisión como algoritmo de inducción, selecciona correctamente los atributos debido a la naturaleza de la base de datos (atributos booleanos y definición de la clase). Este resultado resulta razonable porque es el método de inducción que mejor se adapta al problema.

Si en lugar de un árbol de decisión se utilizara el clasificador bayesiano simplificado en el método Envolvente, nos podemos encontrar como conjunto de atributos relevantes al problema el mismo que puede dar como resultado la medida GD. Esto es así, ya que la tasa de acierto para el conjunto de aprendizaje usando validación cruzada con 10 particiones es menor para el conjunto $\{A0,A1,B0,B1\}$ (83.68%) que para el conjunto $\{A0,A1,B1,C\}$ (99.11%). Para el conjunto de validación se repite también el comportamiento anterior, siendo mayor la tasa de acierto para el conjunto seleccionado por la medida GD (89.97%) que por el conjunto de atributos que definen la clase (86.08%).

Estrategia	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10
SFS	492	457	509	488	508	518	507	507	511	503
SBS	477	474	503	498	513	504	497	500	526	508

Tabla 5.3: Resultados para la base de datos Parity5+5

Estrategia	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10
SFS	35	24	588	627	610	621	608	606	624	652
SBS	65	56	577	596	609	638	600	597	623	634

Tabla 5.4: Resultados para la base de datos Parity5+2

Además si se compara la tasa de acierto con otro clasificador como el vecino más cercano, se observa que para el conjunto de aprendizaje la tasa de acierto del conjunto de atributos que definen la clase es del 100% mientras que el seleccionado por la medida GD es del 98.44%, sin embargo para el conjunto de validación esta tasa de acierto es muy similar para ambos conjuntos 98.57% y 98.33%. Por ello se puede concluir que la ventaja obtenida con el método Envolvente con respecto a la medida GD en esta base de datos viene influenciada por la elección del árbol de decisión como algoritmo de inducción (bastante adecuado al problema) en el método Envolvente. Este hecho ya fue detectado por Scherf y Brauer (Scherf y Brauer, 1997), indicando que los métodos Envolvente pueden sufrir de sobreajuste a los datos, aparte de que su utilización, debido al alto coste computacional de los mismos, estaría limitada a clasificadores de baja complejidad como son Vecino Más Próximos o Árboles de Decisión.

Las Tablas 5.3 y 5.4 muestran los resultados para las bases de datos *Parity5+5* y *Parity5+2*. En ambos casos se puede ver que la medida GD no realiza correctamente la selección de los atributos que definen la clase definida como un problema de OR-exclusivo. Sin embargo para *Parity5+2* sí se puede ver como los atributos que se encuentran completamente correlados los detecta y no son seleccionados. La imposibilidad de la medida GD para detectar los atributos en estos problemas quizás no sea debido a la definición de la clase de un OR-exclusivo de los atributos, sino al tipo de atributos y la distribución de probabilidades de estos atributos. En las dos bases de datos, los atributos son booleanos y las distribuciones de probabilidad, tanto de los atributos como de las clases, son uniformes.

Tal y como se vió en el Capítulo 2, la entropía es máxima para este tipo de distribuciones y por tanto la incertidumbre también lo es, por lo que la cantidad de información que aportan es mínima. Para confirmar esta afirmación se va a construir una nueva base de datos *Parity5+5* con atributos continuos en lugar de atributos booleanos,

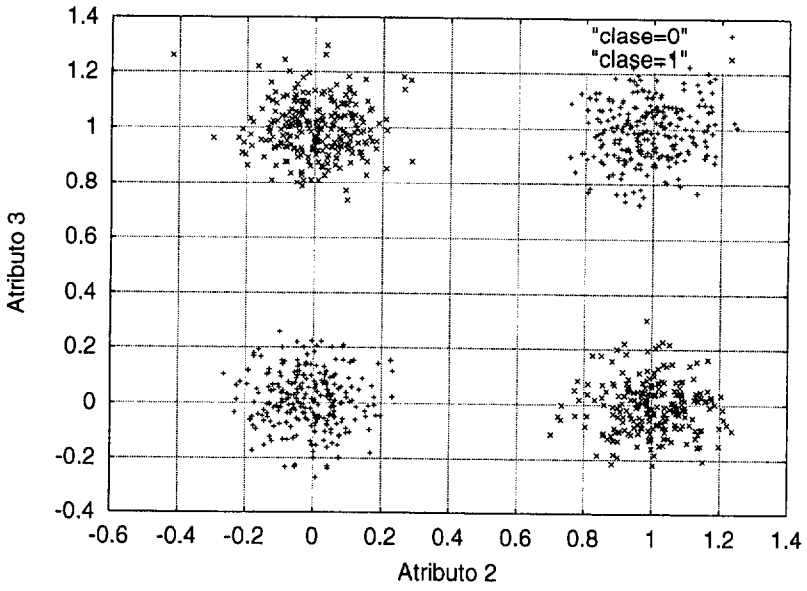


Figura 5.7: Distribución de las clases para el conjunto de datos *Parity2+2* según los atributos 2 y 3

Estrategia	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10
SFS	0	1000	1000	0	1000	0	1000	0	1000	0
SBS	0	1000	1000	0	1000	0	1000	0	1000	0

Tabla 5.5: Resultados para la base de datos Parity5+5 con atributos continuos

de forma que el "1" lógico se convierte en un valor continuo que se distribuye según una normal con media 1.0 y desviación típica 0.1 ($1_{lógico} \equiv N(1, 0.1)$), y el "0" lógico se convierte en un valor continuo distribuido según una normal de media 0.0 y desviación típica 0.1 ($0_{lógico} \equiv N(0.0, 0.1)$), siendo la clase también igual que antes el OR-exclusivo, aunque en este caso los atributos que definen la clase son *b2*, *b3*, *b5*, *b7* y *b9*, el resto de atributo toman valores en el intervalo $[0, 1]$ distribuidos uniformemente. En la Figura 5.7 se puede ver la distribución de las clases en función de los atributos 2 y 3, para la base de datos *Parity2+2* que se define de forma similar a la *Parity5+5* pero con dos atributos relevantes (atributos 2 y 3) y dos distribuidos uniformemente entre 0 y 1.

Repitiendo el experimento que se realizó con la base de datos *Parity5+5* utilizando esta versión con atributos continuos, los resultados que se obtienen son los que se muestran en la Tabla 5.5. Como se puede ver en los 1000 conjuntos generados, los 5 atributos que primero se seleccionan son los que definen la clase, lo cual demuestra que la medida GD puede detectar dependencias del tipo OR-exclusivo siempre que los atributos y las clases no se encuentre distribuidas uniformemente, ya que desde el punto de vista de Teoría de la Información no existe información en estos conjuntos de da-

Estrategia	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10
SFS	479	514	495	484	500	506	515	505	507	495
SBS	503	518	502	482	515	489	516	486	489	500

Tabla 5.6: Resultados para la base de datos Parity5+5 con atributos continuos discretizados en dos intervalos

tos. Se repitieron los experimentos para bases de datos similares a las anteriores para *Parity2+2*, *3+3*, y *4+4*, y en todos los casos la medida GD selecciona primero y correctamente los atributos relevantes, seleccionando los atributos completamente irrelevantes en último lugar.

En la Tabla 5.6 se muestra el resultado del proceso de selección igual al realizado anteriormente con la base de datos *Parity5+5* con valores continuos pero discretizándolos en solo 2 valores discretos en lugar de en 10 valores discretos. Se observa que el resultado es muy similar al que se obtiene con la base de datos original con atributos booleanos. Sin embargo el resultado que se obtiene cuando se discretiza en 3 valores discretos es igual al que se obtiene cuando se discretiza con 10, lo que indica, como se comentó anteriormente, que cuando la distribución que representa los atributos y las clases no es completamente uniforme, la medida GD puede encontrar dependencias del tipo OR-exclusivo.

En general se puede concluir que la medida GD selecciona correctamente los atributos relevantes para diferentes tipos de dependencia de la clase con los atributos, exceptuando la base de datos *CorrAL* que parece está diseñada muy a medida de un método de selección de atributos concreto (Envolvente + árbol de decisión). Por otro lado indicamos que en aquellos casos en los que la Teoría de la Información predice que la información es mínima, como la base de datos *Parity5+5*, la medida GD no puede extraer los atributos relevantes sin tener que ver esto con que la dependencia entre la clase y los atributos sea del tipo OR-exclusivo, como se demostró con una versión con atributos continuos de la misma base de datos. Además se puede observar también que los resultados son iguales para las dos estrategias de búsqueda comparadas, Secuencial hacia Adelante y Atrás, por lo que en el resto de los experimentos solo se expondrán los resultados obtenidos para una de ellas, la búsqueda secuencial hacia adelante.

5.3 Estudio Experimental sobre Bases de Datos Reales

Para las bases de datos artificiales, se pudo comprobar el comportamiento de la medida GD cuando los atributos relevantes son conocidos con anterioridad. El problema de estas bases de datos suele ser que normalmente están definidas para atributos booleanos y las clases se definen como operaciones lógicas de estos atributos, algo que no siempre se puede asegurar en problemas reales, donde la relación entre la clase y los atributos suele ser de naturaleza más compleja y pueden contener muchas veces un conjunto de atributos mixto (atributos nominales, booleanos y continuos).

En este apartado se estudia la bondad de la medida GD en la selección de atributos en bases de datos reales, así como en algunas de las vistas en el apartado anterior. Las bases de datos reales utilizadas son públicas y se han obtenido del almacén de base de datos para Aprendizaje Automático de la Universidad de California Irvine (Blake y Merz, 1998). Una breve descripción de las bases de datos reales utilizadas se da a continuación:

Breast Cancer Ljubljana (BC) Esta base de datos fue una donación del Instituto de Oncología de Ljubljana. Posee 286 muestras distribuidas en dos clases (201 y 86 muestras respectivamente), donde cada muestra esta definida por 9 atributos nominales relativos al paciente al que pertenece la muestra.

Breast Cancer Wisconsin (BW) Esta base de datos al igual que la anterior fue donada por un centro médico, de la Universidad de Wisconsin en este caso (Mangasarian y Wolberg, 1990). Las 699 muestras pertenecientes a dos clases (458 y 241) están definidas por 10 atributos continuos que recogen diferentes medidas sobre las células analizadas.

Credit Card (CR) Base de datos con 15 atributos donde se mezclan atributos continuos y nominales con 690 muestras correspondientes a solicitudes de tarjetas de créditos. Un elemento importante es que los atributos $A4$ y $A5$ están completamente correlados por lo que existe una versión de 14 atributos para ser utilizada por métodos que no se comportan bien con la presencia de atributos correlados, no siendo el caso de la medida GD como se demostró en la Sección 5.2.

Glass (GL) Base de datos que recoge la composición de 7 tipos de cristal diferentes, según la cantidad de 10 componentes químicos. La cantidad de cada componente

se indica como un valor continuo. En la base de datos existen 214 muestras cada clase conteniendo: 70, 17, 76, 0, 13, 9 y 29 muestras.

Glass2 (G2) Esta base de datos es derivada de la anterior donde los 7 tipos de cristal se han agrupado en 2 clases diferentes (163 y 51 muestras respectivamente) según el tipo de procesamiento que recibe el cristal.

Heart Disease (HD) Las 270 muestras que contiene esta base de datos indican la presencia o ausencia de daños en el corazón. Cada muestra está definida por 13 atributos continuos. La distribución de las muestras en las dos clases es de 150 y 120 muestras.

Ionosphere (IO) Es una base de datos con 34 atributos continuos que se corresponden con el expresión compleja de 17 emisiones de radar hacia la ionosfera con diferentes números de pulsos. La base de datos contiene 351 muestras.

Iris (IR) Esta es quizás uno de los conjuntos de datos más referenciados en la literatura tanto de Reconocimiento de Formas como de Aprendizaje Automático. Los datos corresponden al ancho y alto del pétalo y sépalo de tres tipos de lirios.

Led (LE) Esta es la base de datos Led24 utilizada en la Sección 5.2 pero sin la adición de los atributos ruidosos. Van a existir 900 muestras con 7 atributos booleanos que hacen referencia a los 10 dígitos decimales, con la modificación que cada atributo puede tener un 10% de probabilidad de tener el valor complementado, por tanto se ha añadido ruido en los valores de los atributos.

Liver Disorder (LD) Esta base de datos también procede del campo de la medicina, siendo en este caso la identificación de desórdenes en el hígado a partir de 6 atributos continuos que dan diferentes parámetros de las analíticas. Las dos clases poseen 145 y 200 muestras.

Segmentation (SE) Resultados de 19 medidas calculadas sobre las regiones obtenidas a partir de una segmentación manual de imágenes de exterior. Las clases corresponden a 7 tipos diferentes de clases de pixels y de cada clase se han calculado 19 medidas continuas. La base de datos utilizada contiene 210 muestras, 30 por cada clase, aunque existe otra versión de 2100 muestras (300 por clase).

Monk1, Monk2, Monk3 (M1,M2,M3) Estas bases de datos son las mismas que se definieron en la Sección 5.2.

Pima Indian Diabetes (PI) Base de datos que contiene la incidencia de la diabetes en la tribu india Pima. Cada muestra contiene 8 atributos continuos estando dividas las muestras en 2 clases con 500 y 268 muestras, respectivamente.

Post-operative (PO) La tarea de clasificación en esta base de datos consiste en determinar el área de recuperación de un paciente en función de los datos que se recogen en 8 atributos nominales. Existen tres posibles áreas de recuperación con 2, 24 y 64 muestras cada una.

Tic-Tac-Toe (TT) En esta base de datos se codifican las 958 posibles configuraciones finales en el juego del Tic-Tac-Toe. Cada configuración se define por 9 atributos que se corresponden con las 9 casillas de juego pudiendo estar vacía, una "x" o una "o". Las muestras están distribuidas en dos clases.

Voting (VO) El contenido de esta base de datos son las respuestas de 435 individuos (267 demócratas y 168 republicanos) a 16 preguntas de una encuesta donde se mostraban a favor, en contra o no contestaban. La tarea consiste en clasificar a cada individuo como republicano o demócrata según las respuestas a las preguntas formuladas.

Wine (WI) Estos datos son el resultado del análisis químico de tres tipos de vinos italianos. De cada tipo de vino se estudiaban 13 parámetros que se codifican como atributos continuos y en la base de datos existen 59, 71 y 48 muestras de cada tipo de vino respectivamente.

Zoo (ZO) En esta base de datos se dan diferentes características de animales recogidas en 16 atributos booleanos excepto uno nominal. Las características hacen referencia a 7 tipos diferentes de animales de los que existen 41, 20, 5, 13, 4, 8 y 10 ejemplares en la base de datos.

Para estudiar la bondad de la medida GD se compararán los resultados de la selección de atributos con otros dos métodos: la distancia de Mántaras y el método ReliefF utilizando 4 vecinos. El primero ha sido elegido por la similaridad conceptual con la medida GD, extendiendo dicha distancia para un conjunto de atributos mediante la utilización de la norma vectorial de orden 2 (ec. 4.12), mientras que ReliefF ha sido elegido por ser un método bastante referenciado y utilizado en la bibliografía (Caruana y Freitag, 1994; Wettschereck y Dietterich, 1995; Kohavi y John, 1997) y que muestra en general buenos resultados.

La calidad de los atributos seleccionados con cada uno de los métodos se comprueba como la media de la tasa de acierto de tres clasificadores, un árbol de decisión generado con el algoritmo C4.5, IB1 y el clasificador bayesiano simplificado. Como el interés de la comparación radica en estudiar la calidad de los atributos seleccionados se evitó la introducción de sesgos mediante técnicas de optimización en los clasificadores, como poda en el árbol generado o modificación del número de vecinos en el clasificador IB en el que solo se utilizaba un vecino. La implementación de estos algoritmos de inducción se realizó con la librería $\mathcal{MLC}$++ (Kohavi et al., 1996). Para la medida GD y la distancia de Mántaras se discretizaron los atributos continuos en 10 valores dividiendo el rango en 10 intervalos de igual tamaño.

El proceso seguido para comprobar la calidad de los atributos seleccionados por la medida GD fue el siguiente. Para todas las bases de datos se seleccionaban con cada uno de los tres métodos tantos conjuntos como número de atributos tiene cada base de datos, comenzando con un solo atributo, luego dos y así sucesivamente. Para el algoritmo ReliefF los atributos se ordenaban por orden creciente de relevancia, mientras que para la distancia de Mántaras y la medida GD se hacían por orden decreciente, tomando en cada caso el número de atributos del conjunto seleccionado.

Luego se estimaba la tasa de acierto para cada clasificador por separado y cada conjunto y se tomaba como mejor conjunto de atributos aquel para el que la tasa de acierto es mayor. La tasa de acierto se estimaba como la media de 10 ejecuciones de la validación cruzada con 10 particiones. Dietterich (Dietterich, 1998) indica que la utilización de la validación cruzada mediante la técnica antes mencionada es mejor que 5 ejecuciones de validación cruzada con 2 particiones para comparación de métodos, mientras que esta última es más adecuada cuando se intenta estimar la tasa de acierto de un determinado clasificador o método en muestras no utilizadas en el proceso de aprendizaje. Los resultados se muestran en las Tablas 5.7, 5.8 y 5.9.

La validación de los resultados obtenido se ha realizado mediante dos test de hipótesis t pareados con un nivel de significación del 90%. La elección de este test se debe a la utilización del mismo en la comparación de resultados en diferentes trabajos en el área del Aprendizaje Automático (Holte, 1993; Kohavi y Frasca, 1994; Wettschereck y Dietterich, 1995; Liu y Setiono, 1996*c*; Domingos, 1997), que aunque con la validación cruzada no se verifica la independencia entre los conjunto de datos, solo muestra un ligero incremento en el error de Tipo I (Dietterich, 1998).

El primero de los test verifica la diferencia de tasa de acierto obtenida por los dos

	d_{GD}		d_{LM}			*ReliefF*		
	# atr.	% Exact.	# atr.	% Exact.	Concl.	# atr.	% Exact.	Concl.
BC	8	72.61±0.86	7	72.92±0.85	=	8	72.51±0.82	=
BW	1	96.02±0.23	1	96.02±0.23	=	8	96.58±0.20	<
CR	7	85.51±0.41	6	77.77±0.52	>	7	85.51±0.41	=
GL	8	48.08±1.04	3	49.18±1.10	=	3	49.42±1.01	=
G2	4	69.92±1.10	2	61.67±1.10	>	4	65.11±1.10	>
HD	12	85.04±0.62	3	85.15±0.62	=	12	84.26±0.65	>
IO	29	89.46±0.53	32	90.74±0.48	<	31	91.74±0.46	<
IR	2	97.00±0.48	2	97.00±0.48	=	2	97.00±0.48	=
LE	7	74.99±0.44	7	74.99±0.44	=	7	74.99±0.44	=
LD	6	55.58±0.82	6	55.58±0.82	=	1	57.95±0.88	<
M1	1	75.00±0.60	3	75.00±0.60	=	3	75.00±0.60	=
M2	1	67.14±0.74	1	67.14±0.74	=	1	67.14±0.74	=
M3	2	97.22±0.23	4	97.22±0.23	=	2	97.22±0.23	=
PI	6	75.67±0.46	4	75.70±0.47	=	2	76.60±0.50	<
PO	1	70.11±1.43	1	68.45±1.41	>	1	71.11±1.41	<
SE	3	84.09±0.89	10	77.81±0.93	>	10	77.81±0.93	>
TT	6	72.13±0.47	6	73.02±0.44	<	8	72.18±0.45	=
VO	8	95.63±0.28	8	95.63±0.28	=	8	95.63±0.28	=
WI	10	97.53±0.34	3	97.59±0.35	=	4	97.43±0.34	=
ZO	14	93.65±0.74	15	93.75±0.73	=	15	93.75±0.73	=

Tabla 5.7: Resultados para el clasificador Bayesiano Simplificado

conjuntos de atributos.

$$\begin{cases} H_0 : \%Exactitud_{d_{GD}} = \%Exactitud_{ReliefF} \\ H_1 : \%Exactitud_{d_{GD}} \neq \%Exactitud_{ReliefF} \end{cases}$$

$$\begin{cases} H_0 : \%Exactitud_{d_{GD}} = \%Exactitud_{d_{LM}} \\ H_1 : \%Exactitud_{d_{GD}} \neq \%Exactitud_{d_{LM}} \end{cases}$$

Si la hipótesis nula del anterior test es rechazada entonces se efectúa el segundo test de hipótesis, donde se realiza la comparación de las tasas de acierto entre los métodos comparados.

$$\begin{cases} H_0 : \%Exactitud_{d_{GD}} \leq \%Exactitud_{ReliefF} \\ H_1 : \%Exactitud_{d_{GD}} > \%Exactitud_{ReliefF} \end{cases}$$

	d_{GD}		d_{LM}			$ReliefF$		
	# atr.	% Exact.	# atr.	% Exact.	Concl.	# atr.	% Exact.	Concl.
BC	1	72.05±0.77	4	75.45±0.72	<	6	70.79±0.84	=
BW	4	94.61±0.25	6	94.32±0.25	=	4	94.74±0.28	=
CR	7	85.51±0.41	12	84.97±0.39	=	7	85.51±0.41	=
GL	7	69.30±0.94	8	67.36±0.90	>	7	69.30±0.94	=
G2	9	78.76±0.92	5	85.31±0.92	<	3	81.14±0.94	<
HD	10	80.59±0.81	12	79.59±0.72	=	10	81.93±0.67	<
IO	10	90.92±0.50	5	91.88±0.47	<	31	91.54±0.53	=
IR	2	96.50±0.55	2	96.50±0.55	=	2	96.50±0.55	=
LE	7	74.02±0.42	7	74.02±0.42	=	7	74.02±0.42	=
LD	4	64.24±0.84	6	61.92±0.72	>	4	64.24±0.84	=
M1	3	100.00±0.00	5	99.93±0.07	=	3	100.00±0.00	=
M2	5	79.65±0.65	5	79.65±0.65	=	5	79.65±0.65	=
M3	3	100.00±0.00	5	100.00±0.00	=	6	100.00±0.00	=
PI	8	70.60±0.55	8	70.60±0.55	=	7	70.70±0.56	=
PO	1	68.34±1.40	2	69.34±1.40	=	1	71.11±1.41	<
SE	1	88.81±0.73	8	87.91±0.76	>	10	87.19±0.71	>
TT	9	85.61±0.37	9	85.61±0.37	=	9	85.61±0.37	=
VO	8	95.63±0.28	8	95.63±0.28	=	8	95.63±0.28	=
WI	7	95.16±0.50	7	95.16±0.50	=	12	94.10±0.65	>
ZO	7	93.84±0.76	4	93.84±0.76	=	3	93.84±0.78	=

Tabla 5.8: Resultados para árboles de decisión generados con C4.5

$$\begin{cases} H_0 : \%Exactitud_{d_{GD}} \leq \%Exactitud_{d_{LM}} \\ H_1 : \%Exactitud_{d_{GD}} > \%Exactitud_{d_{LM}} \end{cases}$$

En este último test, si se rechaza la hipótesis nula, se puede concluir que el conjunto de atributos seleccionado por la medida GD da mejor tasa de acierto que el seleccionado por la distancia de Mántaras o ReliefF, dependiendo de con cual de ellos se compare. Los resultados de estos test de hipótesis se muestran en las Tablas 5.7, 5.8 y 5.9 bajo la columna etiquetada como *Concl.*.

