Razonamiento Basado en Casos aplicado a Problemas de Clasificación

Manuel Laguía Bonillo

Tesis Doctoral
Programa: “Tratamiento de la Información en Inteligencia Artificial”
Dirigida por: Dr. Juan Luis Castro Peña

Septiembre 2003

Índice General

Índice de Tablas

Índice de Figuras

Capítulo 1

Introducción

1.1 Objetivos

En este trabajo se pretende estudiar el comportamiento y las posibilidades que ofrece el Razonamiento Basado en Casos (RBC) cuando se utiliza en problemas de clasificación. Y más en concreto el trabajo se va a centrar en estudiar el uso de distancias y de métodos de clasificación basados en distancias.

Se pretende comparar los clasificadores basados en distancias entre sí y con otros métodos para conocer la utilidad de estas herramientas. Por este motivo se han elegido dos Bases de Datos, una simple como es la Iris y una compleja como la PIMA. Así podremos observar el comportamiento de los clasificadores en un dominio relativamente sencillo y en un dominio muy dificil.

Se pretende realizar un estudio sobre el comportamiento con estas Bases de Datos de los diferentes clasificadores basados en distancias.

En primer lugar se pretende fijar el marco teórico en el que nos vamos a mover. Se estudiarán diversas formas de definir distancias y se analizará su comportamiento en diversas circunstancias.

A continuación se definirán varios métodos de clasificación basados en distancias, de manera que puedan utilizar cualquiera de las distancias anteriormente definidas. Se comenzará con métodos que han sido descritos en la literatura, pero se pretende definir métodos nuevos que sean aportaciones originales.

Así se podrá comparar los resultados que obtienen los métodos nuevos con los que obtienen los métodos de la literatura.

Se analizarán en detalle los resultados que ofrecen los distintos métodos, y se pretende estudiar: la utilidad de los métodos basados en distancias en general y de los distintos métodos en particular, la influencia que tienen los parámetros que controlan los métodos sobre su rendimiento, explicar el comportamiento que muestran los distintos métodos, obtener pautas y criterios para elegir los parámetros de los métodos, estudiar la influencia que tiene la Base de Casos, estudiar la influencia del tamaño del conjunto de entrenamiento, obtener pautas y criterios para elegir el método más adecuado para un problema de clasificación concreto,

Para estudiar la utilidad de los métodos de clasificación basados en distancias se pretende comparar sus resultados con los que obtienen otros métodos de clasificación recogidos en la literatura.

El planteamiento de este trabajo creemos que está suficientemente justificado debido a que no conocemos la existencia de ningún estudio serio y completo sobre el tema. Ninguno de los trabajos que conocemos analiza en detalle las diferentes posibilidades para definir una distancia, define varias distancias diferentes y estudia el comportamiento de esas distancias. Tampoco son corrientes trabajos que planteen y definan varios métodos de clasificación basados en distancias, y estudie en profundidad su comportamiento.

1.2 Metodología de las Pruebas

El objetivo de las pruebas que se han realizado es evaluar el comportamiento de distintos clasificadores basados en distancias que se han analizado. Las pruebas se han realizado utilizando las Bases de Casos descritas en el apartado 1.2.1. En conjunto se ha usado un gran número de bases de casos para estudiar el comportamiento de

los diferentes clasificadores, y se han realizado las pruebas usando 10 Validación Cruzada (10–CV) [WK91].

Tal y como se menciona en el capítulo 2, un aspecto bastante importante del Razonamiento Basado en Casos es el aprendizaje. El sistema interacciona con el entorno y tiene una forma de evaluar sus decisiones. Esto hace posible que el sistema juzgue la bondad de las soluciones que propone y así pueda en el futuro anticipar y evitar los errores que ya ha cometido antes, y volver a utilizar las soluciones que han dado buenos resultados.

En el problema concreto de la clasificación también nos podemos plantear que el clasificador realice aprendizaje durante la fase de clasificación. Lo que se haría en ese tipo de pruebas es presentar el caso al clasificador para que le asigne una clase, y después se informa al clasificador de la clasificación correcta para que sepa si la clasificación que hizo era correcta o no. Así se consigue que el clasificador tenga una realimentación y que mejore su comportamiento debido a dos razones:

- El razonador es más experto y conoce más casos. Por lo tanto al aumentar su conocimiento y el conjunto de casos conocidos podrá recuperar casos que sean más parecidos a un caso nuevo
- Cuando el RBC comete un error lo incorpora a la librería de casos y puede tenerlo en cuenta para no repetirlo.

En las pruebas que se han realizado no se ha permitido que el clasificador tenga realimentación, es decir, que conozca a posteriori cual es la clasificación correcta. Se ha hecho así porque normalmente no se emplea realimentación en este tipo de pruebas y comparaciones. No obstante, el software que hemos desarrollado está preparado para permitir realimentación, y se puede utilizar en trabajos futuros si se pretende estudiar cómo se comporta el clasificador si recibe información del entorno y averiguar si es capaz de adaptarse con éxito al medio.

Las operaciones que se realizan en las pruebas se muestra en forma de algoritmo en la figura 1.1.

El hecho de que aparezcan algunos de los ejemplos de prueba como no clasificados puede ser debido a que el clasificador no tiene información suficiente y no

1. Dividir la Base de Casos *BC* en 10 conjuntos disjuntos del mismo tamaño $S_1, S_2, \ldots, S_{10}$.

2. Desde *i* = 1 hasta 10 // 10-CV Para prueba de los clasificadores

   2.1 Formar los conjuntos de entrenamiento y prueba como $Entren = BC - S_i$, $Prueba = S_i$.

   2.2 Dividir el conjunto de entrenamiento *Entren* en 10 conjuntos disjuntos del mismo tamaño $S\prime_1, S\prime_2, \ldots, S\prime_{10}$.

   2.3 Desde *j* = 1 hasta 10 // 10-CV Para estimación de parámetros

      2.3.1 Formar los conjuntos de entrenamiento y prueba para la estimación de parámetros como $Entren_{estim} = Entren - S\prime_j$, $Prueba_{estim} = S\prime_j$.

      2.3.2 Proporcionar el Conjunto de Entrenamiento $Entren_{estim}$ al Clasificador para que éste realice la fase de aprendizaje.

      2.3.3 Para cada uno de los ejemplos del Conjunto de Prueba $Prueba_{estim}$

      Usar el clasificador para clasificar el ejemplo

      Comparar la clasificación real con la que ha proporcionado el clasificador y actualizar los valores estadísticos de la estimación de parámetros

3. Proporcionar el Conjunto de Entrenamiento *Entren* al Clasificador para que éste realice la fase de aprendizaje.

4. Para cada uno de los ejemplos del Conjunto de Prueba *Prueba*

   Usar el clasificador para clasificar el ejemplo

   Comparar la clasificación real con la que ha proporcionado el clasificador y actualizar los valores estadísticos de la fase de prueba

Figura 1.1: Algoritmo utilizado en las pruebas de los clasificadores

ha podido clasificarlo, o porque el clasificador es conservador y no se ha atrevido a emitir una clasificación que considera arriesgada. Hay que tener en cuenta que normalmente es preferible no clasificar un ejemplo a clasificarlo mal, aunque todo depende del problema concreto. Un clasificador demasiado conservador y que no se arriesga puede tener tasas de error relativamente bajas, pero a cambio de que la tasa de acierto probablemente también sea más baja que la de otro clasificador más arriesgado. Como siempre, dependiendo del dominio concreto de la aplicación será más interesante un clasificador u otro.