Considerando todos los posibles resultados con los tres métodos de selección y los tres clasificadores para las 20 bases de datos en estudio se obtienen un total de 120 resultados. La utilización del test de signos como propone Salzberg (Salzberg, 1997) da resultados similares a los que se muestran en las tablas anteriores (Lorenzo et al., 1998*b*; Lorenzo et al., 1998*a*). A continuación se pasan a analizar los resultados obtenidos.

En 21 (17.5%) de los 120 casos, el conjunto de atributos seleccionado por la medida GD da mejor resultado que los seleccionados con los otros dos métodos. En 80 (66.7%), los atributos seleccionados dan una tasa de acierto igual al resto de los métodos. Por último en 19 (15.8%) la tasa de acierto es menor. Por ello si bien la medida GD no

	d_{GD}		d_{LM}			$ReliefF$		
	# atr.	% Exact.	# atr.	% Exact.	Concl.	# atr.	% Exact.	Concl.
BC	4	74.70±0.78	5	75.79±0.79	<	8	73.45±0.76	=
BW	5	95.45±0.24	7	95.44±0.24	=	6	95.84±0.22	<
CR	7	85.51±0.41	12	84.10±0.44	>	7	85.51±0.41	=
GL	6	76.53±0.89	9	68.59±0.99	>	6	76.53±0.89	=
G2	5	87.85±0.73	4	86.68±0.83	=	5	87.85±0.73	=
HD	2	80.78±0.69	11	78.59±0.80	>	9	78.63±0.79	>
IO	10	89.32±0.51	7	90.43±0.47	<	31	90.71±0.46	<
IR	2	96.50±0.54	2	96.50±0.54	=	2	96.50±0.54	=
LE	7	73.67±0.41	7	73.67±0.41	=	7	73.67±0.41	=
LD	5	66.01±0.81	5	66.01±0.81	=	5	66.01±0.81	=
M1	3	100.00±0.00	5	100.00±0.00	=	3	100.00±0.00	=
M2	5	79.93±0.66	5	79.93±0.66	=	5	79.93±0.66	=
M3	3	100.00±0.00	5	99.93±0.07	=	6	98.98±0.19	>
PI	8	70.63±0.44	8	70.63±0.44	=	8	70.63±0.44	=
PO	2	69.67±1.44	3	68.89±1.43	=	1	71.11±1.41	<
SE	1	90.14±0.62	8	88.38±0.65	>	10	86.33±0.71	>
TT	9	98.74±0.12	9	98.74±0.12	=	9	98.74±0.12	=
VO	8	95.63±0.28	8	95.63±0.28	=	8	95.63±0.28	=
WI	2	97.54±0.37	13	98.26±0.32	<	3	96.69±0.36	>
ZO	3	98.11±0.42	4	98.01±0.42	=	4	97.12±0.53	>

Tabla 5.9: Resultados para el clasificador IB1

mejora claramente los resultados con respecto a las otras dos medidas comparadas, si se puede observar de los resultados que para las 20 bases de datos y tres clasificadores analizados, no son peores ya que en la mayoría de los casos (84.2%) se obtienen iguales o mejores tasas de aciertos con los conjuntos de atributos seleccionados por la medida propuesta.

Un asunto importante que debe ser estudiado para los casos en los que el conjunto de atributos seleccionado por la medida GD da como resultado tasas de acierto mayores o iguales es la cardinalidad de los subconjuntos seleccionados, ya que si esta tasa de acierto se obtiene a costa de un mayor número de atributos, la calidad de la medida no será igual que si se obtiene con menor número de atributos. En las Figuras 5.8 y 5.9 se muestran el número de casos en los que la cardinalidad de subconjunto seleccionado por la medida GD es menor, igual o mayor que el seleccionado por la distancia de Mántaras y el algoritmo Relief, respectivamente. De estas figuras se deduce que la medida GD selecciona subconjuntos de atributos que dan como resultado tasas de acierto similares a los otros métodos pero con menor o igual número de atributos.

Analizando los resultados para cada uno de los métodos por separado se tiene que, comparada con la distancia propuesta por López de Mántaras, se obtiene que en

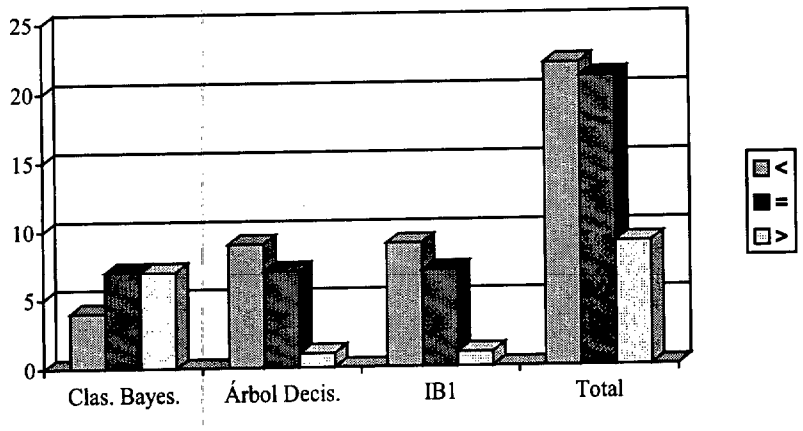


Figura 5.8: Cardinalidad de subconjuntos seleccionados por la medida GD frente a los seleccionados por la distancia de Mántaras para los casos con igual o mayor tasa de acierto.

11 (18.3%), 41 (68.4%) y 8 (13.3%) casos las tasas de acierto resultantes de utilizar los atributos seleccionados con la medida GD son mayores, iguales y menores respectivamente. Mientras que comparada con el algoritmo Relief los resultados muestran que en 11 (18.3%), 39 (65%) y 10 (16.7%) la medida GD selecciona atributos que utilizados en el proceso de clasificación dan mejor, igual o peor tasa de acierto, es decir que posee un comportamiento similar al método ReliefF.

Otro aspecto es el comportamiento de los tres métodos comparados en función de la naturaleza de los atributos de las bases de datos. Éstas se pueden agrupar en tres categorías: solo atributos continuos (BW, GL, G2, HD, IO, IR, LD, PI, SE y WI), atributos nominales y booleanos (BC, LE, M1, M2, M3, PO, TT, VO y ZO) y atributos de diferente naturaleza (CR) en la misma base de datos. Comparando los resultados para cada una de los anteriores grupos de bases de datos, se obtiene que para las bases de datos con atributos continuos en general los atributos seleccionados por la medida GD dan mejores resultados con los clasificadores utilizados que los seleccionados con las otras medidas. En el caso de atributos nominales y booleanos, para tres bases de datos se obtienen mejores tasas de acierto frente a 6 que dan peores tasas de acierto y 45 con igual tasa de acierto. Y por último para la base de datos *Credit Card* los resultados son mejores con la medida GD.

Analizando los resultados obtenidos para las bases de datos con atributos nominales y booleanos más en detalle, se observo que el comportamiento de la medida GD y la propuesta por Mántaras es similar mientras que ReliefF da mejores resultados, debido

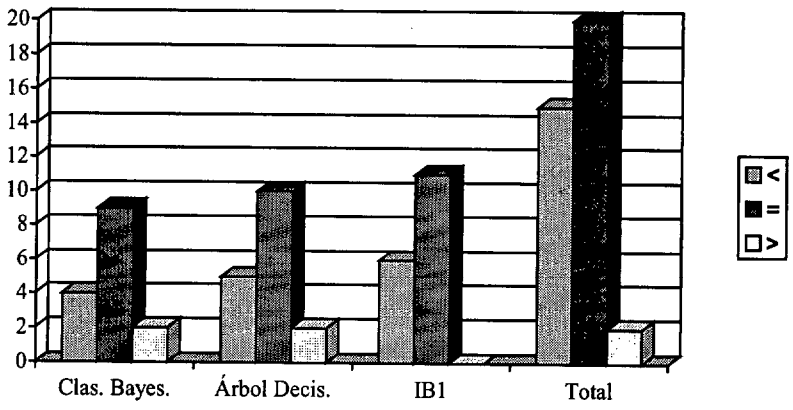


Figura 5.9: Cardinalidad de subconjuntos seleccionados por la medida GD frente a los seleccionados por el método ReliefF para los casos con igual o mayor tasa de acierto.

probablemente al sesgo que introduce los atributos nominales con diferente número de valores en estas medidas, hecho que no se produce en los atributos continuos ya que todos se discretizan en igual número de intervalos discretos.

Después de la evaluación de los resultados a nivel global, nos centraremos en dos bases de datos: *Breast Cancer Wisconsin* y *Credit Card*. En la base de datos *Breast Cancer Wisconsin* existe un atributo que es un identificador único para cada paciente, que posee una capacidad de generalización nula. Este atributo es seleccionado en último lugar tanto por el método Relief como por la medida GD, mientras que la distancia de Mántaras sorprendentemente lo selecciona en primer lugar. En cuanto a la base de datos *Credit Card* ya se comentó que posee los atributos A4 y A5 completamente correlados. Esta correlación entre estos atributos es detectada por la medida GD que selecciona al atributo A5 en último lugar mientras que la distancia de Mántaras y el algoritmo Relief no detectan esta correlación y lo selecciona antes que otros atributos.

Como se puede ver en la Tabla 5.8 donde se muestran los resultados para el árbol de decisión obtenido con C4.5, en general los diferentes métodos de selección de atributos no mejoran significativamente la tasa de error de uno con respecto a otro, debido al método de inducción del árbol de decisión. Por el contrario, la principal ventaja de la utilización de la selección de atributos para la generación de árboles de decisión radica en la reducción del tamaño del árbol generado debido a la presencia de atributos irrelevantes o atributos redundantes. En este aspecto la medida GD no se diferencia del resto de los métodos comparados en cuanto a la tasa de acierto del árbol generado, sin embargo el tamaño del árbol es menor para algunos problemas. En la Tabla

	d_{GD}		d_{LM}			*ReliefF*		
	# atr.	# nodos	# atr.	# nodos	Concl.	# atr.	# nodos	Concl.
BC	1	4.00±0.00	4	41.57±0.16	<	6	214.29±1.45	<
BW	4	69.05±0.47	6	68.98±0.62	=	4	64.09±0.45	>
CR	7	3.00±0.00	12	135.89±0.63	<	7	3.00±0.00	=
GL	7	74.76±0.48	8	73.38±0.48	>	7	74.76±0.48	=
G2	9	39.58±0.34	5	37.88±0.34	>	3	39.76±0.26	=
HD	10	94.68±0.54	12	95.38±0.49	=	10	86.00±0.46	>
IO	10	44.32±0.41	5	44.18±0.41	=	31	53.32±0.42	<
IR	2	8.38±0.10	2	8.38±0.10	=	2	8.38±0.10	=
LD	7	133.20±0.27	7	133.20±0.27	=	7	133.20±0.27	=
LD	4	154.78±0.68	6	133.96±0.80	>	4	154.78±0.68	=
M1	3	41.00±0.00	5	61.44±1.71	<	3	41.00±0.00	=
M2	5	151.55±0.50	5	151.55±0.50	=	5	151.55±0.50	=
M3	3	17.20±0.06	5	17.20±0.06	=	6	17.20±0.06	=
PI	8	244.16±1.14	8	244.16±1.14	=	7	252.06±1.27	<
PO	1	23.00±0.00	2	28.83±0.29	<	1	4.00±0.00	=
SE	1	33.74±0.22	8	34.76±0.26	<	10	35.32±0.29	<
TT	9	225.58±1.80	9	225.58±1.80	=	9	225.58±1.80	=
VO	8	4.00±0.00	8	4.00±0.00	=	8	4.00±0.00	=
WI	7	16.18±0.23	7	16.18±0.23	=	12	14.28±0.23	>
ZO	7	20.98±0.02	4	20.96±0.03	=	3	20.98±0.02	=

Tabla 5.10: Número de nodos del los árboles generados por C4.5 con los atributos seleccionados

5.10, se recoge el número de nodos de los árboles que se generan con los tres métodos de selección de atributos, utilizando para la comparación los mismos test estadísticos que se utilizaron para la tasa de acierto pero en este caso haciendo referencia al número de nodos. Comparando únicamente el número de nodos de los árboles obtenidos para las bases de datos de la Tabla 5.8 en las que la tasa de acierto no es estadísticamente diferente, se observa que en 6 (20.7%) de los casos los árboles inducidos con el conjunto de atributos seleccionados con la medida GD tiene menos nodos que los resultantes de los otros conjuntos de atributos seleccionados, y que en 22 (75.9%) el número de nodos es igual y solo para una base de datos con igual tasa de acierto, el árbol inducido a partir de los atributos seleccionados por la medidas GD es mayor en este caso que los seleccionados con el algoritmo ReliefF.

De los resultados de la comparativa anterior, se puede comprobar que la medida GD y el algoritmo ReliefF dan resultados similares en cuanto a la tasa de acierto que se obtiene con los atributos seleccionados, aunque la medida GD suele dar conjuntos con una menor cardinalidad, aparte de no poseer la limitación de ReliefF con los atributos redundantes, ya que son detectados y seleccionados después que los relevantes no correlacionados. Con respecto a la distancia de Mántaras, que mantiene una gran similitud

con la medida GD, se puede observar de los resultados anteriores que los conjuntos de atributos seleccionados producen tasas de acierto mayores incluso con menor número de atributos, lo que puede explicarse por la introducción de la matriz de Transinformación en la medida GD para recoger la dependencias existentes entre atributos.

Capítulo 6

Aprendizaje de clasificadores en un sistema de visión por computador

En este capítulo se expone una arquitectura funcional para el aprendizaje de clasificadores en el contexto de un Sistema de Visión Basado en Conocimiento (SVBC), donde los dos elementos principales son la medida GD y el módulo encargado de la inducción. La arquitectura se orienta a la abstracción de los clasificadores, lineales y cuadráticos, a partir de muestras seleccionadas por el diseñador en un contexto, en general, borroso. La adquisición y monitorización de muestras se realiza con una herramienta de interacción gráfica. La arquitectura se estructura en un bucle de evaluación global a partir de la calidad del clasificador resultante. Como ejemplo de aplicación de la arquitectura, se exponen los resultados obtenidos en el contexto de dos posibles problemas de Visión Artificial: identificación de elementos en imágenes de exterior y la inducción de un clasificador en un sistema de Visión Activa.

6.1 Introducción

La Visión por Computador es el área dedicada al estudio de teorías y desarrollo de métodos y algoritmos cuyo objetivo final es la obtención de sistemas de percepción visual artificial. Este área ha sufrido una evolución en lo que a metodologías de diseño y desarrollo de dichos sistemas se refiere.

Por un lado, por la consideración de que un Sistema de Visión no es un elemento aislado que se relaciona pasivamente con su entorno, sino que forma un todo indisoluble con éste y que, para su estudio, diseño y desarrollo, es necesario considerar al sistema interactuando con su entorno en un bucle de percepción-acción (Aloimonos y Weis,

1988).

Por otro lado, la evolución se plantea también en la metodología empleada para resolver el compromiso entre flexibilidad y eficacia de los sistemas desarrollados respecto a las modificaciones del entorno. Es un hecho constatado que sistemas muy eficientes que son válidos en la resolución de problemas en ciertos entornos, degradan sus características de comportamiento cuando se producen ligeras variaciones del entorno de trabajo. En este sentido, eficacia y flexibilidad son dos objetivos que muchas veces se contraponen. Es decir, sistemas muy eficaces suelen ser muy específicos, y por tanto costosos de desarrollar y en general su posibilidad de reutilizarlos es baja.

Una forma de resolver este compromiso, facilitando los procesos de diseño y desarrollo y por tanto la interacción diseñador-sistema, se asienta en la segunda tendencia metodológica, relacionada con la integración de diversas tecnologías para el desarrollo de los sistemas, en particular las provenientes de la Inteligencia Artificial de la mano de los Sistemas Basados en Conocimiento (Méndez et al., 1994).

En la Aproximación Basada en Conocimiento se asume que cualquier situación o evento a ser reconocido o detectado puede explicitarse (Vernazza, 1991). Es decir, todos los procesos del sistema de visión, desde los relacionados con los datos sensoriales de entrada hasta los niveles más altos de interpretación involucran un uso intensivo y extensivo de conocimiento. Desde un punto de vista teórico, el uso de estas técnicas no es un requisito estricto, sino más bien *utilitarista*, ya que es un hecho que la representación explícita del conocimiento facilita el proceso de "encaje" de conocimiento específico en el sistema durante el proceso de su desarrollo, así como su verificación, modificación o, más generalmente, reutilización (Clemént y Thonnat, 1993). Asimismo, se facilita la manipulación de heurísticas y la integración de hipótesis diversas.

Un inconveniente que presenta la aproximación basada en conocimiento es el relacionado con la adquisición del conocimiento, la correlación del conocimiento del diseñador con los datos del sistema y en general, todos los procesos relacionados con el aprendizaje (Musen y Van der Lei, 1988; Niemann et al., 1990). Este problema constituye un cuello de botella para los sistemas basados en conocimiento y limitó su éxito más allá incluso del Reconocimiento de Formas. El Aprendizaje Automático ha proporcionado diversas teorías y métodos que intentan solventar este cuello de botella, por lo que muchas de ellas pueden aplicarse en la aproximación de SVBC, donde la naturaleza del problema hace adecuado la utilización del Aprendizaje Supervisado como se verá más adelante.

El aprendizaje en los sistemas de Visión por Computador se puede dividir en nu-

mérico o simbólico dependiendo de tipo de representación utilizado (Bowyer et al., 1994; Nayar y Poggio, 1996). En la primera categoría un trabajo pionero es AVLINN (Pomerleau, 1993), un sistema basado en redes neuronales que aprende a guiar un vehículo en una carretera mediante la observación de un conductor. La entrada al sistema es una imagen de la carretera de baja resolución y la salida es una variable continua que representa el ángulo de giro del volante del vehículo. Otro ejemplo de sistema de aprendizaje numérico es el propuesto por Moghaddam y Pentland (Moghaddam y Pentland, 1995) que se basa en la estimación de las funciones de densidad de un clasificador bayesiano mediante la descomposición en vectores propios (*eigenspace*). Las funciones de densidad estimadas son de dos tipos, una función Gaussiana Multivariante y un Modelo de Mezcla de Gaussianas Multivariantes.

En la arquitectura que se propone, también se utilizan las redes neuronales para extraer el conocimiento de las características numéricas, debido a la falta de conocimiento explícito en este tipo de características. Por cuestiones de simplicidad, se usan diferentes aproximaciones en la resolución de los problemas linealmente separables y no linealmente separables. Para el caso de problemas linealmente separables el clasificador inducido corresponderá a un clasificador lineal que se puede obtener a partir de un Perceptrón de una capa, mientras que para clases no linealmente separables se induce un modelo que corresponde con un conjunto de funciones gaussianas obtenidas a partir de un Red de Funciones de Base Radial. Este último modelo es similar a la Mezcla de Gaussianas utilizado por Moghaddam y Pentland.