Hay que tener en cuenta que en una misma serie de pruebas todos los clasificadores se entrenan y prueban con los mismos datos. Se realizan las mismas particiones de acuerdo a la metodología 10–CV y en cada momento se proporciona a los clasificadores los mismos ejemplos para que realice el entrenamiento y posteriormente se prueba con el mismo conjunto de ejemplos. Por tanto, durante una misma serie de pruebas, *todos los clasificadores son entrenados y probados con exactamente los mismos ejemplos*.

Por tanto, cuando un clasificador muestre un porcentaje de acierto superior a otro, no es porque haya tenido un conjunto de ejemplos más sencillo de clasificar. Sino porque efectivamente ese clasificador es capaz de clasificar correctamente mayor número de ejemplos que el otro, porque ambos se encontraban en situaciones idénticas, entrenando y clasificando con los mismos ejemplos.

Además, esto permite que se puedan realizar comparaciones rigurosas entre los clasificadores y emplear comprobaciones estadísticas empleando el test t–Student para analizar cuándo las diferencias en los resultados obtenidos por distintos clasificadores pueden explicarse símplemente por el azar y la elección concreta de ejemplos de entrenamiento y prueba; y cuándo esas diferencias sólo pueden ser debidas a que realmente existe una diferencia entre los clasificadores, independientemente de las particiones 10–CV que se realicen. Por eso *se han probado todos los clasificadores con exactamente los mismos ejemplos y se ha realizado una prueba t–Student pareada con dos colas con un nivel de significación del 95% para comparar los resultados de los distintos clasificadores*.

Tabla 1.1: Bases de casos del UCI–Repository usadas en las pruebas.

					Nº de atributos	
Índice	Código	Dominio	Tamaño	Nº Clases	Numéricos	Simbólicos
1	IR	Iris Plant	150	3	4	0
2	WI	Wine Recognition	178	3	13	0
3	PI	PIMA Diabetes	768	2	8	0
4	GL	Glass Identification	214	6	9	0
5	CL	Cleveland	303	5	5	8
6	GD	Granada Digits	1000	10	256	0
7	SN	Sonar	208	2	60	0
8	LD	Liver Disorder	345	2	6	0
9	ZO	Zoo	101	7	1	15
10	TT	Tic–Tac–Toe	958	2	0	9
11	L7	Led 7	5000	10	0	7
12	L24	Led 24	5000	10	0	24
13	W21	WaveForm–21	5000	3	21	0
14	W40	WaveForm–40	5000	3	40	0
15	F1	Solar Flare 1	1066	8	0	10
16	F2	Solar Flare 2	1066	6	0	10
17	F3	Solar Flare 3	1066	3	0	10
18	SO	Soybean	47	4	35	0
19	LR	Letter Recognition	20000	26	16	0

1.2.1 Las Bases de Casos

Se han empleado 68 bases de casos: 18 bases bastante utilizadas del UCI–Repository [BM98], una versión reducida de 1,000 ejemplos de la Granada Handwritten Digits[1] (tabla 1.1), y 49 bases sintéticas.

Las bases del UCI–Repository son usadas con frecuencia en la literatura científi-

[1]La base de casos Granada Handwritten Digits tiene en total 11,000 ejemplos, cada uno con 256 atributos numéricos (los valores normalizados de una rejilla de tamaño 16×16), y 10 clases (los dígitos 0,1,2,...,9). Cada ejemplo corresponde a un dígito manuscrito. Esta base de casos es privada y ha sido facilitada por IPSA (Investigación y Programas S.A.).

fica, lo que facilita las comparaciones con los resultados experimentales obtenidos por otros clasificadores introducidos en otros artículos, y las bases sintéticas son útiles para estudiar los clasificadores en un entorno controlado.

Las bases de casos sintéticas se han construido *ad hoc* sobre el cuadrado unidad $[0,1]\times[0,1]$, cada una con 500 ejemplos. Como queremos estudiar la influencia de la distribución de clases de la base de casos, hemos considerado (figuras 1.2 y 1.3):

Bandas (5,10,20): la clase de los puntos se asigna de acuerdo a 5, 10 ó 20 bandas horizontales y una clase por banda. Todas las bandas tienen la misma anchura.

Gauss: la clase de los puntos se asigna de acuerdo a cuatro distribuciones de Gauss centradas en (0.25, 0.25), (0.25, 0.75), (0.75, 0.25) y (0.75, 0.75) y con varianza 0.025.

Anillos con área constante (3,6,9): el espacio se divide en 3, 6 ó 9 anillos concéntricos con área igual y una clase por anillo. Para que las áreas de todos los anillos sean iguales, los radios deben ser diferentes (realmente son diferentes las diferencias entre radios de anillos consecutivos). El área total de las regiones no tiene influencia y permite estudiar la influencia de la forma y el número de clases.

Anillos con radio constante (3,6,9): el espacio se divide en 3, 6 ó 9 anillos concéntricos con radios iguales y una clase por anillo.

Senos (3,6,9): Hay dos clases y la frontera de decision es una curva sinusuidal con 3, 6 ó 9 intervalos $[0,2\pi]$ ajustados en $[0,1]\times[0,1]$.

Cuadrados (2,4,6,8): el espacio se divide en una rejilla de tamaño 2×2, 4×4, 6×6, u 8×8. Todas las variantes tienen cuatro clases con la misma cantidad de espacio, por tanto el área total de las clases no tiene influencia.

Cuando empleamos la denominación de “bases de casos sintéticas” para estas bases de casos debe tenerse en cuenta que esto no quiere decir que las bases del UCI–Repository no sean sintéticas. Algunas bases del UCI–Repository son sintéticas, como por ejemplo las bases LED7 y LED24, y otras no, como por ejemplo la