Un ejemplo de sistema de visión basado en el aprendizaje simbólico es el propuesto por Pellegretti (Pellegretti et al., 1992) que se basa en el algoritmo de aprendizaje INDUCE. En este sistema la transformación de las características en simbólicas se hace mediante la asignación de una función de pertenencia difusa (FPD) a cada característica. La FPD es trapezoidal y se obtiene mediante un proceso de agrupamiento seguido por un refinamiento de la FPD obtenida. SVEX (Méndez et al., 1994), el SVBC utilizado como entorno para la arquitectura de aprendizaje propuesta, también hace uso de conceptos difusos en el modelo de razonamiento basado en reglas que integra. Por tanto los objetos inducidos incluirán un grado de pertenencia a la clase a la que pertenece.

A continuación se enmarca el contexto del SVBC sobre el que se va a desarrollar la arquitectura propuesta, así como la tipología concreta del problema de aprendizaje a abordar.

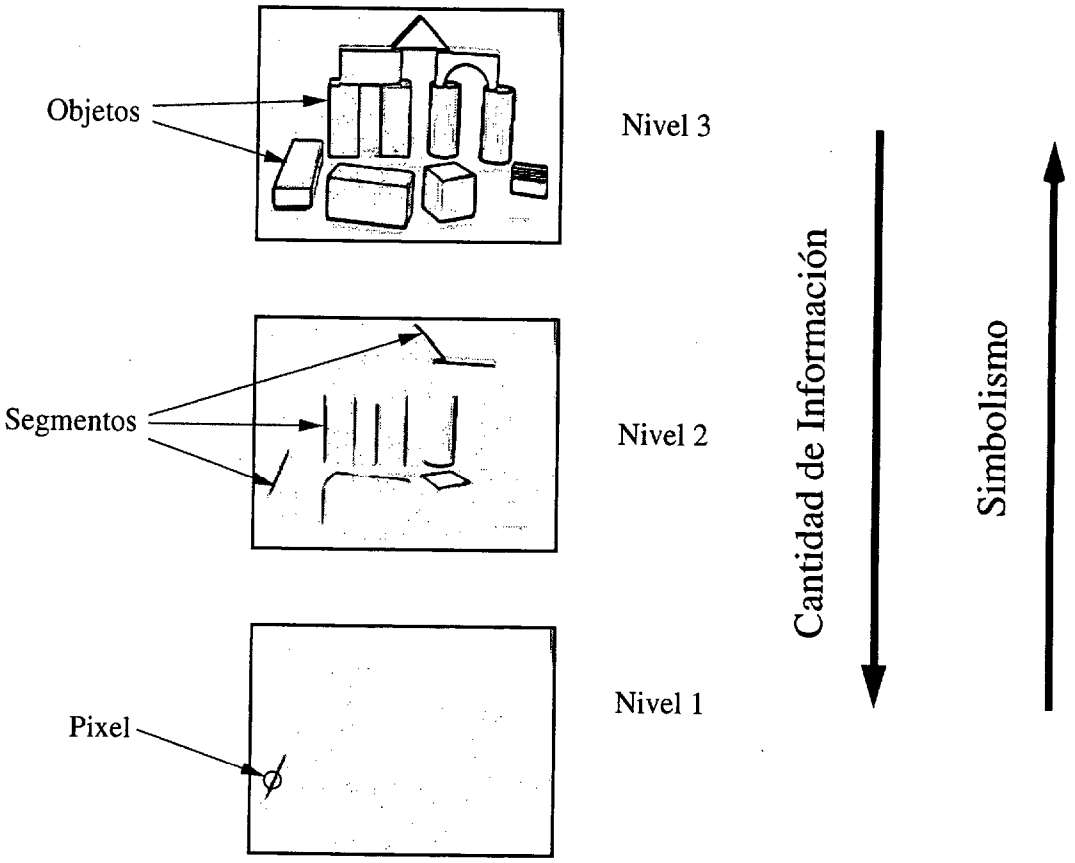


Figura 6.1: Organización por niveles de SVEX

6.2 SVEX: Un Sistema de Visión basado en Conocimiento

SVEX es un SVBC concebido como un sistema percepto-efector estructurado en tres niveles (Fig. 6.1). Cada uno de estos niveles viene definido por la naturaleza del grano de información que se maneja (Méndez et al., 1994; Cabrera, 1995; Hernández et al., 1995). Así, en el nivel inferior, la unidad de información que se maneja es el pixel, en el intermedio es el segmento y en el superior es el objeto. La arquitectura de cada nivel está concebida para que la manipulación del conocimiento resulte claramente explícita, tanto el relacionado con los diferentes aspectos visuales, como con aspectos de control e inferencia (Nandhakumer y Aggarwal, 1985).

En cada nivel coexisten dos dominios, el dominio de las características (numérico) y el dominio de las clases (simbólico) cada uno con sus operadores y datos. En el dominio de las características los operadores son denominados *procedimientos* y calculan las características a partir de los datos de entrada o a partir de otras características previamente computadas. Los datos de entrada en este dominio son casos particulares

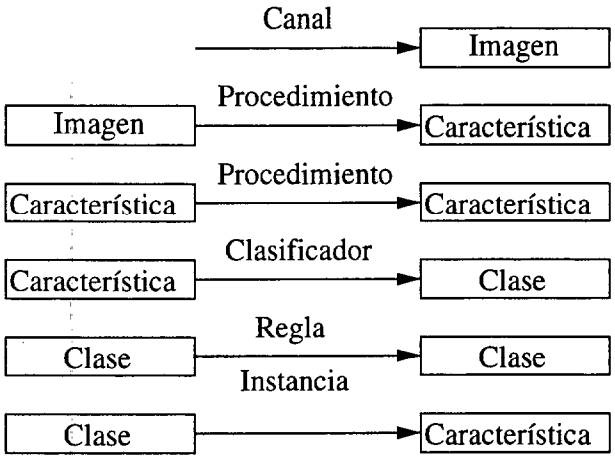


Figura 6.2: Objectos y operadores del Procesador de Pixels

del grano de información correspondiente al nivel, imágenes o mapas de pixels. Los operadores del dominio simbólico son reglas que tienen como entrada clases para generar nuevas clases. Las clases se obtienen a partir de las características mediante la utilización de los clasificadores.

Además se considera la existencia de una máquina virtual para cada nivel denominada respectivamente: Procesador de Pixels, de Segmentos y de Objetos respectivamente, en las cuales las estructuras de datos y los operadores asociados al nivel están orientados a manipular el grano de información correspondiente. Cada una de las máquinas virtuales dispone de un lenguaje de propósito especial y orientado a objeto para la programación de dichas máquinas. En la Figura 6.2 se muestran los diferentes objetos y operadores que están definidos en el procesador de pixels. En estos lenguajes los operadores denominados *procedimientos* son externos y se programan en un lenguaje de propósito general (en la versión utilizada es C) permitiendo la adición de nuevos operadores a medida que son necesarios. En la Figura 6.3 se puede ver la arquitectura del Procesador de Pixels siendo el Procesador de Segmentos (Fig. 6.4) muy similar, con la única diferencia de la adición de un presegmentador para tratar los mapas de pixels suministrados por el procesador de pixels.

Una característica de estas máquinas virtuales es que son orientadas a objetivos, ya que el proceso de computación comienza cuando una clase es solicitada por el nivel superior o por el usuario de la misma. Por tanto, aparece la figura de un elemento en cada procesador denominado *Top-Down* que recibe las peticiones del nivel superior y lanza el proceso de cómputo realizado por el elemento *Bottom-Up* (Fig. 6.3). Un ejemplo de un programa del procesador de pixels se puede ver en la Figura 6.5, donde se pueden ver la instanciación de algunos de los objetos mostrados en la Figura 6.2.

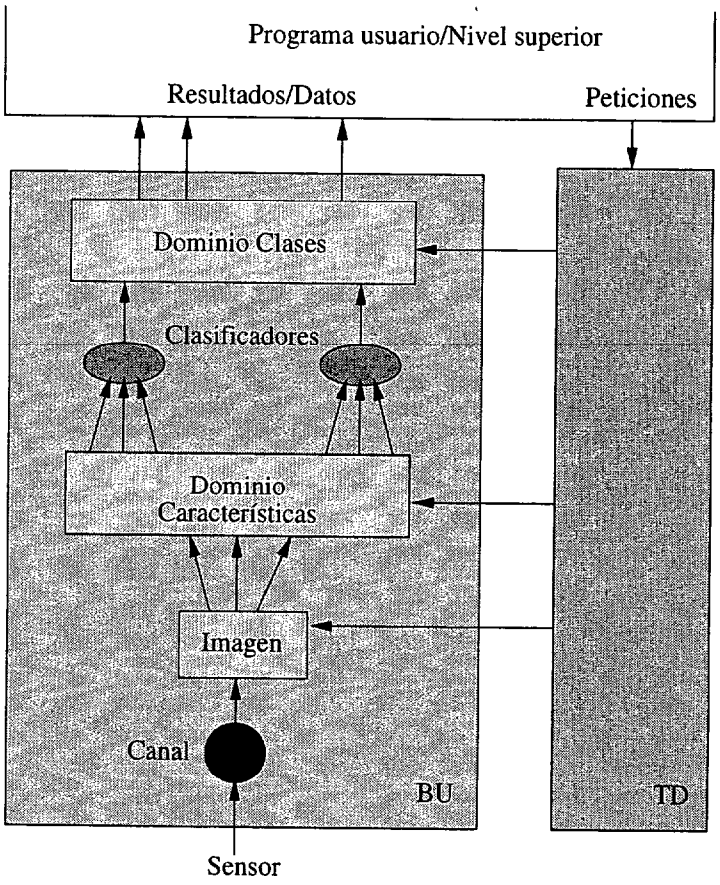


Figura 6.3: Arquitectura del procesador de pixels

6.3 El Problema de la Abstracción

Como se ha comentado anteriormente en la descripción de SVEX, en cada uno de los niveles coexisten dos dominios, un dominio numérico relacionado con las características y un dominio simbólico que se relaciona con las clases. Las características del nivel numérico son los elementos implícitos que aparecen en la unidad de información y que se explicitan por medio de operadores o procedimientos. En el dominio simbólico las clases se asocian a categorías visuales relacionadas con el correspondiente grano de información, y que son parte fundamental de la explicitación del conocimiento acerca de los problemas.

Debido a que la unidad de información se mantiene dentro de cada nivel, el paso del dominio numérico al dominio simbólico consiste en la abstracción, es decir, la asignación a cada ocurrencia de la unidad de información de una categoría o clase. El operador clasificador es el elemento encargado de realizar el paso del dominio numérico al simbólico y es la inducción de los mismos el objetivo de la arquitectura que se propone.

Los problemas de asignación o simbolización en el contexto de los SVBC son un tema importante de estudio. La investigación en categorización humana, es decir, la

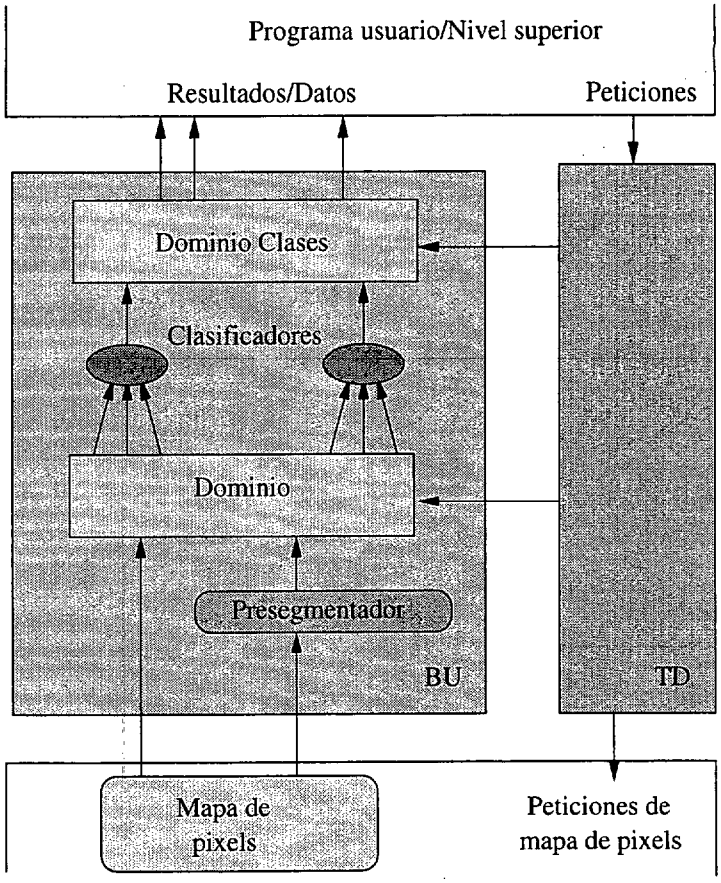


Figura 6.4: Arquitectura del procesador de segmentos

asignación de hechos del mundo a categorías semánticas, efectuada en diversos campos como: ciencia cognitiva, psicología, lingüística, antropología o filosofía, han mostrado que en muchos casos dicha asignación no puede establecerse de forma discreta y dura. En otras palabras, la diferenciación entre los hechos asignados a una categoría y los que no se asignan a ella es un proceso gradual y de contornos no nítidos, existiendo zonas del espacio de hechos con una asignación muy clara en uno u otro sentido, y otras de transición. Esta observación es particularmente aplicable a los aspectos relacionados con las impresiones visuales (Lammens, 1994). Una manera muy adecuada de formalizar este hecho es mediante la utilización de los mecanismos de la Teoría de Conjuntos Borrosos (Zadeh, 1971), en las cuales los grados de pertenencia reflejan de forma numérica la incertidumbre o falta de nitidez mencionada anteriormente. Además, los grados de pertenencia recogen también la forma de las categorías psicológicas (Shepard, 1987), elemento importante al establecer los procesos de adquisición del conocimiento.

El proceso de abstracción en el marco de un SVBC lo efectúan, como se ha dicho, los clasificadores, es decir, mecanismos computacionales cuyo objetivo es la asignación del grado de pertenencia a una clase a cada una de las ocurrencias del grano

```
Define Picture picture
  Channel: CUBEfile
  File :  "../../pictures/a01.Pic"
EndDefine

Define Feature Stadistic
  From: Picture
  Procedure: PicStatistic1
  Parameter: 32;
EndDefine

Define Classifier HighMean
  FeatureList: Stadistic;
  Functional: Unitary
  Decision: Threshold
       Sign: Positive
       Level: 105.0
EndDefine

Define Class HiMeanClass32_75
  Classifier: HighMean;
  Show
  Overlapped: 70
EndDefine

Define Interface
  Class: HiMeanClass32_75;
EndDefine
```

Figura 6.5: Ejemplo de programa en SVEX

de información elemental del nivel correspondiente. La información de entrada a dichos clasificadores serán características a partir de la información aferente al nivel (Fig. 6.6). La arquitectura que se presenta en este capítulo está orientada a la obtención de los clasificadores en el marco del sistema de visión basado en conocimiento SVEX, considerado como un problema de Aprendizaje Automático cuyo objetivo es la asignación de clases semánticas a las diferentes ocurrencias del grano de información en la imagen.

6.4 Arquitectura Propuesta

El problema de la abstracción en un SVBC antes comentado está condicionado por los siguientes elementos:

a) Generalmente no es posible definir explícitamente el clasificador para un cierto problema. Más bien, el conocimiento del problema se refleja en un conjunto finito de ejemplos, es decir muestras consistentes por pares atributo-valor.

b) En general, el conocimiento es escaso tanto en lo referente a qué atributos o ca-

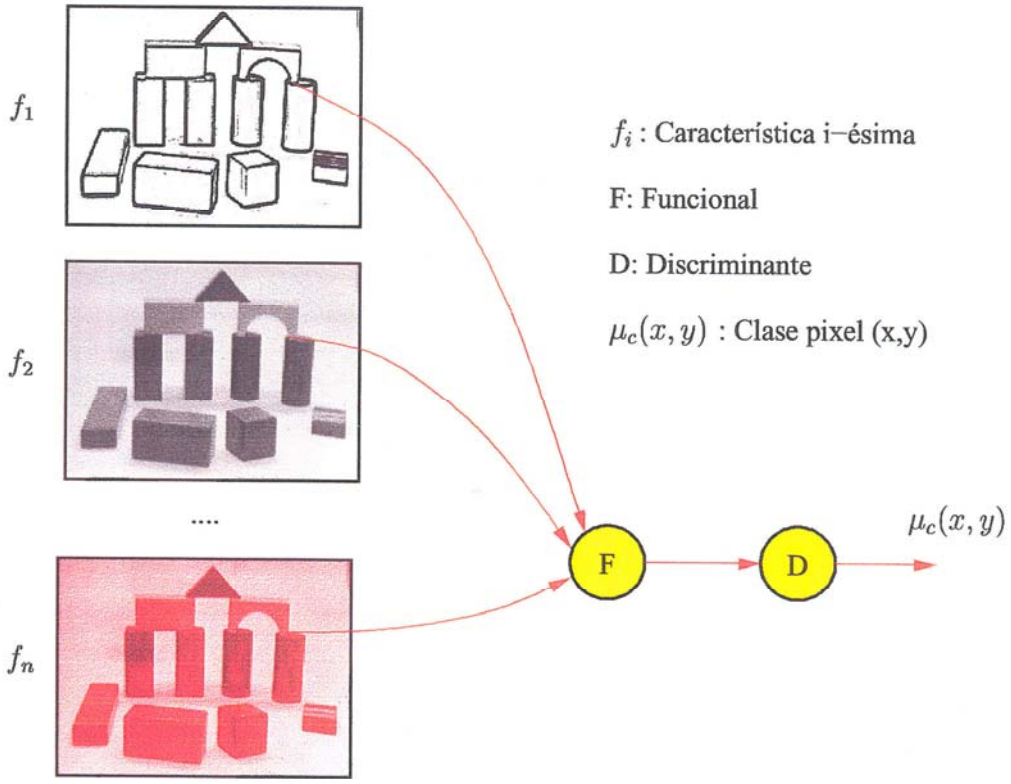


Figura 6.6: Clasificador

racterísticas son relevantes a efectos de la definición de la clase, como a qué pares atributo-valor son significativos a tales efectos.

c) El conocimiento acerca de los problemas de abstracción es limitado y las definiciones de las relaciones entre características y clases, como hemos comentado anteriormente, posee en general incertidumbre.

d) No es adecuado establecer restricciones a priori acerca del comportamiento estadístico de las muestras ni de la distribución de las mismas con respecto a la clase, en el espacio de las características.

Por todo ello, es necesario plantear el proceso de aprendizaje de clasificadores, y por tanto la arquitectura propuesta, atendiendo a los siguientes condicionantes:

1) Definición un mecanismo interactivo, tanto para la adquisición de los ejemplos, como para la evaluación del estado de los procesos y modificación de parámetros y condiciones de los mismos.

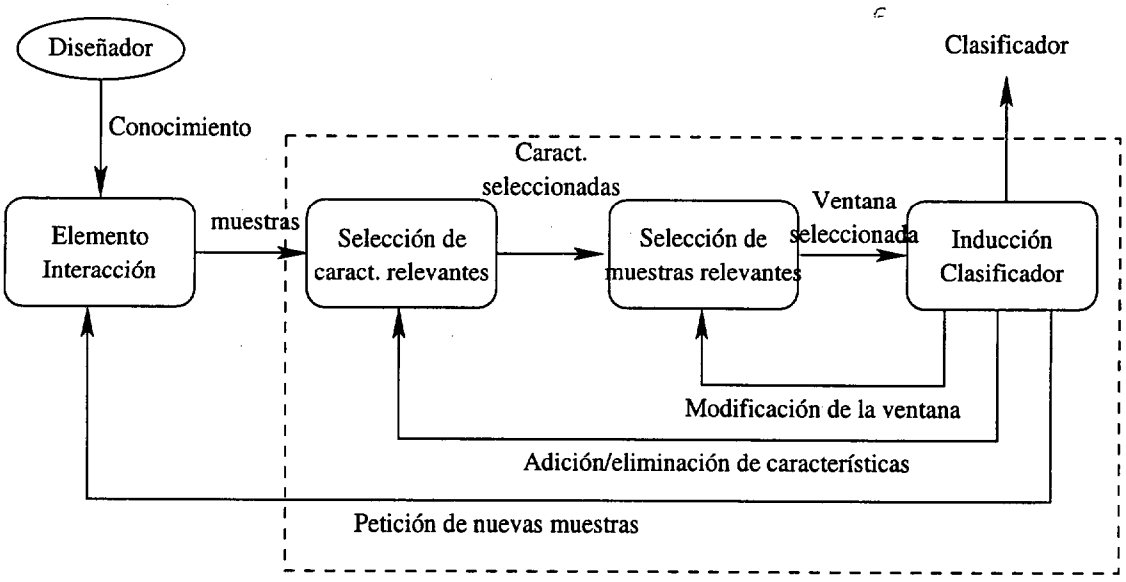


Figura 6.7: Esquema general del aprendizaje de clasificadores

2) Definición de mecanismos para la selección de la información relevante, tanto en lo referente a qué atributos o características son relevantes y cuál es su grado de relevancia comparativo a efecto de los procesos de inducción, como en lo que se refiere a qué muestras o ejemplos son relevantes a esos mismos efectos.