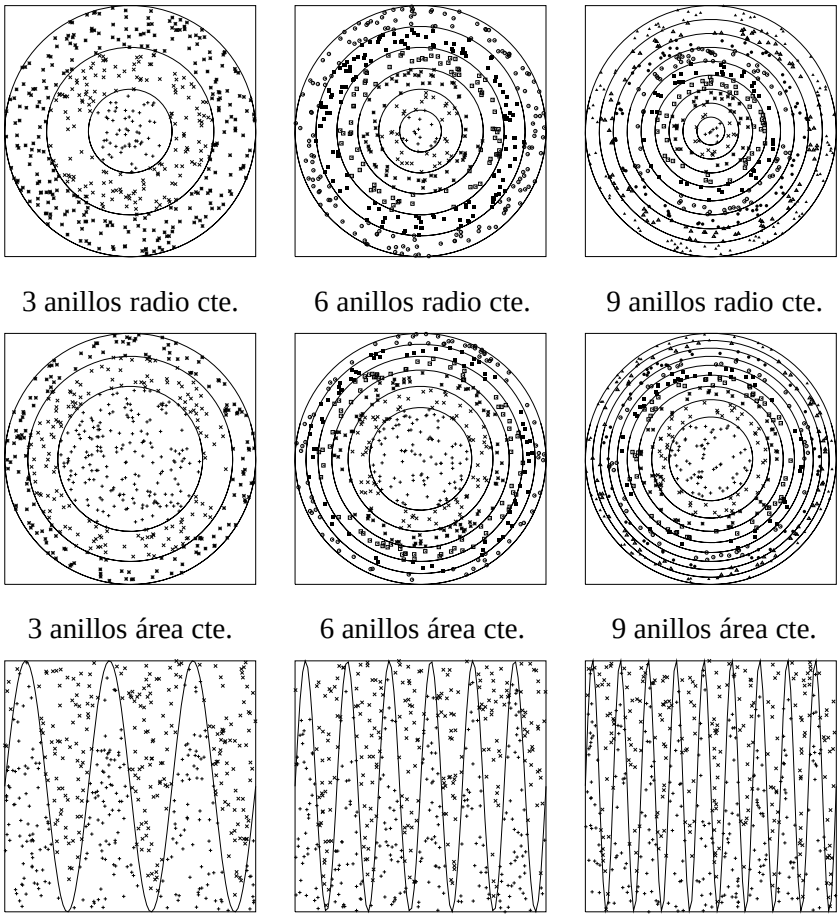


Figura 1.2: Bases de casos sintéticas: anillos con radio constante, anillos con área constante y senos. Símbolos diferentes indican clases diferentes. Las líneas representan las fronteras de decisión.

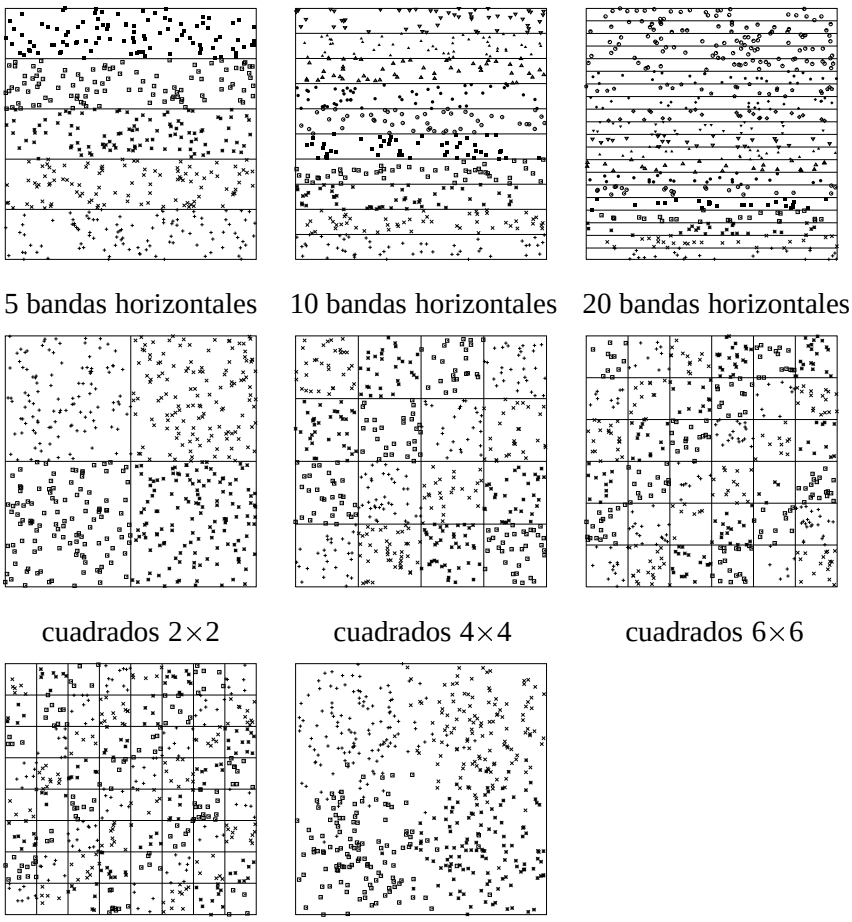
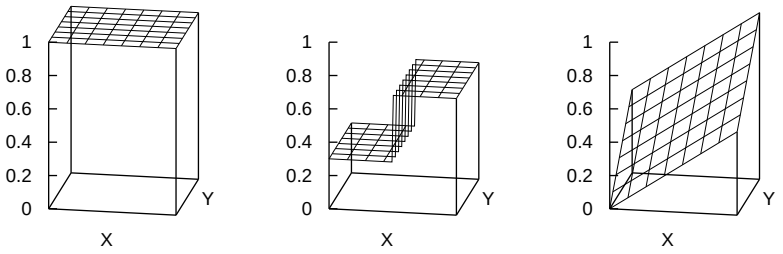


Figura 1.3: Bases de casos sintéticas: bandas, cuadrados y Gauss. Símbolos diferentes indican clases diferentes. Las líneas representan las fronteras de decisión.



a. Uniforme b. Mitad c. Progresiva

Figura 1.4: Distribución de puntos en las tres variantes de las bases de casos sintéticas.

base PIMA. A lo largo de este trabajo empleamos el término de "bases sintéticas" para referirnos al conjunto de bases de casos sintéticas que se han construido *ad hoc* para este estudio.

En sentido opuesto, se usa el término "bases del UCI–Repository" para referirnos al conjunto de bases de casos descargadas desde ese servidor de Internet, y que han sido utilizadas ampliamente en la literatura científica. A pesar de esto, cuando nos referimos a "bases del UCI–Repository" también solemos englobar la base *Granada Handwritten Digits* porque, aunque no pertenece al UCI–Repository, tampoco es una base sintética, y en cambio, es una base del estilo de las que existen en este sitio de Internet.

Se han generado tres variantes de todas las bases de datos sintéticas (excepto por supuesto para la base Gauss) para estudiar cómo se ven afectados los métodos por el hecho de que la densidad de puntos varíe. En estas tres variantes, los puntos se distribuyen con probabilidades diferentes a lo largo del espacio (fig. 1.4). Así se consigue un espacio con densidad de puntos diferente. Estas variantes son:

- *Uniforme*: los puntos se distribuyen uniformemente a lo largo del espacio.
- *Mitad*: el 30% de los puntos se encuentran en la mitad izquierda del espacio ($x < 0.5$) y el restante 70% en la mitad derecha ($x \geq 0.5$): los puntos están distribuidos en dos regiones claramente diferenciadas.
- *Progresiva*: la probabilidad de aceptación de un punto es proporcional a la

suma de sus coordenadas ($x+y$): la densidad de puntos se incrementa progresivamente desde la esquina inferior izquierda hasta la superior derecha.