3) Definición del correspondiente mecanismo de inducción de clasificadores, que no suponga restricciones en cuanto a la forma de las regiones de las clases en el espacio de representación.

La manera convencional de resolver problemas relacionados con la abstracción ha consistido en utilizar un conjunto de muestras de aprendizaje seleccionadas a priori y, con ellas efectuar cada uno de los procesos anteriores por separado (Duda y Hart, 1973). Las desventajas de esta aproximación se centran en que el éxito del resultado final del diseño de clasificadores depende de que tanto las muestras como las características seleccionadas inicialmente contengan la información suficientemente relevante para generar una distinción de aceptable calidad entre clases (Bhandaru et al., 1993).

La arquitectura propuesta (Fig. 6.7) recoge los diferentes módulos que resuelven los condicionantes indicados anteriormente. Los módulos que la componen se verán con más detalle en las siguientes secciones y ahora se introducirá el proceso de funcionamiento general. Como ya se ha expresado a lo largo de este capítulo en varias ocasiones, en Visión por Computador no existe un conocimiento explícito por parte de los expertos que permita realizar el procedimiento de aprendizaje basándose sólo en parámetros como la tasa de error, ya que el resultado es dependiente de la aplicación. En función de esta

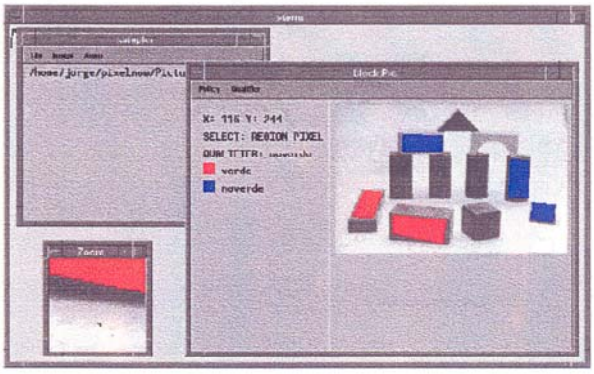


Figura 6.8: Prototipo de la herramienta para la adquisición de muestras

premisa, un elemento importante de la propuesta es el lazo de realimentación que existe, ya que en cada momento el usuario puede ir modificando los diferentes parámetros que intervienen en el proceso.

El ciclo de operación comienza por seleccionar las muestras en las imágenes y la definición del conjunto inicial de características. A continuación se ordenan las características por orden decreciente de relevancia para las clases en estudio y se reduce el conjunto de aprendizaje de forma que solo se utilicen las muestras más relevantes. Por último se obtiene el clasificador y se evalúa su comportamiento en el resto de los pixels de la imagen. Si el resultado no es el adecuado el sistema añadirá nuevas muestras para refinar los resultados o bien se incluirán nuevas características por orden decreciente de relevancia. Este proceso continúa hasta que el resultado se considera satisfactorio por parte del usuario, traduciéndose el clasificador inducido a un programa en SVEX y explicitando de esta forma el conocimiento obtenido. Un elemento importante que se contempla en la arquitectura propuesta es la posibilidad de que el sistema interrogue al usuario acerca de la pertenencia a la clase de algunas muestras que no pertenecen al conjunto de aprendizaje pero que permitirían refinar el resultado del clasificador.

La arquitectura descrita es válida tanto para el procesador de pixels como para el de segmentos ya que en ambos casos el proceso de abstracción es similar. En el procesador de pixels las clases hacen referencia a pixels y las características se obtienen a partir de los pixels y en el procesador de segmentos las clases hacen referencia a segmentos y los procedimientos se aplican a éstos para obtener las características. Particularmente, en el resto del capítulo nos centraremos en la implementación para el procesador de pixels.

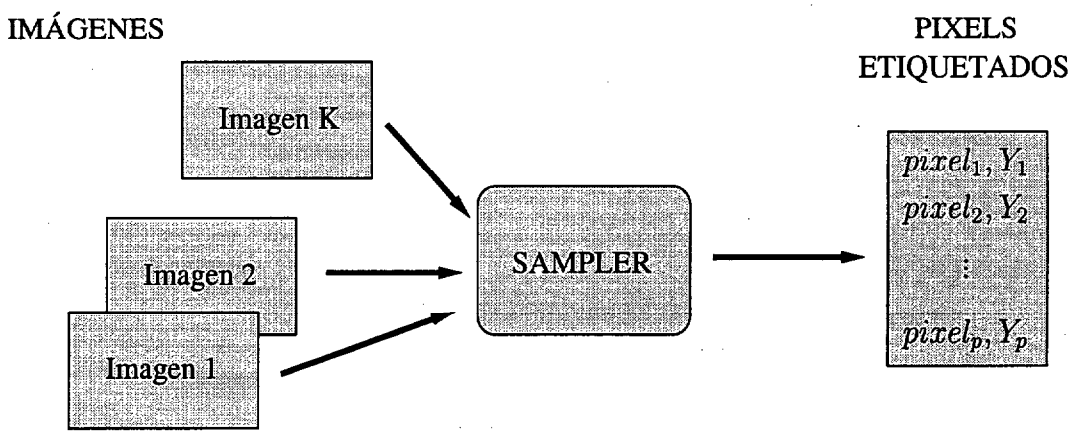


Figura 6.9: Obtención de pixels etiquetados

6.5 Sampler: Interacción y Adquisición de Muestras

El conocimiento del experto/usuario se debe adquirir en forma de ejemplos etiquetados. Por tanto un elemento importante en la arquitectura propuesta es la herramienta que permite al usuario capturar y etiquetar muestras. Resulta esencial en el diseño de esta herramienta la facilidad de utilización por parte del usuario, ya que va ser parte de la interfaz del usuario con el sistema. La interacción entre usuario y sistema propuesto se realizaría en dos sentidos. Por un lado el usuario selecciona el conjunto de muestras etiquetadas necesarias para inducir el clasificador y por otro lado monitoriza/modifica el estado de los procesos.

Una peculiaridad de la inducción de clasificadores en este entorno es que a diferencia de otros problemas, en este se dispone de una amplia población a clasificar (por ejemplo los pixels de las imágenes utilizadas), por lo que el sistema puede interrogar al usuario sobre la pertenencia de muestras no etiquetadas inicialmente, de forma que el sistema pueda refinar el clasificador inducido (Cohn et al., 1994). Una imagen del aspecto visual del prototipo desarrollado de la herramienta de adquisición de muestras diseñada, *Sampler*, se puede ver en la Figura 6.8. *Sampler* solo incluye la interacción con el usuario en un solo sentido, ya que la implementación realizada no posee capacidad de interrogar acerca de la pertenencia de muestras. El esquema de funcionamiento se puede ver resumido en la Figura 6.9. El usuario utiliza un conjunto de imágenes sobre las que define unas clases $\mathcal{Y}$ y marca pixels sobre las imágenes asignándoles valores $Y_i \in \mathcal{Y}$. Con este conjunto de pixels etiquetados y un conjunto de características se generarán las muestras de aprendizaje.

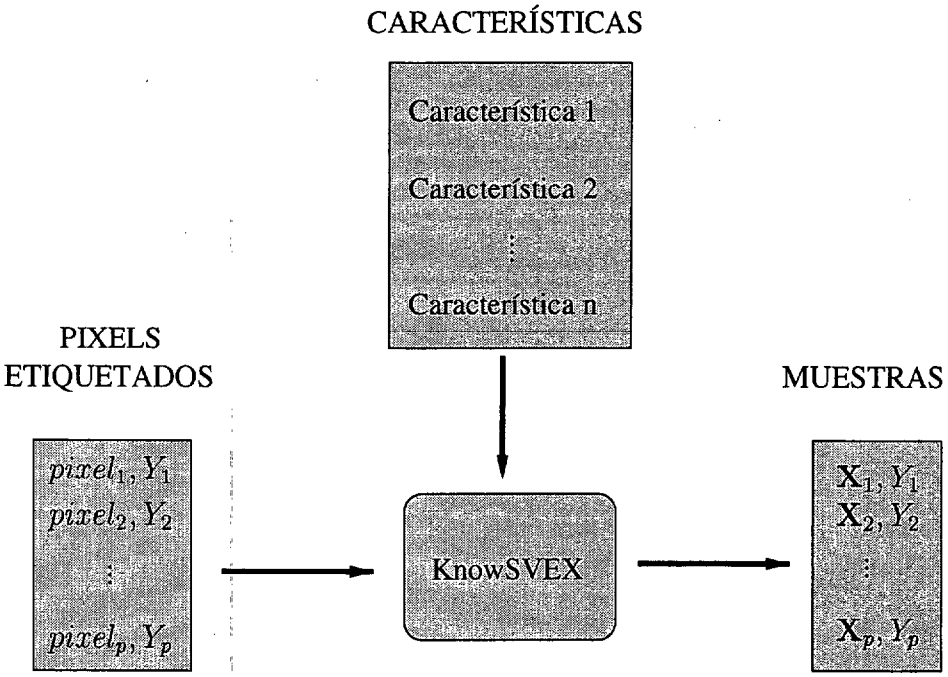


Figura 6.10: Generación de muestras etiquetadas

6.6 KnowSVEX

KnowSVEX es otra herramienta que en unión al Sampler implementa algunos de los módulos de la arquitectura propuesta (Figura 6.7). Concretamente los módulos que implementa *KnowSVEX* son la selección de características, la inducción de los clasificadores y la generación del programa en SVEX.

6.6.1 Selección de Características

La selección de características se fundamenta en el uso de la medida GD. Antes de realizar este proceso es necesario obtener las muestras de aprendizaje a partir de las características iniciales calculadas con procedimientos del procesador de pixels. Utilizando éstas y los pixels seleccionados con Sampler se genera el conjunto de aprendizaje como se puede ver en la Figura 6.10

A partir del conjunto de aprendizaje resultante se utiliza la medida GD para eliminar las características completamente irrelevantes o redundantes, y ordenar las restantes por orden decreciente de relevancia. Una diferencia que existe entre el problema de la inducción de clasificadores en este entorno y otros problemas de Aprendizaje Automático, es que aquí el usuario dispone de toda la imagen para ver el resultado del

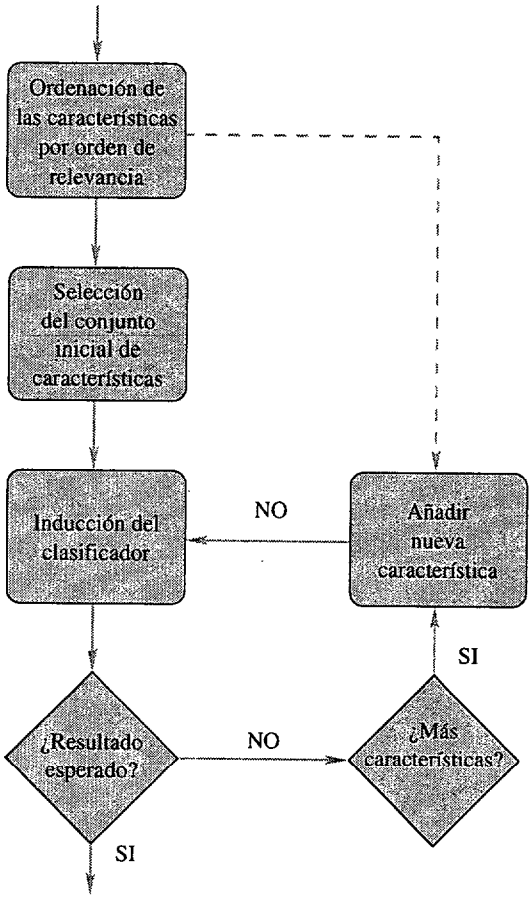


Figura 6.11: Esquema de selección de características

clasificador en el etiquetado de las clases definidas. De esta forma, la calidad del clasificador se evalúa visualmente por parte del usuario y no sólo basándose en el error del clasificador.

Un aspecto importante de utilizar un método de selección de atributos Filtro, como la medida GD, es que permite implementar un esquema híbrido de selección de características (Fig. 6.11). Inicialmente la medida GD las ordena por relevancia y luego es la calidad del clasificador inducido la que define cual es el conjunto de características relevante para el problema concreto.

Otro aspecto significativo de la utilización de la medida GD es que realiza un procedimiento de selección y no de transformación, lo que permite que las características conserven el significado semántico que les ha asociado inicialmente el usuario y éste se conserve como explicitable tras la selección/inducción.

6.6.2 Selección de Muestras Relevantes

Uno de los mayores inconvenientes de los métodos que incluyen una evaluación global del proceso de aprendizaje es el alto coste computacional, ya que para la evaluación de la calidad de cada conjunto de características candidatas es necesario llevar a cabo una ejecución completa del proceso de aprendizaje. Una forma de soslayarlo es mediante la realización de segundo proceso de selección, pero en este caso entre las muestras. Es decir, consiste en proceder dentro del conjunto de aprendizaje, y mediante un esquema de aprendizaje activo (Cohn et al., 1994), a la determinación de qué muestras son relevantes para el proceso de aprendizaje.

Además de la selección de muestras relevantes, es deseable introducir un esquema de aprendizaje incremental donde no todas las muestras se utilicen desde el principio en el proceso de aprendizaje sino que el sistema vaya seleccionando cuántas y qué muestras se utilizan en cada momento (Zhang, 1994) en función de la relevancia de la información introducida por éstas en el procedimiento de inducción. Con este esquema se evita además el problema del sobreajuste de los datos por el exceso de muestras que no suministran información relevante, lo que provoca que el clasificador inducido genere predicciones pobres. Con la utilización de este tipo de esquemas se intenta mantener un compromiso entre velocidad de aprendizaje y capacidad de clasificación correcta del clasificador inducido así como de generalización.

Este módulo no se ha implementado en la herramienta KnowSVEX, aunque el usuario tiene la posibilidad de definir el número de muestras utilizadas en el proceso de inducción. Las muestras se seleccionan de forma aleatoria entre el conjunto inicial.

6.6.3 Inducción de Clasificadores

Un aspecto importante en el diseño, y por tanto en el resultado de la inducción de clasificadores en un SVBC, es la posibilidad de transportar el conocimiento implícito en dicho clasificador al lenguaje natural (Fichera et al., 1992). Debido a la naturaleza del problema donde el conocimiento no se encuentra codificado sino que es necesario extraerlo del conjunto de muestras utilizadas para el aprendizaje correspondiente a características numéricas, un esquema válido para la inducción es el aportado por las redes neuronales (Buchanan, 1989), aunque resolviendo la transcribilidad de la misma, y por tanto del clasificador inducido.

Por razones de eficiencia a la hora de abordar la inducción de un clasificador en KnowSVEX, el enfoque es diferente según se trate una clase linealmente separable

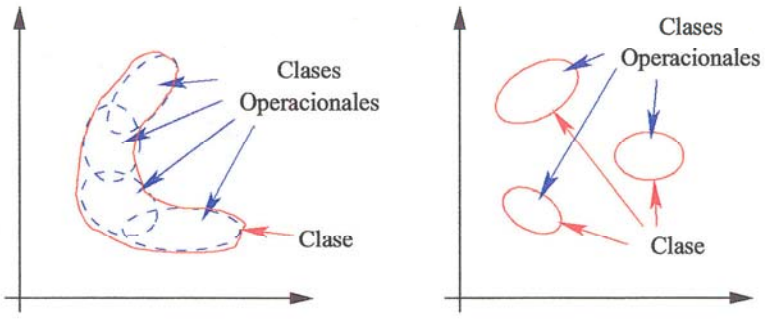


Figura 6.12: Ejemplos de clases compuestas de varias clases operacionales

o no. Para clases linealmente separables, la regla de combinación de los atributos es la combinación lineal, y utilizando el resultado de ésta como entrada a una función discriminante no lineal se puede obtener la clasificación correcta (Duda y Hart, 1973). Una posibilidad dentro del marco de las redes neuronales para inducir los clasificadores en este tipo de clases es mediante el perceptrón de una sola capa (Rosenblatt, 1960; Rumelhart et al., 1986*a*), y que ha sido la implementación utilizada en KnowSVEX.

En el caso de clases no linealmente separables, puede ocurrir que una clase se corresponda con una región de forma genérica en el espacio de características, o incluso con un conjunto de regiones no conexas. Dada la generalidad de la forma de la región correspondiente a la clase, es útil definir un concepto intermedio que denominamos *clase operacional*, entendiéndose como tal a aquella región del espacio de características separable del resto mediante la utilización de un discriminante lineal o cuadrático. De esta forma una clase estaría conformada por una mezcla de una o más clases operacionales (Fig. 6.12), siendo el número de éstas desconocido a priori. La pertenencia de una muestra a la clase vendría dada por una combinación del grado de pertenencia de esa muestra a las diferentes clases operacionales que la componen. Por lo tanto, es tarea del procedimiento de inducción el descubrimiento del número de clases operacionales necesarias, la generación de una función de pertenencia a cada una de las clases operacionales y por último, la combinación del grado de pertenencia a todas las clases operacionales para obtener el grado de pertenencia a la clase.

Un tipo de redes neuronales que permite implementar de manera directa y con las restricciones del problema el concepto de clases operacionales son las denominadas Redes Neuronales Basadas en Funciones de Base Radial (RBFN) (Renals y Rohwer, 1989; Moody y Darken, 1989*b*; Poggio y Girosi, 1990; Lee, 1992; Musavi et al., 1994*a*). Estas redes la conforman una capa de entrada, una capa oculta y una de salida. La capa oculta está compuesta de unidades cuya función de salida es una función de base radial (RBF), generalmente gaussiana aunque existen otras funciones utilizables, centradas en

un punto para el cual la salida es máxima y que decrecen a medida que la distancia con el centro aumenta, siendo la función de activación de la unidad una función que depende de la distancia entre la muestra y el centro de la RBF. En la capa de salida, la función de activación es la suma ponderada salidas de las unidades de la capa oculta, y puede poseer una función de salida que introduzca alguna no linealidad, como es el caso de una sigmoide. De esta forma, la utilización de este tipo de redes permite hacer la asociación entre cada unidad de la capa oculta y una clase operacional, lo que unido al uso de una función de distancia generalizada como la de Mahalanobis, permite que estas clases operacionales asociadas puedan ser separables cuadráticamente.

El procedimiento de inducción se encarga, tanto de determinar el campo receptivo de cada unidad, como de efectuar el reclutamiento de nuevas unidades siempre que sea necesario, lo que permite determinar automáticamente el número de unidades de la capa oculta y por tanto el número de clases operacionales.

La implementación realizada en KnowSVEX es similar a la propuesta por Musavi (Musavi et al., 1994*a*). En una primera fase se realiza un proceso de agrupamiento similar al K-Medias pero teniendo en cuenta la clase a la que pertenece cada muestra, de esta forma los agrupamientos obtenidos contienen muestras pertenecientes a la misma clase. Como resultado de esta primera fase se obtiene el número de clases naturales y su localización en el espacio de características dada por su centroide. Cada agrupamiento se hace corresponder en la red con una unidad de la capa oculta que a su vez corresponde con cada una de las funciones de base radial, φ_i. Estas funciones tienen la expresión:

$$\begin{aligned}\varphi_i(\boldsymbol{x}) &= \varphi_i(\boldsymbol{x}, \boldsymbol{c}^i, \Sigma^i) \\ &= \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{c}^i)^T\Sigma^{i^{-1}}(\boldsymbol{x}-\boldsymbol{c}^i)\right) \\ &= \exp\left(-\frac{1}{2}\sum_k\sum_l\left(h^i_{kl}(\boldsymbol{x}_k-\boldsymbol{c}^i_k)(\boldsymbol{x}_l-\boldsymbol{c}^i_l)\right)\right)\end{aligned}$$

donde $\mathbf{c}^i$ son los centros de los agrupamientos obtenidos anteriormente. Una vez se han calculado los centros de las distintas unidades de la red es necesario obtener los elementos de la matriz de covarianza. En nuestro caso se ha considerado la matriz como diagonal con todos los elementos iguales según la siguiente expresión,

$$\sigma = \frac{1}{M}\sum \|\boldsymbol{c}^i - \boldsymbol{c}^{nn-i}\|^2$$

siendo $\boldsymbol{c}^{nn-i}$ el centro más cercano al centro $\boldsymbol{c}^i$. Es decir, se toma como varianza el promedio de la distancia entre todos los centros de las unidades y el centro de la unidad

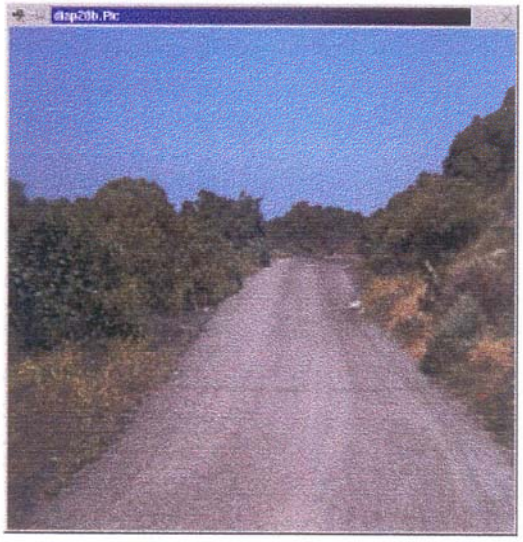


Figura 6.13: Imagen de prueba I

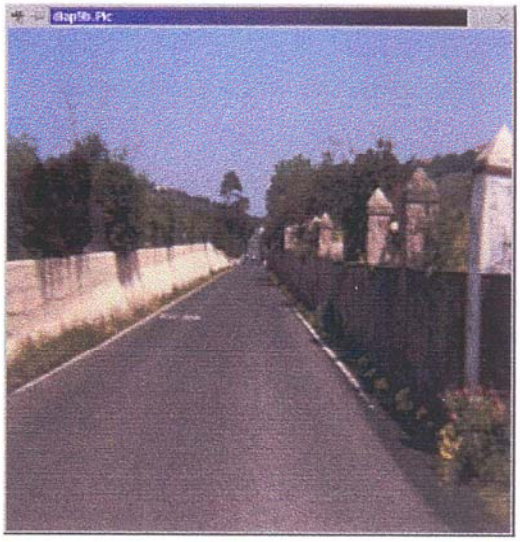


Figura 6.14: Imagen de prueba II

más cercana.

En una segunda fase se procede a obtener los pesos que conectan cada una de las unidades de la capa oculta con la capa de salida que se sigue el Procedimiento de Descenso según el Gradiente comentado en la Sección 1.7.2.

La elección del anterior esquema de entrenamiento de la red se ha elegido porque tiene la ventaja frente a otros esquemas en que el número de unidades de la capa oculta no debe ser fijado a priori, ya que en nuestro caso no se conocen las clases naturales que componen las diferentes clases. También resulta ventajoso no tener que obtener todos los parámetros que definen la red (centros, matrices de covarianza y pesos) por la técnica de descenso según el gradiente, ya que debido al elevado número de parámetros puede obtenerse un clasificador de mala calidad al quedar atrapado el procedimiento de búsqueda en mínimos locales.

6.7 Uso de KnowSVEX en Clasificación en Imágenes de Exterior

Para validar la arquitectura propuesta, y la herramienta KnowSVEX, se ha realizado el aprendizaje de tres clases utilizando imágenes de exteriores en color (Fig.s 6.13 y 6.14). En estas imágenes, el interés residía en detectar los pixels pertenecientes a las clases *Carretera*, *Cielo* y *Arbusto*. Aunque no se conocía con certeza qué características eran las mejores para resolver el problema, se suponía que estarían relacionadas con el color,

por tanto el conjunto inicial consistía en las 18 características relativas a color:

R, G, B Componente roja, verde y azul de los pixels de la imagen.

mean_R, mean_G, mean_B Valor promedio de las componentes R, G y B en una ventana de 9x9 pixels.

var_R, var_G, var_B Varianza de las componente R, G y B en una ventana de 9x9 pixels.

I1, I2, I3 Definidas a partir de las componentes RGB como $I1 = (R + G + B)/3$, $I2 = R - B$ e $I3 = G - (R + B)/2$.

mean_I1, mean_I2, mean_I3 Valor promedio de las componentes I1, I2 e I3 en una ventana de 7x7 pixels.

var_I1, var_I2, var_I3 Varianza de las componentes I1, I2 e I3 en una ventana de 7x7 pixels.

Una vez que se ha definido el conjunto inicial de características, se pasa a tomar muestras positivas y negativas de cada una de las clases de interés en las diferentes imágenes con el programa Sampler. A partir de estas muestras y con el conjunto de características definidas se obtiene el conjunto de aprendizaje. El primer paso con la herramienta KnowSVEX es ordenar el conjunto de características por orden decreciente de relevancia. Para los ejemplos mostrados aquí se utilizó como estrategia la Búsqueda Secuencial hacia Adelante.

Con las características ordenadas, se procede a la inducción del clasificador, comenzando con una sola característica y obteniendo el clasificador lineal que es el más sencillo, ya que si se obtienen resultados satisfactorios no es necesario buscar soluciones más complejas en número de características o tipo de clasificador. En el caso de no obtener un buen resultado se puede intentar añadir más características, más muestras o bien utilizar un clasificador más complejo como el no lineal. La calidad del clasificador inducido se obtiene primero mediante la tasa de error sobre el conjunto de aprendizaje. Si éste está por debajo de un umbral se pasa a comprobar el comportamiento del clasificador generado en las imágenes, bien con las utilizadas en el aprendizaje, como con otras no utilizadas para comprobar la capacidad de generalización.

En las figuras 6.15 y 6.16 se pueden ver resaltados los pixels clasificados como *Cielo* en las imágenes utilizadas para obtener los pixels de entrenamiento. En ambos casos se utilizó el mismo clasificador, uno lineal con una sola característica (componente I2), generado con el perceptrón en tan solo una iteración con un conjunto de aprendizaje de aproximadamente 1800 muestras. Además de la característica utilizada, se generaron

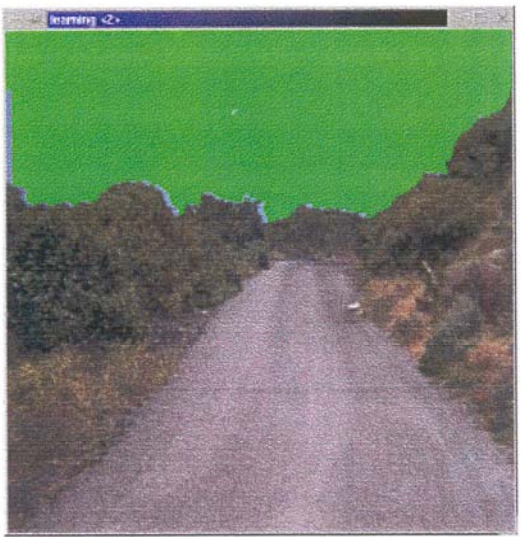


Figura 6.15: Clase *Cielo* de la imagen de prueba I

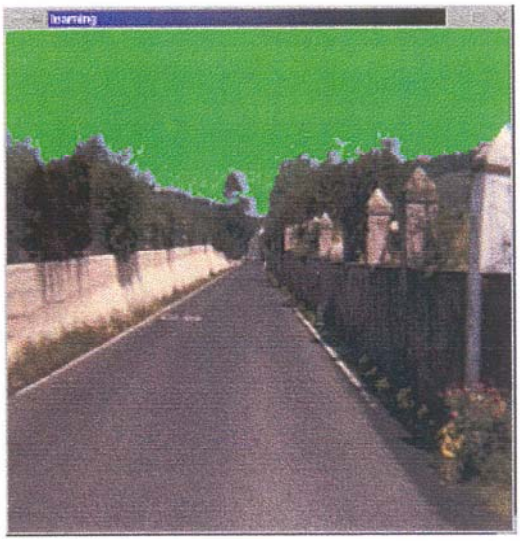


Figura 6.16: Clase *Cielo* de la imagen de prueba II

otros clasificadores utilizando más características, pero el resultado no se modificó considerablemente. Para comprobar la capacidad de generalización del clasificador generado y la característica seleccionada, se realizó la clasificación de los pixels de una imagen no utilizada en el aprendizaje y los resultados también fueron aceptables como se puede ver en la figura 6.17.

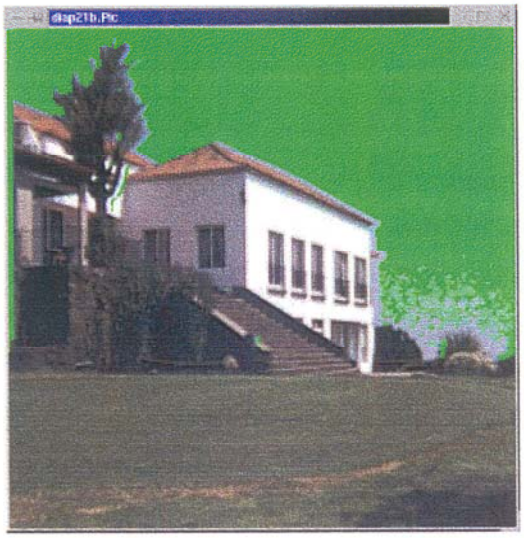


Figura 6.17: Clase *Cielo* en una imagen no utilizada en el aprendizaje

Para la clase *Carretera* fue necesario la utilización de un clasificador no lineal ya que después de varios intentos con diferentes conjuntos de características y un clasificador

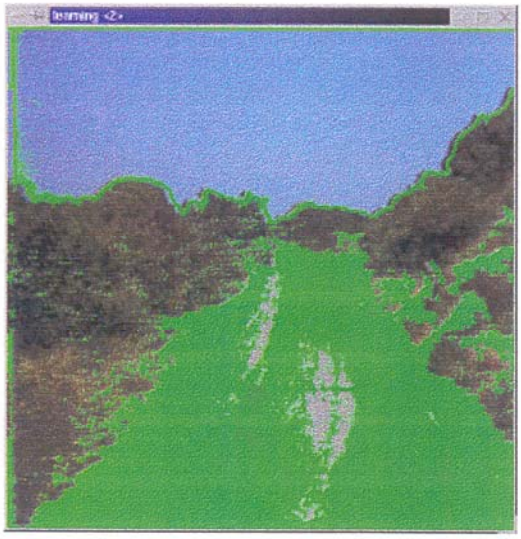


Figura 6.18: Clase *Carretera* de la imagen de prueba I

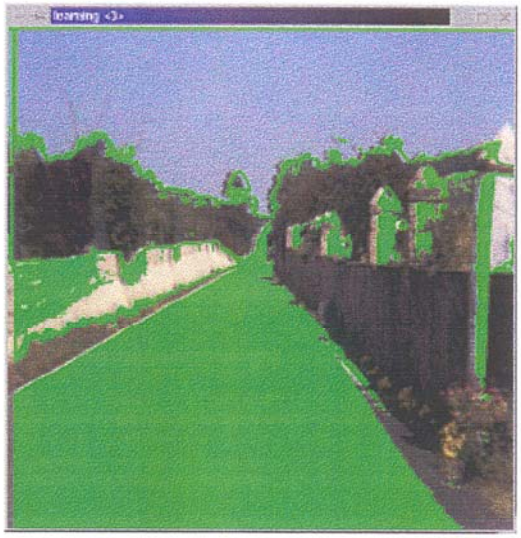


Figura 6.19: Clase *Carretera* de la imagen de prueba II

lineal no se consiguió un resultado correcto. Por tanto se consideró la utilización de un clasificador no lineal ya que en principio parece ser un problema de clases linealmente separables. Los resultados utilizando un clasificador no lineal y un conjunto de tres características (media de I3, media de Blue y componente Blue) se pueden observar en las Figuras 6.18 y 6.19.

Se puede apreciar que para la imagen de prueba I, existen unas partes de la carretera que no se han marcado como tal, mientras que en la imagen de prueba II, algunas zonas en los muros y la unión de los arbustos son marcados como pertenecientes a la carretera. Esto es así debido a la diferencia de tonos (camino en una y asfaltada en otra) que poseen las carreteras en ambas imágenes lo que supone que las muestras negativas (muros) en una imagen sean positivas en la otra imagen. Si en lugar de tomar muestras de las dos imágenes de prueba solo se utiliza una (imagen II) los resultados son mejores utilizando las mismas características con un clasificador no lineal, como puede apreciarse en la figura 6.20.

Lo anterior viene a evidenciar que algunos problemas en visión necesitan de información adicional procesada en niveles superiores para poder resolverse, o bien a nivel del Procesador de Segmentos o más arriba en el Procesador Relacional, ya que existe elementos adicionales a la señal que nos permiten a los humanos realizar el proceso de la visión.

Un elemento importante en la evaluación de la arquitectura propuesta, es la comparación de los resultados que proporciona KnowSVEX frente a los que puede obtener

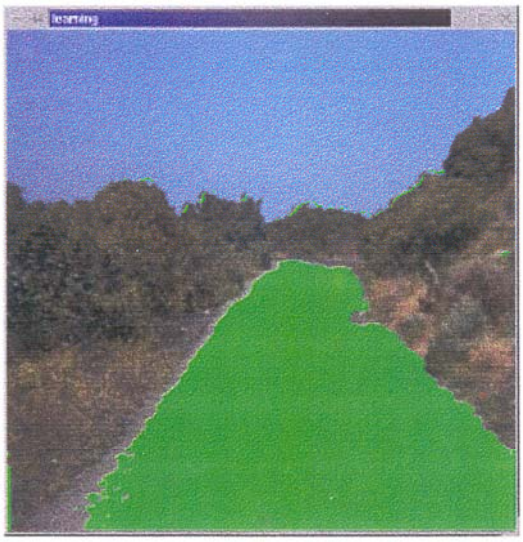


Figura 6.20: Clase *Carretera* utilizando solo muestras en el aprendizaje de la imagen de prueba I

un experto utilizando el lenguaje de SVEX y haciendo uso de su conocimiento en cuanto a características relevantes, tipo de clasificador y combinación mediante reglas de clases intermedias. Para mostrar esta comparación se escogió un ejemplo utilizando la clase *Arbusto*. El sistema generó el clasificador con una sola característica (Fig. 6.21) al igual que el experto (Fig. 6.22). Los resultados poseen una calidad bastante similar, pero la diferencia fue el tiempo necesario para llegar a estos resultados, ya que con el prototipo implementado se obtuvo el resultado en pocos minutos, mientras que el experto necesitó varias horas en un proceso de prueba y error, comprobando diferentes características y valores para los parámetros del clasificador.

Por último en la Figura 6.23 se puede ver un ejemplo de un programa en SVEX generado por la herramienta KnowSVEX a partir de un clasificador lineal entrenado con las características seleccionadas con la medida GD. En concreto es el generado para la clase *Arbusto* mostrada anteriormente. En este programa se puede ver que existen algunos comentarios relativos a elementos que tiene que introducir el usuario, como la imagen sobre la que trabajar o los procedimientos necesarios para generar las diferentes características utilizadas y que no se conocen desde KnowSVEX al no encontrarse integrado en la arquitectura de SVEX, sino que es un módulo externo a esta última.

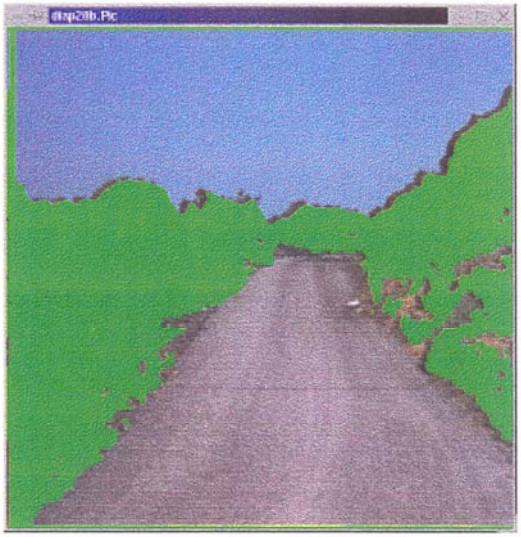


Figura 6.21: Clase *Arbusto* para la imagen de prueba I obtenida con KnowSVEX

Figura 6.22: Clase *Arbusto* para la imagen de prueba I obtenida por un experto

6.8 Uso KnowSVEX en un Sistema de Visión Activa

Otra aplicación en la que se ha utilizado KnowSVEX para comprobar la calidad de los resultados que proporciona, ha sido la clasificación de pixels en imágenes en color como pertenecientes a la clases *piel* y *no_piel*. La resolución de este problema es de gran utilidad en sistemas de detección y seguimiento de caras como DESEO (Castrillón-Santana et al., 1998; Hernández et al., 1999), y en la posterior identificación de las mismas. En (Hernández et al., 1999), la detección de las caras se realiza mediante la búsqueda de la mayor región conexa que aparece en la escena y tienen un color similar al de la piel (solo estudia el caso de individuos de raza caucásica), mientras que en (Castrillón Santana et al., 2001) los clasificadores se organizan en cascada de manera oportunista y se activan por diferentes circunstancias permitiendo refinar la clasificación de una región como cara o no a partir de una detectada como color de piel. Estos clasificadores pueden ser la forma de la región, similar a una elipse, la existencia de simetría o la pose de la cabeza, aunque siempre se aplican a regiones cuyos pixels han sido clasificados como color de piel, por lo que una detección incorrecta de estas regiones supone la imposibilidad de identificar la cara.

El clasificador que se ha utilizado en (Castrillón-Santana et al., 1998; Hernández et al., 1999; Castrillón Santana et al., 2001) se basa en rectángulos paralelos a los ejes, donde una muestra se clasifica como perteneciente a la clase si se encuentra en alguno de los rectángulos que definen dicha clase. En (Hernández et al., 1999; Castrillón

```
//
//      Program generated by KnowSVEX 1.0
//
Define Picture picture
        Channel: DT_COLOR_file
        File: "your file"
EndDefine

//
//      Features
//
Define Feature Feature_1
        // PROCEDURE TO COMPUTE stat_blue_mean FEATURE
EndDefine

// NOTE: take into account if feature Feature_1 has selectors
Define Feature nFeature_1
        From: Feature
                FeatureList: Feature_1;
        Procedure: DivFeatureByK
        Parameter: 100;
EndDefine

Define Feature threshold
        From: Feature
                FeatureList: nFeature_1;
        Procedure: ConstKFeature
        Parameter: 1.0;
EndDefine

//
//      Classifier
//
Define Classifier classifier
        FeatureList: nFeature_1, threshold;
        Functional:
                Lineal Coefficient: -6.619[illegible] 629659;
        Decision:
                SigmoidS
                        Sign: Posit[illegible]
                        Mode: Cen[illegible]
                        Level: [illegible]93, 0.043977
EndDefine

//
//      Class
//
Define Class grass
        Classifier: classifier;
        Show
        Overlapped: 80
        Picture: picture
EndDefine

//
//      Interface
//
Define Interface
        Class: grass;
EndDefine
```

Figura 6.23: Programa en SVEX generado automáticamente por KnowSVEX

Santana et al., 2001) se han utilizado las coordenadas de color UV del espacio YUV y el ajuste de los rectángulos se realiza de forma manual. Por tanto se considera que es un problema adecuado para estudiar la respuesta de KnowSVEX, tanto en las características seleccionadas como en la clasificación de los pixels en las imágenes.

Para mostrar el uso de la arquitectura implementada con KnowSVEX en este problema de identificación de caras se han utilizado como imágenes de entrenamiento las mostradas en las Figuras 6.24, 6.25, 6.26 y 6.27, que se corresponden a cuatro frames de una secuencia obtenida con DESEO y que poseen la dificultad de que el fondo tiene un color similar al de la cara. Utilizando la herramienta Sampler se seleccionan muestras de la clase *piel* y *no_piel*, y se utiliza como conjunto inicial de características las componentes de color RGB e YUV así como los promedios de éstas en una ventana de 7x7 (12 características en total). Después de realizar la selección de las características, las tres con mayor relevancia son las componentes V, promedio de V y la componente U, que serán las utilizadas en el proceso de inducción del clasificador no lineal.

Los resultados que se obtienen con el clasificador generado por KnowSVEX se pueden ver en las Figuras 6.28, 6.30, 6.32 y 6.34. En general poseen bastante ruido ya que se detectan muchas regiones pequeñas, aunque esto puede no ser un problema ya que en el sistema DESEO se considera solo la región mayor, por lo que todas las pequeñas se descartarían. En las figuras 6.28, 6.30 y 6.32 se observa que la mayor nube puntos recubre la zona de la cara, sin embargo en la Figura 6.34 la mayor región corresponde al brazo ya que ocupa más espacio en la escena. Como elemento de comparación se muestran en las Figuras 6.29, 6.31, 6.33 y 6.35 los resultados obtenidos con el clasificador por rectángulos del sistema DESEO.

Una vez que se comprobó el comportamiento del clasificador generado en las imágenes utilizadas en el aprendizaje y con un escenario similar, se utilizó el mismo clasificador sobre otras imágenes con diferente escenario y no utilizadas en el proceso de aprendizaje. Los resultados se observan en las Figuras 6.36 y 6.38, mientras que los obtenidos con DESEO se pueden ver en las Figuras 6.37 y 6.39. En ambos casos se puede ver que los resultados son similares, aunque los obtenidos con KnowSVEX son más ruidosos debido al proceso de eliminación de regiones pequeñas que incluye el sistema DESEO como una etapa posterior a la clasificación de los pixels y que no incluye KnowSVEX.