1.2.2 Software Utilizado

Para la realizar de las pruebas y obtener resultados sobre el comportamiento de diferentes clasificadores se ha implementado un programa en C++. Este programa ha sido compilado utilizando el programa "gcc", y se ha obtenido como resultado un ejecutable bajo el sistema operativo Linux.

Para realizar los cálculos de los hiperplanos que se utilizan en los clasificadores del capítulo 5 se ha utilizado código escrito para MATLAB. Esta ha sido la forma más cómoda de trabajar para realizar pruebas previas y estimaciones sobre el comportamiento de las bandas y la forma en que afectan los distintos parámetros y decisiones de diseño que se han debido tomar. Después, con todo implementado en MATLAB, la forma más cómoda y directa de trabajar ha sido usar MATLAB para aprender los hiperplanos y el programa escrito en C++ para realizar las pruebas. Con esta excepción, todas las pruebas se han realizado con el programa desarrollado a propósito para este trabajo.

Desde el principio se planteó que el programa escrito en C++ debía ser una herramienta que facilitara la experimentación con distintos métodos de clasificación y con diferentes Bases de Casos. La idea original era crear una primera versión del programa con la funcionalidad básica para llevar a cabo la experimentación y comenzar a trabajar, y poco a poco ir añadiéndole nuevas capacidades que permitieran realizar más pruebas y obtener más resultados, así como facilitar la interacción del usuario hasta crear una interfaz cómoda. El programa ha ido sufriendo una paulatina pero constante evolución que todavía no ha concluído, ya que se pretende continuar el trabajo que se presenta aquí.

Conforme se han ido añadiendo más capacidades y funciones al programa han quedado patentes las limitaciones de la implementación original que aportaba la funcionalidad básica para empezar a experimentar. En el estado actual de la implementación y de cara a una posible difusión sería necesaria la reestructuración de la interfaz de usuario y en concreto añadir la posibilidad de utilización mediante un

sistema de ventanas.

Otro aspecto importante de la implementación era que el software debía permitir añadir de forma sencilla nuevas Bases de Casos y nuevos métodos de clasificación. Esto se consigue en la implementación actual editando ficheros donde se ha aislado las definiciones y el código que dependen de ellos y recompilando. Está previsto que en una versión futura se integren las labores de definición y modificación de los clasificadores y de declaración de Bases de Casos dentro del entorno de trabajo.

Para hacer la implementación independiente del clasificador que se utilice, se ha aislado las funciones básicas que necesita un clasificador genérico para trabajar. Se ha utilizado orientación a objetos para implementar cada clasificador como un objeto, y cada una de sus funciones básicas como un método de ese objeto. Es la implementación concreta de estos métodos lo que determina el comportamiento del clasificador. El lenguaje utilizado es C++, porque proporciona una gran flexibilidad y eficiencia a la hora de definir el comportamiento de los clasificadores. Todo el código donde se recoge la implementación de los clasificadores está aislado en un fichero.

Los métodos que se utilizan para implementar el comportamiento de cualquier clasificador genérico son:

- *Inicialización*: El clasificador recibe un conjunto de valores iniciales. A veces es necesario que el clasificador realice algún tipo de operación de inicialización. Aquí es donde puede realizarla.

- *Finalización*: Este método se usa cuando ya no se va a necesitar más el clasificador. A veces es necesario realizar algunas operaciones en este momento, como por ejemplo desalojar la memoria dinámica que haya reservado el clasificador.

- *Nombre*: Devuelve el nombre del clasificador.

- *Entrenamiento*: El clasificador recibe el conjunto de ejemplos de entrenamiento y realiza dicha fase.

- *Clasificación*: El clasificador recibe un caso nuevo y devuelve la clasificación

que hace de él.

- *Realimentación*: El clasificador recibe un caso, la clasificación que hizo y la clasificación correcta. Permite realimentar al sistema y conseguir que mejore con la experiencia.

- *Explicación sencilla*: Como resultado el clasificador escribe en un fichero la explicación de por qué hizo una clasificación.

- *Explicación completa*: El clasificador escribe en un fichero toda la información disponible relacionada con una clasificación: resultados del test, ejemplos que hay en los conjuntos de entrenamiento y prueba, orden en que se han presentado los casos, casos en los que ha habido problemas y una explicación de por qué se ha hecho esa clasificación.

Se crea un objeto de cada una de las clases. Así cada objeto muestra el comportamiento de un clasificador.

El programa sigue el algoritmo que se muestra en la figura 1.1 Utiliza los métodos anteriores para inicializar el clasificador al principio, indicarle al final que ya no se va a necesitar más, consultar su nombre, entrenar el clasificador, clasificar un ejemplo de prueba. Si se ha elegido la opción de realimentación entonces proporciona al clasificador información sobre si su clasificación ha sido correcta o no. Y según el nivel de detalle solicitado por el usuario para las explicaciones, el clasificador proporciona una explicación sencilla, completa, o ninguna explicación sobre por qué ha asignado una clase concreta a un ejemplo.

Cada uno de los clasificadores dispone de información sobre la Base de Casos que se utiliza en las pruebas. En concreto conoce sobre la Base de Casos número de ejemplos, número de atributos y el número y nombre de las clases. Para cada atributo conoce los valores mínimo, máximo, media estadística, desviación típica y correlación.

Las pruebas se han realizado sobre ordenadores PC compatibles con Sistema Operativo Linux. Para obtener los resultados de este trabajo se ha empleado más de un año de calculos (24 horas/dia, 7 dias/semana) y un total de 16 ordenadores,

incluídos 6 biprocesadores. Los ficheros que se han generado con resultados finales e intermedios ocupan comprimidos unos 3 GBytes de información.

Además se han realizado numerosas pruebas dedicadas a probar, ajustar, comprender y obtener todos los detalles sobre el comportamiento de los clasificadores. También se han realizado pruebas previas sobre algunas líneas que no se han llegado a desarrollar ahora y que se continuarán investigando en trabajos futuros.

1.3 Estructura del trabajo

La estructura del trabajo, organizado por capítulos, es la siguiente:

Capítulo 1 *Introducción*. Es este capítulo. Se presenta el trabajo, se muestran los objetivos que se pretenden conseguir, cómo se han realizado las pruebas, las bases de casos utilizadas y cómo está organizado el trabajo.

Capítulo 2 *El Razonamiento Basado en Casos*. Se analiza el Razonamiento Basado en Casos (RBC) en general: tipos de razonadores, fundamentos del RBC, funcionamiento, ventajas e inconvenientes de su uso, y cómo se debe comportar un Razonador Basado en Casos.