Figura 6.24: Imagen m113

Figura 6.25: Imagen m305

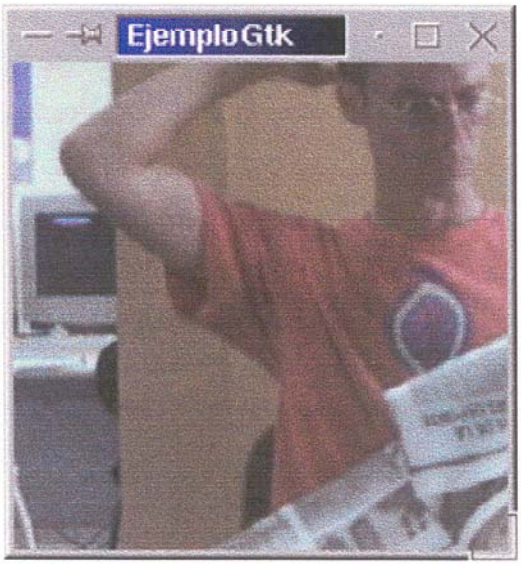


Figura 6.26: Imagen m448

Figura 6.27: Imagen m669

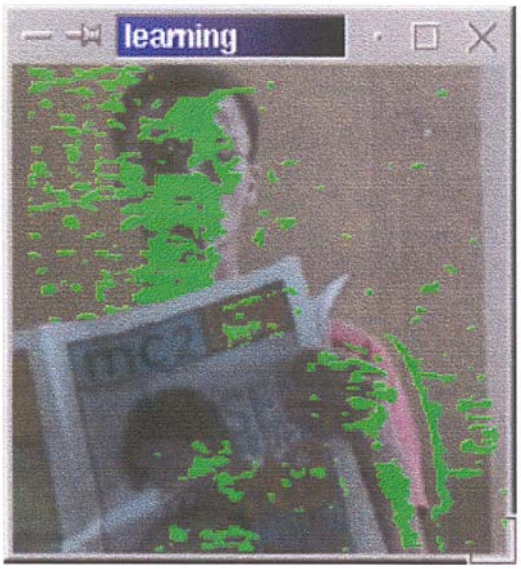


Figura 6.28: Resultado de procesar m113 con el clasificador obtenido con KnowSVEX

Figura 6.29: Resultado obtenido con DESEO en la imagen m113

Figura 6.30: Resultado de procesar m305 con el clasificador obtenido con KnowSVEX

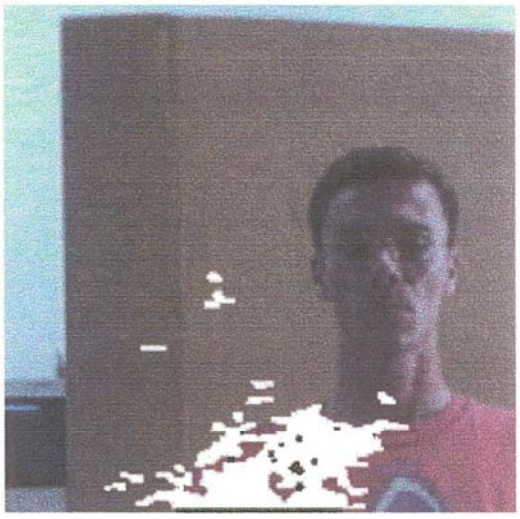


Figura 6.31: Resultado obtenido con DESEO en la imagen m305

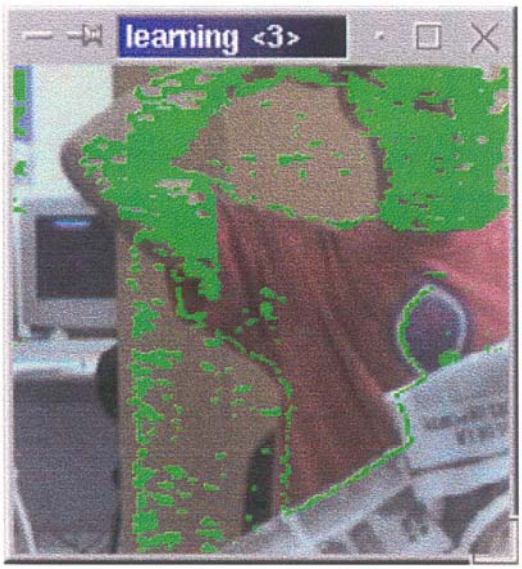


Figura 6.32: Resultado de procesar m448 con el clasificador obtenido con KnowSVEX

Figura 6.33: Resultado obtenido con DESEO en la imagen m448

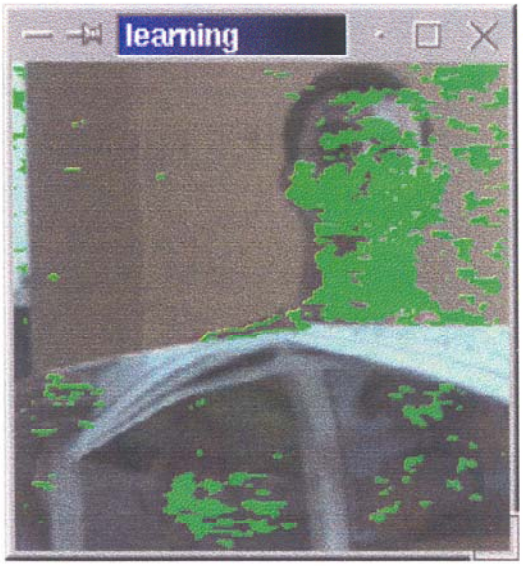


Figura 6.34: Resultado de procesar m669 con el clasificador obtenido con KnowSVEX

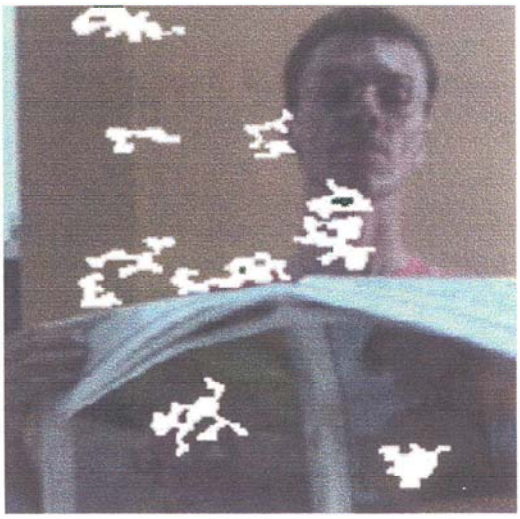


Figura 6.35: Resultado obtenido con DESEO en la imagen m669

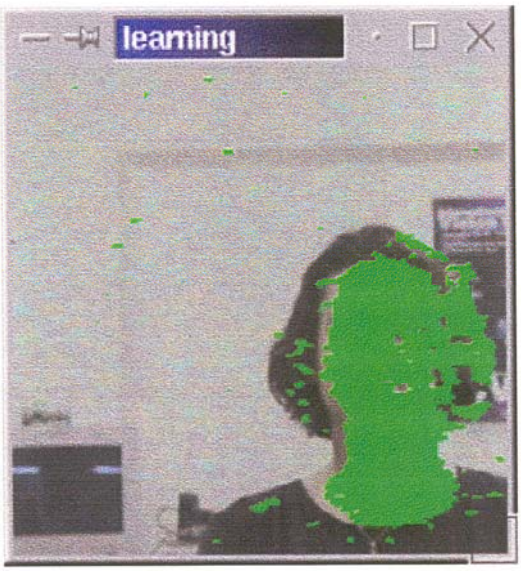


Figura 6.36: Resultado de procesar c651 con el clasificador obtenido con KnowSVEX

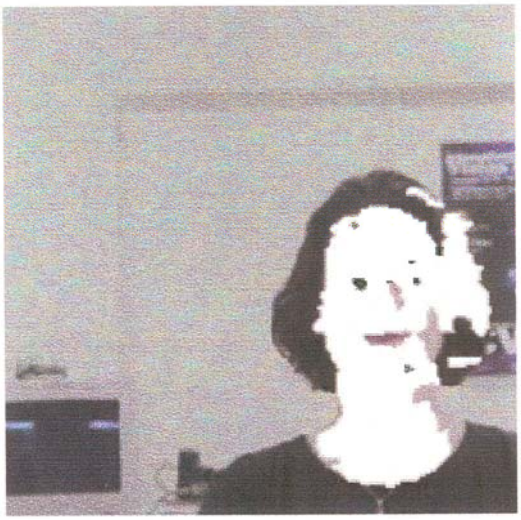


Figura 6.37: Resultado obtenido con DESEO en la imagen c651

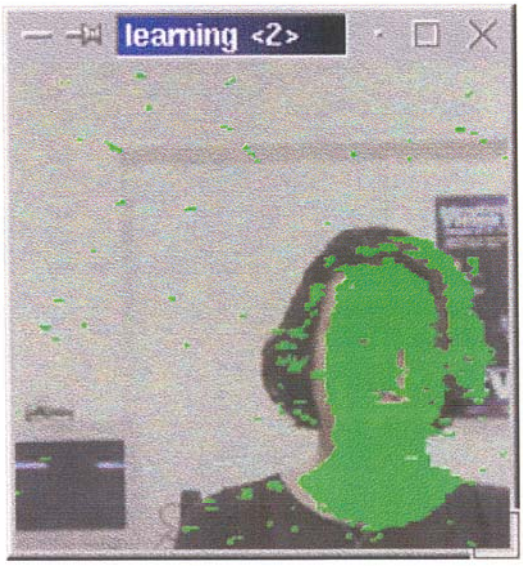


Figura 6.38: Resultado de procesar c999 con el clasificador obtenido con KnowSVEX

Figura 6.39: Resultado obtenido con DESEO en la imagen c999

Conclusiones

La selección de atributos en campos como el Aprendizaje Automático o el Reconocimiento de Formas sigue siendo un tema de interés Para comprobarlo basta con revisar los trabajos publicados en los últimos años sobre el tema. En estos trabajos se intenta encontrar métodos que permitan detectar aquellos atributos que aporten la mayor información posible al proceso de aprendizaje y cuya obtención posea un costo computacional abordable. Como se ha comentado a lo largo del documento, los métodos de selección se pueden clasificar en general como Filtro que son independientes del clasificador utilizado, y los Envolventes (*wrapper*) que basan la búsqueda en el rendimiento del clasificador.

La propuesta realizada en esta tesis se encuadra dentro de los métodos Filtro con la introducción de la medida GD que está basada en Teoría de la Información y que permite obtener los atributos más relevantes con una implementación que evita la estimación de distribuciones de probabilidad multivariantes, que supondría un costo computacional exponencial con el número de atributos del problema. No obstante se recogen las posibles dependencias entre atributos.

Conclusiones

A continuación se exponen las conclusiones que presentamos como resultado de la realización de esta tesis.

1. Una conclusión a la que se ha llegado es que el problema de la selección de atributos en Aprendizaje Automático, es un tema abierto a pesar del tiempo que se lleva tratando, a tenor de la bibliografía encontrada en la revisión bibliográfica realizada. En la revisión bibliográfica también se puede constatar que no es un problema que se pueda enfocar desde un único marco conceptual. En efecto, distintos autores han abordado este problema dándole diferentes enfoques. Este hecho precisa la

clasificación de los métodos en función de diferentes parámetros. En concreto en esta tesis se ha utilizado la clasificación propuesta por Doak.

2. El marco conceptual de esta tesis es la Teoría de la Información. Haciendo uso de ella se ha encontrado una relación entre un clasificador considerado como una caja negra que acepta como entradas muestras y a su salida devuelve la clase a la que pertenece esta muestra, y un canal de información como lo define la Teoría de la Información, que acepta símbolos de un alfabeto de entrada y da como resultado símbolos de un alfabeto de salida.

3. Haciendo uso del anterior marco se definen los conceptos subyacentes de dicha teoría aplicada a la selección de atributos entre los que se destacan: *a)* el de Subconjunto de Atributos Suficientes (SAS), similar al propuesto por Wang, como los atributos que aportan la información al proceso de aprendizaje; *b)* la diferenciación para los restantes atributos entre no informativos y redundantes, demostrando que estos últimos pueden ser intercambiados con otros del SAS sin pérdida de información.

4. Para evitar el elevado costo computacional que supone la implementación de los conceptos expuestos en el marco conceptual de la tesis, se propone la medida GD para obtener el SAS recogiendo las posibles interdependencias de los atributos y sin necesidad de estimar funciones de probabilidad multivariantes. Esta medida se puede considerar una generalización de la distancia de Mántaras, ya que introduce la Matriz de Transinformación como mecanismo para recoger la interdependencia de los atributos y además detectar de forma sencilla los atributos redundantes dos a dos. Para la utilización de la medida GD para selección se proponen dos procesos de búsquedas, Secuencial hacia Adelante (GD-SFS) y la Branch&Bound (GD-BB). Encontrando que en las bases de datos utilizadas (unas decenas de atributos) en los experimentos, los resultados de las búsquedas, incluyendo la Secuencial hacia Atrás, son bastante similares.

5. La utilización de la medida GD supone la estimación de funciones de probabilidad, pero éstas se pueden ver afectadas por la existencia de atributos perdidos. Por ello se propone un esquema de sustitución que se demuestra que no introduce un sesgo en la relevancia de los atributos, penalizando negativamente aquellos atributos con mayor cantidad de valores perdidos, que se considera razonable ya que la información que aportan al proceso de aprendizaje con respecto al resto es menor.

6. La calidad de la medida GD para obtener el conjunto de atributos relevantes se

evalúa de forma empírica. Para comprobar diferentes aspectos de la medida GD se diseñó y ejecutó un conjunto de experimentos, que resumimos:

(a) La medida GD a diferencia de las otras medidas basadas en Teoría de la Información utilizadas en la selección de atributos con las que se comparó, muestra un sesgo menor hacia los atributos con mayor número de valores frente a atributos con un número menor. Las evidencias que lo muestran se obtuvieron a partir del experimento propuesto por Kononenko para estudiar el sesgo introducido por medidas de selección de atributos en función del número de valores de los atributos y la dependencia de la clase con estos atributos. Además para los atributos relevantes el valor de la medida GD es menor que para los irrelevantes, recogiendo por tanto la relevancia de los mismos.

(b) La bondad de la medida GD para la selección de atributos se demostró en un conjunto de bases de datos sintéticas de diferente complejidad en cuanto a la dependencia de la clase con los atributos y de los atributos entre si. En todos los casos se seleccionó el conjunto correcto de los atributos cuando el número de estos es conocido. La única excepción apareció con una base de datos propuesta por John que es la CorrAL, en la que los métodos wrapper parecen demostrar una superioridad frente a los métodos Filtro. Pero como se comprobó también, esta ventaja de los métodos wrapper radica en el clasificador utilizado, ya que dependiendo de éste, el resultado es diferente y para el caso del clasificador bayesiano, el método wrapper y la medida GD dan iguales resultados. Un tipo de bases de datos bastante difícil para los métodos Filtro es el de aquellas en las que la dependencia de la clase con los atributos es un OR-exclusivo. En este caso, si los atributos son lógicos la medida GD no es capaz de encontrar los atributos relevantes, sin embargo si los atributos son continuos la medida GD detecta correctamente los atributos que definen la clase. Este hecho se basa en que con atributos continuos se rompe la total incertidumbre que desde el punto de vista de la Teoría de la Información se produce cuando se tienen atributos y clases equiprobables, que es la situación de mayor entropía y por tanto mayor incertidumbre.

(c) La medida GD muestra también un buen comportamiento en bases de datos reales, donde la dependencia con los atributos no es conocida a priori y por tanto la bondad del método se puede estimar en base a la tasa de acierto obtenida por los clasificadores utilizando el conjunto de atributos seleccionados. En la comparativa con otros dos métodos de selección y con tres

clasificadores diferentes y validados con un doble test de hipótesis, la medida GD da resultados mejores que los otros dos métodos en general y tomando en consideración el árbol de decisión, se obtiene iguales o mejores tasas de acierto con conjuntos seleccionados con la medida GD con una cardinalidad menor, por lo que se cumple el objetivo en el que se fundamenta la tesis de encontrar una medida que cumpla con el Principio Empírico de la Cuchilla de Occam, que a igual tasa de acierto seleccione conjuntos de atributos con menor cardinalidad.

7. Como último elemento de la tesis, se propone y realiza la implementación de una arquitectura propuesta para el aprendizaje de clasificadores en un entorno de un sistema de visión basado en conocimiento mediante la herramienta KnowSVEX, en la que se incluye la medida GD como selector de atributos. Esta implementación se muestra eficaz para aliviar el cuello de botella que supone la obtención de conocimiento a la hora de diseñar clasificadores, así como el incremento de calidad de los resultados con un número reducido de atributos, repercutiendo en una mayor simplicidad del programa generado en SVEX. Este aspecto quedó demostrado en la comparación del resultado obtenido por un experto haciendo uso de su conocimiento sobre el problema y el que se obtiene con KnowSVEX. En ambos casos la calidad fue similar, aunque con KnowSVEX se resolvió el problema en poco minutos mientras que el experto se vió obligado a realizar un proceso de ajuste manual de los distintos parámetros hasta encontrar un resultado satisfactorio.

Trabajos Futuros

En esta sección se indican algunas ideas que pueden continuar el trabajo presentado en esta tesis.

1. Un punto que queda abierto es la demostración analítica de la no singularidad de la matriz de transinformación, que solo se ha encontrado para dimensión 2 y 3 pero no para dimensiones mayores, aunque en todas las pruebas realizadas no ha existido ninguna matriz de transinformación que después de haber eliminado los atributos completamente redundantes fuera singular.

2. Estudiar la utilización de la medida GD en Minería de Datos donde un problema de estudio es la detección de dependencia entre atributos para obtener patrones en grandes bases de datos. Una diferencia de estos problemas con los problemas

habituales en Aprendizaje Automático es el alto número de atributos, lo que permitirá comprobar el comportamiento de la matriz de información y por tanto de la medida GD en problemas de elevada dimensionalidad, del orden de cientos de atributos.

3. Incluir la medida GD como selector de atributos en problemas de aprendizaje en robótica para seleccionar características calculadas a partir de los sensores que permitan determinar con mayor exactitud el entorno en el que se encuentra un robot en un momento dado, o computar configuraciones que ayuden a la localización del robot en un entorno determinado.

Bibliografía

Aamodt, A. y Plaza, E. (1994). 'Case-based reasoning: foundational issues, methodological variations, and system approaches.' *AI Communications*, **7**(1), págs. 39–59.

Abramson, N. (1986). *Teoría de la Información y Codificación.* Paraninfo S.A., Madrid.

Aha, D. W., editor (1997). *Lazy Learning.* Kluwer, Norwell, MA.

Aha, D. W. y Bankert, R. L. (1994). 'Feature selection for case-based classification of cloud types: An empirical comparison.' En 'Proceedings of the 1994 AAAI Workshop on Case-Based Reasoning,' págs. 106–112. AAAI Press.

Aha, D. W. y Bankert, R. L. (1995). 'A comparative evaluation of sequential feature selection algorithms.' En 'Proceedings of the Fifth Int. Workshop on Artificial Intelligence and Statistics,' págs. 1–7. Unpublished.

Aha, D. W., Kibler, D. y Albert, M. K. (1991). 'Instance-based learning algorithms.' *Machine Learning*, **6**, págs. 37–66.

Almuallim, H. y Dietterich, T. G. (1992). 'Efficient algorithms for identifying relevant features.' En 'Proceedings of the Ninth Canadian Conf. on Artificial Intelligence,' págs. 38–45. Morgan-Kauffman.

Almuallim, H. y Dietterich, T. G. (1994). 'Learning boolean concepts in the presence of many irrelevant features.' *Artificial Intelligence*, **69**(1-2), págs. 279–306.

Aloimonos, J. y Weis, I. (1988). 'Active vision.' *International Journal of Computer Vision*, págs. 333–356.

Anderberg, M. R. (1973). *Cluster Analysis for Applications.* Academic Press Inc., New York.

Anzai, Y. (1992). *Pattern Recognition and Machine Learning.* Academic Press, Inc.

Ash, R. B. (1965). *Information Theory.* Dover Publishing, Inc., New York.

Barto, A. G., Sutton, R. S. y Anderson, C. W. (1983). 'Neuronlike adaptative elements that can solve difficult learning control problems.' *IEEE Trans. on Systems, Man and Cybernetics*, **SMC-13**(5), págs. 834–846.

Battiti, R. (1994). 'Using mutual information for selecting features in supervised neural net learning.' *IEEE Trans. on Neural Networks*, **5**(4), págs. 537–550.

Bellman, R. (1957). *Dynamic Programming.* Princeton University Press, Princeton, New Jersey.

Bertsekas, D. P. (1987). *Dynamic Programming: Deterministic and Stochastic Models.* Prentice Hall, Englewood Cliffs, NJ.

Bhandaru, M., Draper, B. A. y Lesser, V. R. (1993). 'Learning image to symbol conversion.' En A. Press, editor, 'Proceedings of the 1993 AAAI Fall Symposium on Machine Learning in Computer Vision: What, Why and How?', págs. 6–9.

Blahut, R. E. (1991). *Principles and Practice of Information Theory.* Addison-Wesley Publishing Co.

Blake, C. y Merz, C. (1998). 'UCI repository of machine learning databases.'

Blum, A. L. y Langley, P. (1997). 'Selection of relevant features and examples in machine learning.' *Artificial Intelligence*, págs. 245–271.

Blumer, A., Enrenfeucht, A., Haussler, D. y Warmuth, M. K. (1987). 'Occam's razor.' *Information Processing Letters*, **24**, págs. 377–380.

Bow, S. (1992). *Pattern Recognition and Image Preprocessing*, capítulo Dimensionality Reduction and Feature Selection, págs. 142–152. Marcel Dekkers, New York.

Bowyer, K. W., Hall, L., Langley, P., Bhanu, B. y Drapper, B. A. (1994). 'Report of the AAAI fall symposium on machine learning and computer vision: What, why and how?' En '1994 ARPA Image Understanding Workshop,' Morgan Kaufman, San Mateo, California.

Bradley, P. S. y Mangasarian, O. L. (1998). 'Feature selection via convave minimization and support vector machines.' En 'Proceedings of the 15th International Conference on Machine Learning,' págs. 82–90. Morgan Kaufmann, San Francisco.

Bradley, P. S., Mangasarian, O. L. y Street, W. N. (1998). 'Feature selection via mathematical programming.' *INFORMS Journal on Computing*, **10**, págs. 209–217.

Brassard, G. y Bratley, P. (1996). *Fundamentals of Algorithms.* Prentice Hall, Englewood Cliffs, New Jersey.

Breiman, L., Friedman, J. H., Olshen, R. A. y Stone, C. J. (1993). *Classification and Regression Trees.* Wadsworth & Brooks.

Bridle, J. S., Heading, A. J. R. y MacKay, D. J. C. (1992). 'Unsupervised classifiers, mutual information and 'phantom targets'.' En J. E. Moody, S. J. Hanson y R. P. Lippmann, editores, 'Advances in Neural Information Processing Systems,' tomo 4, págs. 1096–1101. Morgan Kaufmann Publishers, Inc.

Brill, F. Z., Brown, D. E. y Martin, W. N. (1992). 'Fast genetic selection of features for neural classifiers.' *IEEE Trans. on Neural Networks*, **3**(2), págs. 324–328.

Bronson, R. (1991). *Matrix Methods.* Academic Press Inc., segunda edición.

Broomhead, D. S. y Lowe, D. (1988). 'Multivariate functional interpolation and adaptative networks.' *Complex Systems*, **2**, págs. 321–355.

Buchanan, B. (1989). 'Can machine learning offer anything to expert systems?' *Machine Learning*, págs. 251–254.

Cabrera, J. (1995). *Sistema Basado en Conocimiento para Segmentación de Imagenes. Desarrollos y Aplicaciones.* Tesis Doctoral, Dpto.de Informática y Sistemas, Univ. de Las Palmas de Gran Canaria.

Carbonell, J. G. (1983). 'Learning by analogy: Formulating and generalizing plans from past experience.' En J. G. Carbonell, R. S. Michalski y T. M. Mitchell, editores, 'Machine Learning: An Artificial Intelligence Approach,' Morgan Kaufmann, Los Altos, CA.

Carbonell, J. G. (1989). 'Introduction: Paradigms for machine learning.' En J. G. Carbonell, editor, 'Machine Learning. Paradigms and methods,' Elsevier Science Publishers, Amsterdam, The Netherlands.

Carbonell, J. G., Michalski, R. S. y Mitchel, T. M. (1983). 'An overview of machine learning.' En R. S. Michalski, J. G. Carbonell y T. M. Mitchel, editores, 'Machine Learning,' Spinger Verlag, Berlin.

Cardie, C. (1993). 'Using decision trees to improve case-based learning.' En 'Proceedings of the 10th International Conference on Machine Learning,' págs. 25–32. Morgan Kaufmann.

Cardie, C. (2000). 'A cognitive bias approach to feature selection and weighting for case-based learners.' *Machine Learning*, **41**, págs. 85–116.

Cardie, C. y Howe, N. (1997). 'Empirical methods in information extraction.' En D. Fischer, editor, 'Proceedings of the 14th International Conference on Machine Learning,' págs. 65–79. Morgan Kaufmann.

Caruana, R. y Freitag, D. (1994). 'Greedy attribute selection.' En (Cohen y Hirsh, 1994), págs. 28–36.

Castrillón-Santana, M., Guerra-Artal, C., Hernández-Sosa, J., Domínguez-Brito, A., Isern-González, J., Cabrera-Gámez, J. y Hernández-Tejera, F. (1998). 'An active vision system integrating fast and slow processes.' En 'roc. of the SPIE'98 Symposium on Intelligent Systems and Advanced Manufacturing,' págs. 487–496. Boston (USA).

Castrillón Santana, M., Lorenzo Navarro, J., Hernández Tejera, M. y Cabrera Gámez, J. (2001). 'Before characterizing faces.' En 'Proc. of IX Spanish Symposium on Pattern Recognition and Image Analysis,' Castellón,Spain.

Cleary, J. G. y Trigg, L. E. (1995). 'K*: An instance-based leaner using an entropic distance measure.' En 'Proceedings of the 12th Int. Conference on Machine Learning,' págs. 108–114. Morgan Kaufmann.

Clemént, V. y Thonnat, M. (1993). 'A knowledge-based approach to integration of image processing procedures.' *CVGIP: Image Understanding*, **57**(2), págs. 166–184.

Cohen, W. y Hirsh, H., editores (1994). *Proceedings of the 11th International Conference on Machine Learning.* Morgan Kaufmann.

Cohn, D., Atlas, L. y Ladner, R. (1994). 'Improving generalization with active learning.' *Machine Learning*, **15**, págs. 201–221.

Comon, P. (1995). 'Supervised classification: a probabilistic approach.' En M. Verleysen, editor, 'ESANN-95 European Symposium on Artificial Neural Networks,' págs. 111–128. D facto Pub., Brussels, Belgium.

Cover, T. M. (1965). 'Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition.' *IEEE Trans. on Electronics Computers*, **EC-14**, págs. 326–334.

Cover, T. M. y Thomas, J. A. (1991). *Elements of Information Theory.* John Wiley & Sons Inc.

Cristianini, N. y Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods).* CUP.

Cuadras Avellana, C. M. (1981). *Métodos de Análisis Multivariante.* EUNIBAR - Editorial Universitaria de Barcelona.

Cybenko, G. (1989). 'Approximation by superpositions of a sigmoidal function.' *Mathematical of Control, Signals and Systems*, **2**, págs. 303–314.

Daelemans, W. y van den Bosch, A. (1992). 'Generalization performance of backpropagation learning on a syllabification task.' En 'Proceedings of the Third Twente Workshop on Language Technology,' págs. 27–38.

Dash, M. y Liu, H. (1997). 'Feature selection for classification.' *Intelligent Data Analysis*, **1**(3).

Dash, M., Liu, H. y Motoda, H. (2000). 'Consistency based feature selection.' En 'Proceedings of the Pacific-Asian Knowledge and Data Discovery Conference,' págs. 89–109. Springer-Verlag, Kyoto, Japan.

Davies, S. y Russell, S. (1994). 'NP-Completeness of searches for smallest possible feature sets.' En 'Proceedings of the AAAI Fall Symposium on Relevance,' págs. 37–39. New Orleans.

Davis, L., editor (1991). *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York.

Deco, G., Finnoff, W. y Zimmermann, H. G. (1995). 'Unsupervised mutual information criterion for elimination of overtraining in supervised multilayer networks.' *Neural Computation*, **7**, págs. 86–105.

Devijver, P. A. y Kittler, J. (1982). *Pattern Recognition: A Statistical Approach.* Prentice-Hall, Englewood Cliffs, New Jersey.

Dietterich, T. G. (1998). 'Approximate statistical tests for comparing supervised classification learning algorithms.' *Neural Computation*, **10**(7), págs. 1895–1924.

Doak, J. (1994). 'An evaluation of search algorithms for feature selection.' Informe técnico, Safeguards Systems Group. Los Alamos National Laboratory.

Domingos, P. (1997). 'Context-sensitive feature selection for lazy learners.' *Artificial Intelligence Review. Special Issue on Lazy Learners*, **11**, págs. 227–253.

Domingos, P. (1999). 'The role of occam's razor in knowledge discovery.' *Data Mining and Knowledge Discovery*, **3**(4), págs. 409–425.

Duda, R. y Hart, P. (1973). *Pattern Classification and Scene Analysis.* John Willey and Sons.

Efron, B. (1979). 'Bootstrap methods: another look at the jacknife.' *Anals of Statistics*, **7**(1), págs. 1–26.

Efron, B. (1983). 'Estimating the error rate of a prediction rule: some improvements on cross-validation.' *Journal of the American Statistical Association*, **78**, págs. 316–331.

Efron, B. y Tibshirani, R. J. (1993). *An Introduction to the Bootstrap.* Chapman and Hall, London, UK.

Efron, B. y Tibshirani, R. J. (1995). 'Cross-validation and the Bootstrap: Estimating the error rate of a prediction rule.' Informe Técnico 178, Division of Biostatistics. Stanford University.

Fayyad, U. M. y Irani, K. B. (1992). 'The attribute selection problem in decision tree generation.' En W. Swartout, editor, 'Proceedings of the 10th National Conference on Artificial Intelligence,' págs. 104–110. MIT Press, San Jose, CA. ISBN 0-262-51063-4.

Fichera, O., Pellegretti, P., Roli, F. y Serpico, S. B. (1992). 'Automatic acquisition of visual models for image recognition.' En 'Proceedings of the 11th. Int. Conf. on Pattern Recognition,' págs. 95–98.

Fikes, R., Hart, P. y Nilsson, N. (1972). 'Learning and executing generalized robot plans.' *Artificial Intelligence*, **3**, págs. 251–288.

Fisher, D. H. (1987). 'Knowledge acquisiton via incremental conceptual clustering.' *Machine Learning*, **2**, págs. 139–172.

Foroutan, I. (1987). 'Feature selection for automatic classification of non-gaussian data.' *IEEE Trans. on Systems, Man and Cybernetics*, **17**(2), págs. 187–198.

Forsyth, R. (1989). 'The logic of induction.' En Chapman y H. Ltd., editores, 'Machine Learning. Principles and Techniques,' Richard Forsyth.

Forsyth, R. y Rada, R. (1986). *Machine Learning. Applications in Expert Systems and Information Retrieval.* Ellis Horwood Limited, England.

Fraser, A. M. (1989). 'Information and entropy in strange attractors.' *IEEE Trans. Information Theory*, **35**(2), págs. 245–262.

Fraser, A. M. y Swinney, H. L. (1986). 'Independent coordinates for strange attractors from mutual information.' *Physical Review A*, **33**(2), págs. 1134–1140.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition.* Academic Press Inc., segunda edición.

Gennari, J. H., Langley, P. y Fisher, D. (1989). 'Models of incremental concept formation.' *Artificial Intelligence*, **40**(1-3), págs. 11–61.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Pub. Co.

González, A. y Pérez, R. (1997). 'Using information measures for determining the relevance of the predictive variables in learning problems.' En 'Proceedins of the Congress of FUZZ-IEEE'97,' págs. 1423–1428. Barcelona (Spain).

Grefenstette, J. (1988). 'Credit assignment in rule discovery systems.' *Machine Learning*, **3**(2/3), págs. 225–246.

Guerra-Salcedo, C., Chen, S., Whitley, D. y Smith, S. (1999). 'Fast and accurate feature selection using hybrid genetic strategies.' En P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao y A. Zalzala, editores, 'Proceedings of the Congress on Evolutionary Computation,' tomo 1, págs. 177–184. IEEE Press.

Hall, M. A. (2000). 'Correlation-based feature selection for discrete and numeric class machine learning.' En 'Proceedings of the ICML-2000, 17th International Conference on Machine Learning,' págs. 359–366. Morgan Kaufmann Pub., San Francisco, CA.

Hamamoto, Y., Uchimura, S. y Tomita, S. (1996). 'On the behavior of artificial neural networks classifiers in high-dimensional spaces.' *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **18**(5), págs. 571–574.

Hand, D. J. (1986). *Discrimination and Classification.* John Wiley and Sons.

Harp, S. A. y Samad, T. (1991). 'Genetic synthesis of neural network architecture.' En (Davis, 1991).

Hass, N. y Hendrix, G. G. (1983). 'Learning by being told: Acquiring knowledge for information management.' En R. S. Michalski, J. G. Carbonell y T. M. Mitchell, editores, 'Machine Learning: An Artificial Intelligence Approach,' págs. 405–427. Springer Verlag.

Haykin, S. (1994). *Neural Networks; A comprehensive Foundation.* Macmillan, New York, primera edición.

Hecht-Nielsen, R. (1989). *Neurocomputing.* Addison-Wesley Publishing Company, Inc., Menlo Park, CA.

Hernández, F., Cabrera, J., Castrillón, M., Dominguez, A., Guerra, C., Hernández, D. y Isern, J. (1999). 'DESEO: An active vision system for detection, tracking and recognition.' En H. I. Christensen, editor, 'Lectures Notes in Computer Science, International Conference on Vision Systems (ICVS'99),' tomo 1542, págs. 379–391.

Hernández, J. D., Cabrera, J., A., F. y Hernández, M. (1995). 'SVEX: A knowledge-based tool for image segmentation.' En 'Proceedings of 1995 Int. Conference on Acoustics, Speech and Signal Processing,' tomo 57, págs. 166–184.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press. Reeditado por MIT Press en 1992.

Holland, J. H. (1986). 'Escaping brittleness: the possibilities of general purpose algorithms applied to parallel rule-based systems.' En R. S. Michalski, J. G. Carbonell y T. M. Mitchell, editores, 'Machine Learning, an Artificial Intelligence approach,' tomo 2, págs. 593–623. Morgan Kaufmann, San Mateo, California.

Holte, R. C. (1993). 'Very simple classification rules perform well on most commonly use datasets.' *Machine Learning*, **11**, págs. 63–91.

Hornick, K., Stinchcombe, M. y White, H. (1989). 'Multilayer feedforward networks are universal approximators.' *Neural Networks*, **2**, págs. 359–366.

Hunt, E. B., Marin, J. y Stone, P. T. (1966). *Experiments in Induction.* Academic Press, New York.

Ichino, M. y Sklansky, J. (1984). 'Optimum feature selection by zero-one integer programming.' *IEEE Trans. on Systems, Man and Cybernetics*, **14**(5), págs. 737–746.

Imam, I. F. y Vafaie, H. (1994). 'An empirical comparison between global and greedy-like search for feature selection.' En 'Proceedings of the Florida AI Research Symposium (FLAIRS-94),' págs. 66–70.

Intrator, N. (1992). 'Feature extraction using an unsupervised neural network.' *Neural Computation*, **4**, págs. 98–107.

Jain, A. y Zongker, D. (1997). 'Feature selection: Evaluation, application, and small sample performance.' *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **19**(2), págs. 153–157.

Jain, A. K., Dubes, R. C. y Chen, C. (1987). 'Bootstrap techniques for error estimation.' *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **9**(5), págs. 628–633.

Jensen, J. R. (1986). *Introductory Digital Image Processing*, capítulo Thematic Information Extration. Prentice Hall.

John, G. H., Kohavi, R. y Pfleger, K. (1984). 'Irrelevant features and the subset selection problem.' En W. W. Cohen y H. Hirsh, editores, 'Machine Learning: Proceedings of the Eleventh International Conference,' págs. 121–129. Morgan Kaufmann Publishers, Inc., San Francisco, CA.

Jun, B. H., Kim, C. S., Song, H.-Y. y Kim, J. (1997). 'A new criterion in selection and discretization of attributes for the generation of decision trees.' *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **19**(12), págs. 1371–1375.

Kearns, M. J. y Vazirani, U. V. (1994). *An Introduction to Computational Learning Theory.* The MIT Press, Cambridge, Massachusetts.

Kerber, R. (1992). 'ChiMerge: Discretization of numeric attributes.' En 'AAAI-92, Proceedings of the Ninth National Conference on Artificial Intelligence,' págs. 129–134. AAAI Press/The MIT Press.

Kira, K. y Rendell, L. A. (1992). 'The feature selection problem: Traditional methods and a new algorithm.' En 'Proceedings of the 10th National Conf. on Artificial Intelligence,' págs. 129–134.

Kittler, J. (1986). 'Feature selection and extraction.' En T. Y. Young y K. S. Fu, editores, 'Handbook of Pattern Recognition and Image Processing,' págs. 56–83. Academic Press Inc., Orlando, Florida.

Kodratoff, Y. (1988). *Introduction to Machine Learning.* Pitman Pub., London.

Kohavi, B. y Frasca, B. (1994). 'Useful feature subsets and rough set reducts.' En 'Proceedings of the Third International Workshop on Rough Set and Soft Computing (RCSSC-94),' págs. 310–317.

Kohavi, R. (1994). 'Feature subset selection as search with probabilistic estimates.' En 'AAAI Fall Symposium on Relevance,' págs. 122–126.

Kohavi, R. (1995*a*). 'A study of cross-validation and bootstrap for accuracy estimation and model selection.' En C. S. Mellish, editor, 'Procedings of the 14th International Joint Conference on Artificial Intelligence,' Morgan Kaufmann Publisher, Inc.

Kohavi, R. (1995*b*). *Wrappers for Performance Enhancement and Oblivious Decision Graphs.* Tesis Doctoral, Dept. of Computer Science. University of Standford.

Kohavi, R. y John, G. H. (1997). 'Wrappers for feature selection.' *Artificial Intelligence*, **97**, págs. 273–324.

Kohavi, R., Langley, P. y Yun, Y. (1997). 'The utility of feature weighting in nearest neighbor algorithms.' En M. va Someren y G. Widmer, editores, 'Poster Papers: 9th European Conference on Machine Learning,' Prague, Czech Republic. Unpublished.

Kohavi, R. y Provost, F. (1998). 'Glossary of terms.' *Machine Learning*, **30**, págs. 271–274.

Kohavi, R., Sommerfield, D. y Dougherty, J. (1996). 'Data mining using MLC++: A machine learning library in C++.' En 'Tools with Artificial Intelligence,' págs. 234–245. IEEE Computer Society Press.

Koller, D. y Sahami, M. (1996). 'Toward optimal feature selection.' En (Saitta, 1996), págs. 284–292.

Kononenko, I. (1994). 'Estimating attributes: Analysis and extensions of RELIEF.' En F. Bergadano y L. de Raedt, editores, 'Machine Learning: ECML-94,' págs. 171–182. Springer, Berlin.

Kononenko, I. (1995). 'On biases in estimating multi-valued attributes.' En 'Proceedings of the Int. Joint Conference on Artificial Intelligence (IJCAI-95),' Montreal, Canada.

Koza, J. R. (1992). *Genetic Programming. On the Programming of Computers by Means of Natural Selection.* MIT Press.

Koza, J. R., Bennett III, F. H., Andre, D. y Keane, M. A. (1996). 'Four problems for which a computer program evolved by genetic programming is competitive with human performance.' En 'Proceedings of the 1996 IEEE International Conference on Evolutionary Computation,' tomo 1, págs. 1–10. IEEE Press.

Lammens, J. M. (1994). *A Computational Model of Color Perception and Color Naming.* Tesis Doctoral, Faculty of the Graduate School of State, Univ. of New York at Buffalo.

Lancaster, P. y Tismenetsky, M. (1985). *The Theory of Matrices.* Academic Press Inc., segunda edición.

Langley, P. (1994). 'Selection of relevant features in machine learning.' En 'Procs. of the AAAI Fall Symposium on Relevance,' AAAI Press, New Orleans, LA.

Langley, P. (1996). *Elements of Machine Learning.* Morgan Kaufmann Publishers, Inc., San Francisco.

Langley, P. y Sage, S. (1994). 'Oblivious decision trees and abstract cases.' En 'Working Notes of the AAAI-94 Workshop on Case-Based Reasoning,' AAAI Press, Seattle, WA.

Lee, S. (1992). 'Supervised learning with gaussian potentials.' En B. Kosko, editor, 'Neural Networks for Signal Processing,' Prentice-Hall.

Lee, S. y Kil, R. M. (1988). 'Multilayer feedforward potential function network.' En 'Proceedings of 2nd. Int. Conference on Neural Networks,' tomo I, págs. 161–171.

Li, W. (1990). 'Mutual information functions versus correlation functions.' *Journal of Statistical Physics*, **60**(5/6), págs. 823–837.

Linsker, R. (1989). 'How to generate ordered maps by maximizing the mutual information between input and output signals.' *Neural Computation*, **1**(3), págs. 402–411.

Littlestone, N. (1988). 'Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm.' *Machine Learning*, **2**, págs. 285–318.

Littlestone, N., Long, P. M. y Warmuth, M. K. (1991). 'On-line learning of linear functions.' En 'Proceedings of the 23rd ACM Symposium on the Theory of Computing - STOC 91,' págs. 465–475.