Capítulo 3 *Clasificación y Razonamiento Basado en Casos*. Este capítulo se centra en el problema de la clasificación y cómo abordarlo. Comienza con la descripción de qué es un problema de clasificación en general. A continuación se presentan algunos métodos que se han utilizado en la literatura para abordar los problemas de clasificación, como Árboles de Decisión, Reglas, ... Se muestra cómo utilizar el Razonamiento Basado en Casos para realizar labores de clasificación, y las peculiaridades que plantea aplicar RBC en este tipo concreto de problemas. Cuando utilizamos RBC el peso de la clasificación recae en gran medida en el concepto de similitud que consideremos. Las funciones de distancia constituyen la pieza básica con la que vamos a trabajar, porque toda medida de distancia nos define implicitamente similitud. Por tanto nos van a permitir definir similitud entre ejemplos, y dependiendo de la función de distancia que usemos tendremos una noción de similitud u

otra. Se aclara lo que entendemos por función de distancia a lo largo de este trabajo y se discute que éstas pueden ser menos restrictivas que las funciones de distancia utilizadas tradicionalmente en Geometría. Por último se presenta una nueva medida de distancia: la distancia basada en bandas o hiperplanos.

Capítulo 4 *Métodos de Clasificación Basados en Distancias*. Se analizan los métodos de clasificación típicos que utilizan las medidas de distancia como pieza básica para realizar la labores de clasificación. Se describen varios métodos para asignar una clase a un ejemplo nuevo: vecino más cercano y k vecinos más cercanos. Todos los métodos están basados en medidas de distancia que determinan el grado en que dos ejemplos son similares. Como aportación se presentan variantes del método k–NN que difiere del método básico en hasta tres características: los ε–entornos, usar k–NN cuando el ε–entorno está vacío, y usar una heurística para seleccionar la medida de distancia que debe emplearse en una base de casos concreta. Los métodos de clasificación que se aparecen en este capítulo pueden utilizarse con cualquiera de las funciones de distancia del capítulo anterior. Así podemos realizar múltiples combinaciones y obtendremos distintos clasificadores.

Capítulo 5 *Métodos de Clasificación Basados en la Distancia de las Bandas* En este capítulo se profundiza en la medida de distancia basada en bandas que se presentó en el capítulo 3, y se desarrolla hasta que resulta operativa para utilizarla en problemas de clasificación. Existen muchos problemas donde la relaciones de similitud propias del dominio no se pueden recoger adecuadamente con las medidas de distancia típicas de la Geometría. Además se propone un cambio del enfoque habitual, en lugar de intentar buscar los ejemplos más similares a un nuevo caso que se debe clasificar, se propone que los casos existentes aprendan información de la zona donde se encuentran y cuando se deba clasificar un caso nuevo, sean ellos los que indiquen cómo de similar o lejano "ven" al caso nuevo. Se analiza las implicaciones y ventajas que aporta este cambio de enfoque. Se proponen algoritmos para aprender las bandas o hiperplanos que pasan por cada punto de aprendizaje y se ajustan mejor a la información del entorno y se realiza una batería de pruebas para analizar el comportamiento de los clasificadores que emplean este nuevo tipo de medida

de distancia.

Capítulo 6 *Resumen, Conclusiones y Trabajos Futuros* Se realiza un resumen de los aspectos más relevantes del trabajo, se presentan las conclusiones finales de este trabajo y se analizan las líneas de trabajo que parecen más prometedoras para continuar el trabajo que se ha iniciado con esta Tesis Doctoral.

Capítulo 2

El Razonamiento Basado en Casos

Sus orígenes se remontan a algunas décadas atrás como resultado del estudio de la forma de razonar en el hombre. Y en particular sobre cómo aprovechamos nuestra experiencia en la vida cotidiana y los procesos que utilizamos para razonar basándonos en experiencias pasadas. Por tanto el Razonamiento Basado en Casos está relacionado con la filosofía, la psicología y la Inteligencia Artificial. Y según el área de conocimiento a la que demos más peso podremos considerarlo como un Modelo Cognitivo o como una metodología de desarrollo de Sistemas Expertos.

EL RBC toma como modelo la forma de razonar humana, y se basa sobre todo en el uso de la experiencia previa para afrontar problemas y situaciones nuevas. Está despertando bastante interés debido a que resulta muy intuitivo, permite aprendizaje, y se han construido algunos sistemas con bastante éxito en dominios complejos donde otras técnicas no resultan apropiadas.

Aquí nos vamos a centrar en la disciplina de la Inteligencia Artificial, donde el RBC es una metodología de desarrollo de Sistemas Expertos y una técnica de resolución de problemas relativamente nueva que se engloba dentro del campo del Razonamiento Aproximado.

Vamos a hablar de Razonamiento Basado en Casos en general, pero por supuesto

hay muchas aplicaciones de muy distinto tipo. Cada sistema se ha construido de forma diferente y con propósitos diferentes, pero todos comparten características comunes. Vamos a mostrar en qué se fundamenta esta metodología de desarrollo de sistemas expertos y las características que suelen tener la mayoría de los sistemas basados en casos.

En este capítulo se va a hacer una exposición sobre los fundamentos y el funcionamiento en general del RBC. Para profundizar sobre el tema se recomienda consultar la bibliografía que aparece al final, especialmente [Kol93], [Kol92].

2.1 El Razonamiento Basado en Casos en la vida cotidiana

El Razonamiento Basado en Casos se fundamenta en el uso de la experiencia para resolver problemas. Cuando el sistema se enfrenta a un problema, recuerda soluciones que funcionaron bien con problemas similares y las utiliza como punto de partida en la resolución.

De hecho, esto es lo mismo que hacemos nosotros en multitud de ocasiones. Lo utilizamos por ejemplo cuando resolvemos problemas de estadística que siguen un mismo patrón; cuando un abogado tiene en cuenta la jurisprudencia previa; o cuando a un médico se le presenta por segunda vez un paciente con unos síntomas poco frecuentes: utiliza directamente como hipótesis de trabajo el mismo diagnóstico al que llegó antes, y ahorra así mucho tiempo y esfuerzo.

En la vida cotidiana utilizamos también el RBC, por ejemplo, cuando cocinamos spaguetti somos capaces de hacerlo basándonos en cómo preparamos los macarrones. Cuando vamos a comer a un restaurante, a menudo pedimos los platos de acuerdo a nuestra experiencia previa sobre si estaban buenos o no en ese restaurante o en otros parecidos. Cuando llegamos a una ciudad nueva intentamos comprar el Bono-bús o el billete de metro de forma parecida a como lo hemos hecho antes en otras ciudades.

Además, en cualquier campo nos vamos a encontrar con que *la segunda vez que*

resolvemos un problema es más fácil que la primera vez que lo hacemos, porque recordamos lo que hemos hecho y repetimos la solución que ya hemos utilizado antes con buenos resultados.

Más formalmente podríamos decir que en el RBC, el razonador resuelve problemas nuevos adaptando soluciones que ya fueron utilizadas con éxito en problemas anteriores parecidos.

Para hacer esto, debemos comparar el problema al que nos enfrentamos actualmente con aquellos que resolvimos satisfactoriamente en el pasado. Y una vez que hemos recordado problemas parecidos al actual podemos usar la misma solución que utilizamos en el pasado, aunque a veces es necesario realizar algún tipo de adaptación para que funcione en la situación actual.