Liu, H. y Motoda, H., editores (1998*a*). *Feature Extraction, Construction and Selection. A Data Mining Perspective.* Kluwer Academic Pub., Norwell, MA.

Liu, H. y Motoda, H. (1998*b*). *Feature Selection for Knowledge Discovery and Data Mining.* Kluwer Academic Publishers, Norwell, MA.

Liu, H., Motoda, H. y Dash, M. (1998). 'A monotonic measure for optimal feature selection.' En C. Nédellec y C. Rouveirol, editores, 'Proceedings of the 10th European Conference on Machine Learning (ECML-98),' tomo 1398 de *LNAI*, págs. 101–106. Springer, Berlin. ISBN 3-540-64417-2.

Liu, H. y Setiono, R. (1995). 'Chi2: Feature selection and discretization of numeric attributes.' En 'Proceedings of the 7th IEEE International Conference on Tools with Artificial Intelligence (TAI'95),' págs. 388–391. Washington DC.

Liu, H. y Setiono, R. (1996*a*). 'Dimensionality reduction via discretization.' En 'Proceedings of the 9th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems,' .

Liu, H. y Setiono, R. (1996*b*). 'Feature selection and classification - A probabilistic wrapper approach.' En 'Proceedings of the 9th International Conference on Industrial & Engineering Applications of AI & Expert Systems,' págs. 419–424. Fukunoka, Japan.

Liu, H. y Setiono, R. (1996*c*). 'A probabilistic approach to feature selection - A filter solution.' En (Saitta, 1996), págs. 319–327.

Liu, H. y Setiono, R. (1997). 'Feature selection via discretization.' *IEEE Trans. on Knowledge and Data Engineering*, **9**(4).

Liu, H. y Setiono, R. (1998*a*). 'Incremental feature selection.' *Applied Intelligence*, **9**(3), págs. 217–230.

Liu, H. y Setiono, R. (1998*b*). 'Some issues on scalable feature selection.' *Expert Systems with Application*, **15**, págs. 333–339.

López de Mántaras, R. (1991). 'A distance-based attribute selection measure for decision tree induction.' *Machine Learning*, **6**, págs. 81–92.

Lorenzo, J., Hernández, M. y Méndez, J. (1998*a*). 'Detection of interdependences in attribute selection.' *Lecture Notes in Computer Science*, **1510**, págs. 212–220.

Lorenzo, J., Hernández, M. y Méndez, J. (1998*b*). 'GD: A measure based on information theory for attribute selection.' En H. Coelho, editor, 'Proceedings of the 6th Ibero-American Conference on AI on Progress in Artificial Intelligence (IBERAMIA-98,' tomo 1484 de *LNAI*, págs. 124–135. Springer, Berlin. ISBN 3-540-64992-1.

MacKay, D. J. (1997). 'Information theory, inference and learning algorithms.' `http://wol.ra.phy.cam.ac.uk/mackay/itprnn/book.ps.gz`.

Mangasarian, O. L. y Wolberg, W. H. (1990). 'Cancer diagnosis via linear programming.' *SIAM News*, **23**(5), págs. 1–18.

Martín Bautista, M. J. y Villa, M. A. (1999). 'A survey of genetic feature selection in mining issues.' En P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao y A. Zalzala, editores, 'Proceedings of the Congress on Evolutionary Computation,' tomo 2, págs. 1314–1321. IEEE Press, Mayflower Hotel, Washington D.C., USA.

McCulloch, W. S. y Pitts, W. (1943). 'Anlogical calculus of the ideas immanent in nervous activity.' *Bulletin of Mathematical Biophysics*, **5**, págs. 115–137.

McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition.* Wiley Series in Probability and Mathematical Statistics. John Wiley and sons.

Méndez, J., Falcón, A., Hernández, F. M. y Cabrera, J. (1994). 'Development tool for computer vision at pixel level.' *Cybernetics and Systems*, **25**(2), págs. 289–319.

Michalski, R. S. (1983). 'A theory and methodology of inductive learning.' *Artificial Intelligence*, **20**, págs. 111–116.

Michalski, R. S. y Chilausky, R. (1980). 'Knowledge acquisition by encoding expert rules versus computer induction from examples: A case study involving soybean pathology.' *International Journal of Man-Machine Studies*, **12**(1), págs. 63–87.

Michaslki, R. S. y Stepp, R. E. (1983). 'Learning from observation: Conceptual clustering.' En R. S. Michalski, J. G. Carbonell y T. M. Mitchell, editores, 'Machine Learning: An Artificial Intelligence Approach,' págs. 331–364. Springer Verlag.

Milosavljevic, A. (1995). 'Discovering dependencies via algorithmic mutual information: a case study in DNA sequence comparison.' *Machine Learning*, **21**, págs. 35–50.

Minton, S. (1984). 'Constraint-based generalization.' En 'Proceedings of the AAAI-84,' págs. 251–254.

Minton, S., Carbonell, J. G., Knoblock, C. A., Kuokka, D. R., Etzioni, O. y Gil, Y. (1989). 'Explanation-based learning: A problem solving perspective.' *Artificial Intelligence*, **40**, págs. 63–118.

Mitchell, T., Caruana, R., Freitag, D., McDermott, J. y Zabowski, D. (1994). 'Experience with a learing personal assistant.' *Communications of the ACM, Special Issue on Intelligent Agents*, **37**(7), págs. 88–91.

Mitchell, T. M. (1980). 'The need for biases in learning generalizations.' Informe Técnico CB-TR-117, Rutgers University. Reeditado en Machine Learning, Jude W. Shavlik and Thomas G. Dietterich, editores, 1990.

Mitchell, T. M. (1997). *Machine Learning.* McGraw-Hill. ISBN 0-07-042807-7.

Mitchell, T. M., Keller, R. y Kedar-Cabelli, S. (1986). 'Explanation-based generalization: A unifying view.' *Machine Learning*, **1**, págs. 47–80.

Mladenić, D. (1998). 'Feature subset selection in text learning.' En C. Nédellec y C. Rouveirol, editores, 'Proceedings of the 10th European Conference on Machine Learning (ECML-98),' tomo 1398 de *LNAI*, págs. 95–100. Springer, Berlin. ISBN 3-540-64417-2.

Mladenić, D. y Grobelnik, M. (1999). 'Feature selection for unbalanced class distribution and naive bayes.' En I. Bratko y S. Dzeroski, editores, 'Proceedings of ICML-99, 16th International Conference on Machine Learning,' págs. 258–267. Morgan Kaufmann Publishers, San Francisco, US.

Moddemeijer, R. (1989). 'On estimation of entropy and mutual information of continuous distributions.' *Signal Processing*, **16**, págs. 233–248.

Moddemeijer, R. (1999). 'A statistic to estimate the variance of the histogram-based mutual information estimator based on dependent pairs of observations.' *Signal Processing*, **75**, págs. 51–63.

Moghaddam, B. y Pentland, A. (1995). 'Probabilistic visual learning for object detection.' En 'Proceedings of the 5th Int. Conf. on Computer Vision,' Cambridge.

Moody, J. y Darken, C. J. (1989*a*). 'Fast learning in networks of locally-tuned processing units.' *Meural Computation*, **1**, págs. 281–294.

Moody, J. y Darken, C. J. (1989*b*). 'Fast learning in networks of locally-tuned processing units.' *Neural Computation*, **1**, págs. 281–294.

Moon, Y.-I., Rajagopalan, B. y Lall, U. (1995). 'Estimation of mutual information using kernel density estimators.' *Physical Review E*, **52**(3), págs. 2318–2321.

Mooney, R. y DeJong, G. F. (1985). 'Learning schemata for natural language processing.' *IJCAI-85*, **1**, págs. 681–687.

Moore, A. W. y Lee, M. S. (1994). 'Efficient algorithms for minimizing cross validation error.' En (Cohen y Hirsh, 1994), págs. 190–198.

Musavi, M. T., Cha, K. H., Hummels, D. M. y Kalantri, K. (1994*a*). 'In the generalization ability of neural networks classifiers.' *IEEE Trans. on Pattern Analysis and Machine Learning*, **16**(6), págs. 659–663.

Musavi, M. T., Chan, K. H., Hummels, D. M. y Kalantri, K. (1994*b*). 'On the generalization ability of neural networks classifiers.' *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **16**(4), págs. 659–663.

Musen, M. y Van der Lei, J. (1988). 'Of brittleness and bottlenecks: Challenges in the creation of pattern-recognition and expert-system models.' En E. S. Gelsema y L. N. Kanal, editores, 'Pattern Recognition and Artificial Intelligence,' págs. 335–352.

Nandhakumer, N. y Aggarwal, J. K. (1985). 'The artificial intelligence approach to pattern recognition: A perspective and an overview.' *Pattern Recognition*, **18**(6), págs. 383–389.

Narendra, P. y Fukunaga, K. (1977). 'A branch and bound algorithm for feature selection.' *IEEE Trans. on Computers*, **26**, págs. 917–922.

Natarajan, B. K. (1991). *Machine Learning, A Theoretical Approach.* Morgan Kaufmann, San Mateo, CA.

Nayar, S. K. y Poggio, T. (1996). *Early Visual Learning.* Oxford University Press, New York.

Niemann, H., Brünig, H., Salzbrunn, R. y Scröder, S. (1990). 'A knowledge-based vision system for industrial applications.' *Machine Vision and Applications*, **3**, págs. 201–229.

Pellegretti, P., Roli, F., Serpico, S. B. y Vernazza, G. (1992). 'A system for learning of descriptions for image recognition purposes.' Informe Técnico D.I.B.E.-LR92, Dept. of Biophysical and Electronic Engineering. Univ. of Genoa.

Poggio, T. y Girosi, F. (1990). 'Networks for approximation and learning.' *Proceedings of the IEEE*, **78**, págs. 1481–1487.

Pomerleau, D. A. (1993). *Neural Network Perception for Mobile Robot Guidance.* Kluwer, Dordrecht, The Netherlands.

Popper, K. (1959). *The Logic of Scientific Discovery.* Basic Books, New York.

Pudil, P., Novovicova, J. y Kittler, J. (1994). 'Floating search methods in feature selection.' *Pattern Recognition Letters*, **15**, págs. 1119–1125.

Puterman, M. L. (1994). *Markov Decision Processes - Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., New York, NY.

Quinlan, J. R. (1986). 'Induction of decision trees.' *Machine Learning*, **1**, págs. 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kauffman Pub., Inc., Los Altos, California.

Rauber, T. W. y Steiger-Garção, A. S. (1993). 'Feature selection of categorical attributes based on contingency tables analysis.' En 'Proceedings of the 5th Portuguese Conference on Pattern Recognition,' Porto, Portugal.

Renals, S. y Rohwer, R. (1989). 'Phoneme classification experiments using radial basis functions.' En 'Proceedings of Int. Joint Conference on Neural Networks I,' págs. 416–467.

Richards, J. A. (1986). *Remote Sensing Digital Image Analysis: An Introduction*, capítulo Feature Reduction, págs. 206–224. Springer Verlag.

Ritter, G. L., Woodruff, H. B., Lowry, S. R. y Isenhour, T. L. (1975). 'An algorithm for a selective nearest neigbour decision rule.' *IEEE Trans. on Information Theory*, **21**, págs. 665–669.

Rosenblatt, F. (1958). 'The perceptron: A probabilistic model for information storage and organization in the brain.' *Psychological Review*, págs. 386–408.

Rosenblatt, F. (1960). 'On the convergence of reinforcement procedures in simple perceptrons.' Informe Técnico VG-1196-G-4, Cornell Aeronautical, Buffalo, NY.

Rumelhart, D. E., Hinton, G. E. y Williams, R. J. (1986*a*). 'Learning internal representation by error propagation.' En D. E. Rumelhart y J. L. McClelland, editores, 'Parallel Distributed Processing: Explorations in the Microstructures of Cognition,' tomo 1, págs. 318–362. MIT Press, Cambridge: MA.

Rumelhart, D. E., Hinton, G. E. y Williams, R. J. (1986*b*). 'Learning representation by back-propagation errors.' *Nature*, **323**, págs. 533–536.

Saitta, L., editor (1996). *Proceedings of the 13th International Conference on Machine Learning*. Morgan Kaufmann.

Salzberg, S. L. (1997). 'On comparing classifiers: Pitfalls to avoid and a recommended approach.' *Data Mining and Knowledge Discovery*, **1**(3), págs. 317–327.

Samet, H. (1990). *The Design and Analysis of Spatial Data Structures*. Addison-Wesley.

Sánchez, J. S., Pla, F. y Ferri, F. J. (1997). 'On the equivalency between decision tree classifiers and the nearest neigbour rule.' En V. Botti, editor, 'Actas de la VII Conferencia Española para la Inteligencia Artificial,' págs. 197–206. Málaga.

Sánchez García, M. (1978). *Modelos Estadísticos Aplicados a Tratamiento de Datos*, capítulo Análisis de Componentes Principales, págs. 51–71. Univ. Complutense de Madrid, Centro de Cálculo, Madrid.

Scherf, M. y Brauer, W. (1997). 'Improving RBF networks by the feature selection approach EUBAFES.' En 'Proceedings of the 7th Int. Conference on Artificial Neural Networks (ICANN'97),' págs. 391–396. Laussane, Switzerland.

Schölkopf, B., Burges, C. J. y Smola, A. J., editores (1999). *Advances in Kernel Methods - Support Vector Learning*. The MIT Press, Cambridge, Massachusetts.

Setiono, R. y Liu, H. (1996). 'Improving backpropagation learning with feature selection.' *Applied Intelligence*, **6**, págs. 129–139.

Setiono, R. y Liu, H. (1997). 'Neural-network feature selector.' *IEEE Trans. on Neural Networks*, **8**(3), págs. 654–662.

Shannon, C. E. (1948). 'A mathematical theory of communication.' *Bell System Technical Journal*, **27**, págs. 379–423,623–656.

Shapiro, S. C., editor (1992). *Encyclopedia of Artificial Intelligence.* John Wiley & sons, Inc., segunda edición.

Shepard, R. N. (1987). 'Toward a universal of generalization for psychological science.' *Science*, **237**, págs. 1317–1323.

Siedlecki, W. y Sklansky, J. (1988). 'On automatic feature selection.' *Int. Journal of Pattern Recognition and Artificial Intelligence*, **2**(2), págs. 197–220.

Siedlecki, W. y Sklansky, J. (1989). 'A note on genetic algorithms for large-scale feature selection.' *Pattern Recognition Letters*, **10**, págs. 335–347.

Simon, H. A. (1983). 'Why should machines learn?' En R. S. Michalki, J. G. Carbonell y T. M. Mitchell, editores, 'Machine Learning: An artificial intelligence approach,' tomo I. Morgan Kaufmann.

Skalak, D. B. (1994). 'Prototype and feature selection by sampling and random mutation hill climbing algorithms.' En (Cohen y Hirsh, 1994), págs. 293–301.

Spath, H. (1980). *Cluster analysis algorithms: For data reduction and classification of objects.* Ellis Horwood Limited.

Sutton, R. S. (1988). 'Learning to predict by the method of temporal differences.' *Machine Learning*, **3**(1), págs. 9–44.

Talavera, L. y Béjar, J. (1998). 'Efficient construction of comprehensible hierarchical clustering.' *Lectures Notes in Artificial Intelligence*, **1510**, págs. 93–101.

Teller, A. y Veloso, M. (1994). 'PADO: A new learning architecture for object recognition.' En K. Ikeuchi y M. Veloso, editores, 'Symbolic Visual Learning,' págs. 81–116. Oxford University Press, Oxford, England.

Thrun, S. B. (1991). 'The monk's problem: A performance comparison of different learning algorithms.' Informe Técnico CMU-CS-91-197, Carnegie Mellon Univ.

Torkkola, K. y Campbell, W. (2000). 'Mutual information in learning feature transformations.' En P. Langley, editor, 'Proceedings of the Seventeenth International Conference on Machine Learning,' págs. 1015–1022. Morgan Kaufmann, San Francisco, CA.

Tou, J. T. y Gonzalez, R. C. (1974). *Pattern Recognition Principles.* Addison-Wesley Publishing Co.

Toussaint, G. T., Bhattacharya, B. K. y Poulsen, R. S. (1984). 'The application of voronoi diagrams to nonparametric decision rules.' En 'Proceedings of Computer Science and Statistics,' págs. 97–108.

Vafaie, H. y De Jong, K. (1993). 'Robust feature selection algorithms.' En 'Proceedings of the 5th IEEE International Conference on Tools for Artificial Intelligence,' págs. 356–363. IEEE Press.

Vafaie, H. y De Jong, K. (1994). 'Improving a rule induction system using genetic algorithms.' En R. S. Michalski y G. Tecuci, editores, 'Machine Learning: A Multistrategy Approach,' tomo IV. Morgan Kaufmann, San Mateo, CA.

Vafaie, H. y Imam, I. F. (1994). 'Feature selection methods: Genetic algorithms vs. greedy-like search.' En 'Proceedings of the 3rd International Conference on Fuzzy Systems and Intelligent Control,' .

Valiant, L. G. (1984). 'A theory of the learnable.' *Communications of the ACM*, **27**(11), págs. 1134–1142.

Vernazza, G. (1991). 'From numerical to symbolic image processing: Integration of computational and knowledge-based approaches.' En S.-V. ESPRIT Basic Research, editor, 'Computer Vision: Croaft, Engineering, and Science,' Berlin.

Viola, P. A. (1995). 'Alignment by maximization of mutual information.' Informe Técnico AITR-1548, Massachusetts Institute of Technology.

Viola, P. A., Schraudolph, N. N. y Sejnowski, T. J. (1995). 'Empirical entropy manipulation for real-world problems.' En M. M. David S. Touretzky y M. Perrone, editores, 'Advances in Neural Information Processing,' tomo 8. MIT Press, Cambridge, Denver.

Wang, H. (1996). *Towards a Unified Framework of Relevance.* Tesis Doctoral, School of Information and Software Engineering. Univ. of Ulster.

Wang, H., Bell, D. y Murtagh, F. (1998). 'Relevance approach to feature subset selection.' En (Liu y Motoda, 1998*a*).

Wang, H., Bell, D. y Murtagh, F. (1999). 'Axiomatic approach to feature subset selection based on relevance.' *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **21**(3), págs. 271–277.

Wang, K. y Sundaresh, S. (1998). 'Selecting features by vertical compactness of data.' En (Liu y Motodà, 1998*a*).

Watkings, C. J. y Dayan, P. (1992). 'Q-learning.' *Machine Learning*, **8**(3), págs. 279–292.

Watrous, R. (1987). 'Learning algorithms for connectionist networks: Applied gradient methods of nonlinear optimization.' En M. Caudill y C. Butler, editores, 'IEEE First International Conference on Neural Networks,' tomo 2, págs. 619–627. San Diego, CA.

Weiss, S. M. (1991). 'Small sample error rate estimation for k-NN classifiers.' *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **13**(3), págs. 285–289.

Weiss, S. M. y Kulikowski, C. A. (1991). *Computer Systems that Learn.* Morgan Kaufmann Publishers, Inc., San Francisco, CA.

Wettschereck, D. y Dietterich, T. G. (1995). 'An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms.' *Machine Learning*, **19**, págs. 5–27.

White, A. P. y Liu, W. Z. (1994). 'Bias in information-based measures in decision tree induction.' *Machine Learning*, **15**, págs. 321–329.

Widrow, B. (1962). 'Generalization and information storage in networks of adaline.' En M. Yovits, G. Jacobi y G. Goldstein, editores, 'Self-organizing systems,' Spartan Books.

Widrow, B. y Hoff, M. (1960). 'Adaptive circuit switching.' En 'IRE WESCON Convention Record,' tomo 4, págs. 96–104. New York: IRE.

Winston, P. H. (1980). 'Learning and reasoning by analogy.' *Communications of the ACM*, **23**(12), págs. 689–702.

Yang, H. H. y Moody, J. (1999). 'Feature selection based on joint mutual information.' En 'Advences in Intelligent Data Analysis (AIDA), Computational Intelligence Methods and Applications (CIMA),' Rochester New York.

Yang, J. y Honavar, V. (1998). 'Feature subset selection using a genetic algorithm.' En (Liu y Motoda, 1998*a*).

Yang, J., Parekh, R. y Honovar, V. (1998). 'DistAI: An intern-pattern distance-based constructive learning algorithm.' En 'Proceedings of the International Joint Conference on Neural NNetoworks,' Anchorage, Alaska.

Yao, Y. Y., Wong, S. K. M. y Butz, C. J. (1999). 'On information-theoretic measures of attribute importance.' En N. Zhong y L. Zhou, editores, 'Proceedings of the 3rd Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining (PAKDD-99),' tomo 1574 de *LNAI*, págs. 133–137. Springer, Berlin. ISBN 3-540-65866-1.

Zadeh, L. A. (1971). 'Quantitative fuzzy semantics.' *Information Sciences*, **3**, págs. 159–176.

Zhang, B. (1994). 'Accelerated learning by active example selection.' *International Journal of Neural Networks*, **5**(1), págs. 67–75.