Pero no nos vamos a quedar ahí, porque también podemos aprender de nuestros errores e intentar evitarlos en situaciones similares que se nos presenten en el futuro.

En el Razonamiento Basado en Casos se puede utilizar la experiencia previa para:

- Proponer soluciones a problemas nuevos adaptando soluciones que funcionaron bien con otros problemas parecidos.

- Evitar y anticipar los errores cometidos en el pasado, es decir, utilizar casos anteriores para criticar soluciones propuestas y si es necesario corregirlas.

- Explicar situaciones nuevas a partir de nuestro conocimiento previo sobre situaciones similares.

- Interpretar una situación nueva razonando a partir de precedentes.

También es importante conocer cómo realizamos analogías y usamos el razonamiento por analogía porque permite pasar resultados y conclusiones de un dominio a otro diferente, o comprender una situación basándonos en otra.

2.2 Tipos de Razonadores Basados en Casos

Casi todos los sistemas de RBC son muy distintos entre sí. Atendiendo a la finalidad del sistema se pueden dividir en dos grandes bloques:

- Resolución de problemas: se recuerdan soluciones antiguas y se adaptan para resolver un problema nuevo. Aquí se incluye una gran variedad de tareas de resolución como:
  - planificación: dar una secuencia de pasos o acciones para conseguir un objetivo: que en el mundo se alcance algún estado concreto o se cumpla alguna condición.
  - diagnosis: se proporciona un conjunto de síntomas y se pide al razonador que encuentre una explicación para esos síntomas.
  - diseño: se define un conjunto de restricciones y el razonador debe diseñar un objeto o artefacto que cumpla esas restricciones.
- Interpretación: se evalúan situaciones o soluciones tomando como punto de partida la experiencia previa. Se debe recordar situaciones anteriores y se comparan con la nueva para realizar una interpretación o clasificación de la situación nueva. Aquí se pueden incluir problemas como:
  - justificación: dado un argumento o solución se demuestra que son correctos y por qué.
  - interpretación y clasificación: se intenta colocar una situación nueva dentro de un contexto y ver dentro de qué categoría clasificamos esa situación.
  - proyección o previsión: se predicen los efectos de una solución, decisión o plan antes de llevarlos a la práctica. Dentro de este apartado entraría la toma de decisiones.

Hay que tener en cuenta que esta clasificación de sistemas puede resultar un poco artificiosa, ya que la mayoría de los sistemas realizan tareas de varios de los

tipos anteriores. Es muy normal que un sistema que realiza resolución, en un momento dado utilice interpretación para predecir la utilidad, calidad o resultados de una solución. Así por ejemplo, un planificador puede realizar tareas de previsión para anticipar qué va a pasar en el mundo cuando intente ejecutar el plan que acaba de diseñar.

También se puede tener en cuenta la interacción del sistema con el usuario. Hay algunos que funcionan de forma totalmente automática y autónoma. Otros permiten cierto grado de interacción con el usuario. Y algunos están concebidos para que sean interactivos y funcionan como sistemas de ayuda o sirven para "aumentar la memoria".

Algunos ejemplos de Razonadores son los siguientes:

- JULIA: realiza diseño dentro del dominio de la confección de menús de comidas. Usa adaptación para transformar y modificar menús previos y hacer que cumplan nuevas restricciones.
- HYPO: genera argumentos legales citando casos pasados a favor y en contra como justificación de sus argumentos.
- PERSUADER: genera soluciones para negociaciones laborales que sean aceptables para ambas partes
- CLAVIER: indica cómo deben colocarse las piezas dentro de una autoclave. Usa casos para tener distribuciones de piezas, sugerir sustituciones y criticar soluciones. Proporciona soluciones en un dominio que carece de un modelo y resulta complejo incluso para los expertos. Mejora los resultados de los expertos.
- CASEY: utiliza RBC para acelerar el razonamiento basado en modelos. Forma una explicación de los síntomas de un enfermo de corazón. Está construido sobre un razonador que utiliza un modelo causal para acelerar el razonamiento basado en modelos.
- Battle Planner: sistema de aprendizaje interactivo. El usuario describe la situación de una batalla y elige un plan. El sistema predice el resultado.

- ARCHIE-2: sistema de ayuda de diseño en arquitectura. El usuario pide información sobre algún aspecto y el sistema le ofrece información relevante, ejemplos y sugerencias.

2.3 El Caso y la Librería de Casos

Un *caso* es una porción de conocimiento que representa una experiencia concreta y el contexto en que sucedió. Es importante almacenar el contexto en que se produjo la experiencia porque nos sirve para determinar cuando es aplicable ese conocimiento. Por ejemplo, el tratamiento que el doctor recomendó a un paciente de edad avanzada que presentaba unos determinados síntomas es aplicable a ese paciente en esa situación, pero probablemente no será adecuado para un recién nacido aunque presente síntomas parecidos. También es importante si esos síntomas se presentan de forma aislada o como parte de una epidemia.

Los casos pueden ser de tamaños diferentes, y se van a almacenar en una memoria que se conoce con el nombre de *librería de casos*.

No nos va a interesar cualquier experiencia que se produzca dentro del dominio de la aplicación, sino sólo aquellas que vayan a ser relevantes para nuestros fines. Y de entre éstas, vamos almacenar sólo aquellas experiencias memorables, es decir, aquellas que sean diferentes de lo normal, de lo que se esperaba, o simplemente que no tengamos almacenada ninguna parecida en nuestra librería de casos.

A veces la lección que enseña el caso podemos deducirla de otros casos de la librería, pero si para ello se necesita un gran coste entonces puede interesar almacenarlo para ahorrar esfuerzo. Hay que intentar mantener un equilibrio, y en general si la lección que enseña el caso es fácilmente deducible a partir de otros casos de la librería entonces no debe guardarse porque gastamos espacio en experiencias que no aportan nada nuevo.

En cuanto a la forma de representar un caso, hay muchos formalismos dentro de la Inteligencia Artificial que podemos utilizar: marcos, redes semánticas, reglas, valores de atributos,... La elección del formalismo de representación depende mucho del sistema, finalidad, gustos del autor...

Un caso representa una experiencia asociada a un determinado contexto, pero independientemente del formalismo que se utilice, debe almacenar al menos:

- Descripción del problema y/o situación en que se produjo (contexto):
  objetivos que deben alcanzarse, restricciones o condiciones sobre esos objetivos, características de la situación del problema, información que se ha utilizado para alcanzar los objetivos,...
- Solución al problema o reacción que se ha tomado ante la situación:
  la solución, pasos de razonamiento seguidos, justificaciones (¿por qué se hizo?), soluciones alternativas, resultados esperados, ...
- Resultados obtenidos:
  el resultado, si ha sido éxito o fracaso, si ha cumplido o no las expectativas, estrategia de reparación, referencia a la siguiente solución del problema,...

Si va a haber interacción con el usuario, los casos deberían incluir también texto e información de tipo gráfico para que el usuario se sienta cómodo con el sistema, y facilitar así la interacción.

2.4 El Ciclo del Razonamiento Basado en Casos

Como ya hemos visto antes, en el RBC debemos comparar el problema al que nos enfrentamos actualmente con aquellos que resolvimos satisfactoriamente en el pasado. Y una vez que hemos recordado el problema más parecido al actual, a veces podemos usar la misma solución que utilizamos en el pasado, y otras veces es necesario realizar algún tipo de adaptación para que la solución funcione ahora.

A pesar de la gran variedad de sistemas que existen, todos siguen unas pautas parecidas. Y el proceso de razonamiento que en general siguen todos ellos se puede descomponer en una serie de pasos, como se muestra gráficamente en la fig. 2.1.

Cada vez que el sistema se plantea un problema nuevo sigue estos pasos, por eso se les conoce con el nombre de ciclo del RBC. Algunos razonadores incluirán

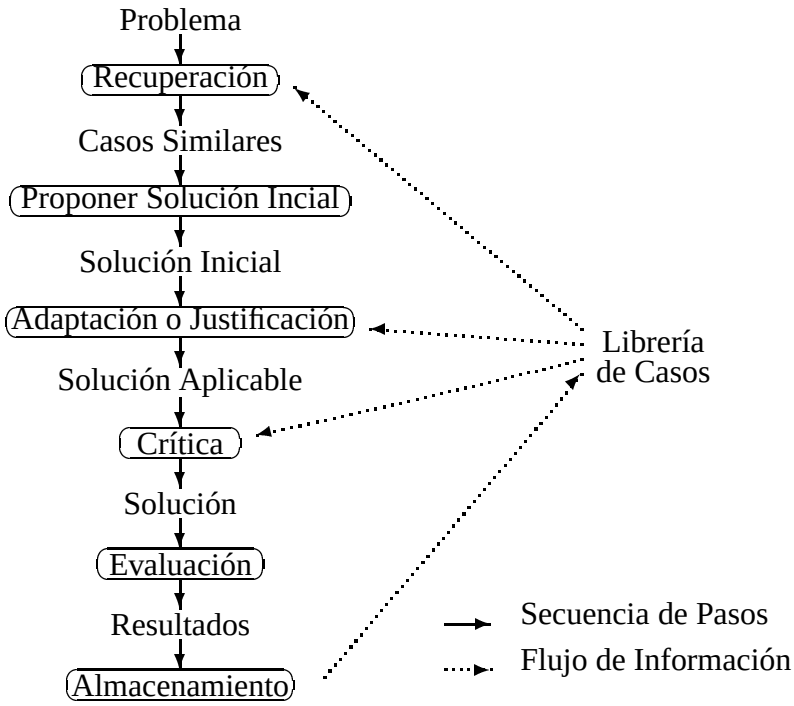


Figura 2.1: El ciclo del RBC

todos los pasos. Otros en cambio tendrán sólo algunos. En los apartados siguientes se describe qué se hace en cada uno de los pasos que forman el ciclo del RBC.

2.4.1 Recuperación

En este paso se recuerdan las experiencias pasadas que pueden ser útiles. Se consulta la librería de casos y se recupera uno o varios casos similares al problema actual.

En general se suele hacer en dos etapas:

- *Recordar casos previos*: En esta etapa se hace una primera selección de los casos que van a ser útiles para resolver el problema. Como veremos más adelante, se utiliza la indexación para dirigir la búsqueda y realizarla de forma

eficiente. Por tanto, lo primero que hay que hacer es una evaluación de la situación, es decir, considerar las características del problema nuevo y calcular los índices que debemos utilizar para recuperar los casos útiles. Después buscamos en la librería de casos y recuperamos aquellos que tengan una o varias de las características expresadas en los índices.

- *Seleccionar el mejor subconjunto*: Entre los casos recuperados en el paso anterior se hace una segunda selección y nos quedamos con el o los casos más prometedores. Así el razonador puede concentrarse en esos casos y analizarlos en profundidad. Para determinar los casos que son más prometedores se suelen utilizar dos técnicas: *Matching* y *Ranking*.

2.4.1.1 Matching y Ranking

En el Matching se compara dos casos y se determina el grado de similaridad entre los casos, o simplemente se dice si son suficientemente similares o no. Por ejemplo, se puede utilizar una función numérica, como en el método del vecino más cercano que veremos más adelante.

En el Ranking se realiza una ordenación relativa de los casos de acuerdo a su utilidad o su parecido con un caso dado. A menudo se ordena de acuerdo a los resultados del matching, pero no tiene por qué ser así. Se puede hacer una ordenación de todos los casos de la librería comparando si un caso es más útil que otro o no.

El matching y ranking pueden ser estáticos o dinámicos según vayan evolucionando en el tiempo o no, y globales o locales según sean siempre iguales para todos los casos o sean sensibles al contexto.

Una de las formas más simples consiste en calcular el grado de similaridad mediante una función numérica que calcule distancias entre casos. Este método se conoce con el nombre de búsqueda del *vecino más cercano*. Si hablamos en términos de similaridad, se determina el parecido entre los dos casos en cada característica y se realiza una media ponderada según la fórmula

$$\frac{\sum_{i=1}^{n} w_i \times sim(P_i, R_i)}{\sum_{i=1}^{n} w_i}$$

donde w_i es la importancia de cada característica, *sim* es la función de similaridad, P_i y R_i son los valores de la característica i en el problema y en el caso recuperado respectivamente, y n es el número de características que estamos considerando.

El problema es más complejo de lo que parece a primera vista. Hay casos que no comparten características superficiales, pero sin embargo sí que tienen en común características derivadas de las anteriores o que son más abstractas. Por ejemplo, para predecir quién va a ganar una batalla es importante la proporción de fuerzas atacantes y defensoras, pero no el número absoluto de fuerzas atacantes o defensoras por separado. Las estrategias en el ajedrez y el fútbol tienen bastante en común a pesar de que ambos deportes son muy diferentes. En ambos juegos hay dos partes enfrentadas que quieren ganar y que pierda la otra parte; hay ataque, defensa y contraataque; son importantes las posiciones dentro del "terreno de juego",...

2.4.1.2 Indexación

Cuando la librería de casos es grande, nos vamos a encontrar con que la búsqueda de casos parecidos va a ser lenta si se hace de forma secuencial, recorriendo toda la librería de casos y comparándolos uno por uno hasta encontrar el más parecido.

Básicamente el problema de la indexación consiste en cómo recuperar en un tiempo razonable los casos que van a ser útiles. Para conseguirlo se utiliza una estrategia similar a la que se emplea en los libros, donde se añade al final un índice de materias que permite acceder de forma rápida a las hojas donde se trata una determinada materia.

En la indexación a cada caso se asocia uno o varios índices que van a indicar cuando ese caso va a ser útil. Y diremos que el caso esta indexado por esos índices.

Aquí se nos van a plantear dos problemas. El primero es la elección de un *vocabulario de índices*. Consiste en determinar qué tipo de etiquetas va a ser relevante para cada tipo de caso, o sea, el tipo de índices que debemos utilizar para indexar los casos. Haciendo un símil con las Bases de Datos debemos determinar qué campos deben tener y los valores que pueden tomar. El segundo problema es la *selección de índices* y consiste en elegir las etiquetas o índices que deben asignarse a cada caso

particular.

Como los índices se van a utilizar para recuperar casos, deben ser predictivos y anticipar cuando el caso va a ser útil en el futuro. Por eso se suele analizar qué lecciones enseña el caso, cuando van a ser útiles esas lecciones, si esas lecciones se pueden generalizar para aplicarlas en más situaciones, qué características diferencian a este caso de los demás,...

Cuando nos enfrentamos a un problema y necesitamos recuperar casos similares, lo primero que vamos hacer es analizar el problema y determinar qué características deben tener los casos que nos pueden ser útiles. Así determinaremos los índices que debemos utilizar para buscar en la librería de casos.

Estos índices van a ser parte de la información que almacena el caso, es decir, van a ser algunos de sus campos. Aunque en general los casos no tienen por qué tener todos los mismos campos, y mucho menos los mismos índices.

Por ejemplo, podemos utilizar como índice el contexto en que se produjo el caso, porque en principio va a indicar cuando el caso se va a poder aplicar en el futuro y va a ser útil.

Por lo tanto vamos a tener que la memoria del sistema va a estar compuesta por la librería de casos propiamente dicha que almacena los casos, y unos procedimientos de acceso que van a permitir aumentar la eficiencia de la búsqueda.

Otra forma de afrontar este problema es realizar una búsqueda en paralelo. Si disponemos de hardware que permita paralelismo SIMD (*Single Instruction Multiple Data*: la misma instrucción se aplica a varios conjuntos de datos a la vez) o MIMD (*Multiple Instruction Multiple Data*: varios procesadores ejecutan instrucciones diferentes en paralelo), podemos aprovechar la potencia que nos ofrecen estos sistemas y repartir la búsqueda entre las diferentes Unidades de Ejecución[1] para disminuir considerablemente el tiempo empleado en las labores de búsqueda. Si disponemos del suficiente número de Unidades de Ejecución podemos llegar incluso a consultar toda la librería de casos a la vez y emplear O(1), tiempo constante, en la búsqueda.

[1] Si hablamos de MIMD se entiende "Unidad de Ejecución" como Procesador y en SIMD como Unidad Aritmetico-Lógica

En ocasiones la Librería de Casos contiene miles de casos y ésta es la única posibilidad de conseguir un sistema de RBC capaz de proporcionar respuestas en un tiempo razonable.

2.4.2 Propuesta de una Solución Inicial

En este paso se consideran los casos recuperados en la etapa anterior para elaborar una solución. Si se intenta resolver un problema, normalmente se toma como punto de partida la solución del problema recuperado, parte de ella o una combinación de las soluciones recuperadas.

A esta solución inicial se le suelen hacer adaptaciones sencillas, que podríamos denominar de sentido común, antes de pasar a la fase de adaptación propiamente dicha. Por ejemplo se puede tener en cuenta la inflación para actualizar los salarios.

Si se trata de un problema de interpretación se suele agrupar los casos recuperados de acuerdo con la interpretación que hacen. Y a partir de estos grupos se asigna una interpretación inicial de la situación. También puede ocurrir que la interpretación inicial nos venga ya dada y no sea necesario este paso. Ocurre por ejemplo con un abogado que debe defender a su cliente: la interpretación inicial es la favorable a su cliente, y después debe encontrar argumentos que apoyen esta interpretación.

2.4.3 Adaptación

Normalmente la solución inicial propuesta en el paso anterior no va a servir directamente en el problema nuevo. Por eso será necesario adaptar de algún modo esa solución y hacer que cumpla los requerimientos de la nueva situación. Primero se debe determinar qué necesita ser adaptado y después se realiza esa adaptación.

Para identificar qué debe ser corregido se pueden usar varios métodos:

- Diferencias entre las especificaciones del problema recuperado y el nuevo.
- Usar una lista de comprobaciones que se deben realizar.

- Detectar inconsistencias entre la solución recuperada y los objetivos y restricciones del problema nuevo.
- Prever los efectos que va a tener la solución, por ejemplo utilizando modelos, casos o simulación.
- Llevar a cabo la solución y analizar el resultado.

Se ha estudiado cómo realizar la adaptación y se ha encontrado que hay algunos métodos generales que son válidos independientemente del dominio de aplicación, y otros que son dependientes del dominio. Básicamente se puede hablar de [Kol93]:

- Métodos de sustitución: se sustituye algunos valores o elementos de la solución antigua por otros adecuados al problema nuevo. Estos métodos van desde la reinstanciación o el ajuste de parámetros hasta la sustitución basada en casos. Es lo que se hace al utilizar la receta de los macarrones para cocinar spaghetti.
- Métodos de transformación: se realizan modificaciones más importantes en la solución antigua. Ahora se utiliza reparación guiada por modelos o heurísticas de sentido común que añaden, eliminan o reemplazan algunos componentes de la solución.
- Adaptación y reparación de propósito especial: son métodos específicos del dominio que realizan sustituciones, transformaciones estructurales o reparaciones que intentar corregir los fallos que se producen al ejecutar la solución.
- Repetición de la derivación: repetir los pasos que se han usado al derivar la solución antigua para obtener la solución al problema nuevo. Es lo que se hace al resolver un problema de estadística siguiendo los mismos pasos que usamos antes en otro problema igual pero con valores distintos.

También se puede observar que el tipo de RBC influye en el uso que se hace de la adaptación [HKSC95]:

- Las tareas de identificación, reconocimiento y clasificación se pueden realizar sin adaptación.

- Las tareas de predicción necesitan adaptación sólo en cierto grado.
- Las tareas de diseño y planificación necesitan en general gran cantidad de adaptación de todo tipo.
- Los sistemas que utilizan soluciones con partes que interactúan necesitan usar también gran cantidad de adaptación.

2.4.4 Justificación y Crítica

Si el Razonamiento Basado en Casos se utiliza para interpretar situaciones, entonces se realiza una fase de justificación en lugar de la fase de adaptación.

En estas dos fases se justifica o critica una solución o interpretación antes de probarla en el mundo. Se han considerado juntas porque la función de ambas es muy parecida: buscar argumentos a favor y en contra para valorar la solución que se ha obtenido. Esto se puede hacer básicamente de dos formas:

- Comparando y contrastando la solución propuesta con otras parecidas que se encuentran en la librería de casos.
- Proponiendo situaciones hipotéticas para comprobar la robustez de la solución.

La fase de crítica puede llegar a la conclusión de que la solución no funciona y que, por tanto, deberá ser modificada. La adaptación que se realiza en este momento se conoce con el nombre de *reparación*.

2.4.5 Evaluación

El resultado del razonamiento se prueba en el mundo real. El sistema analiza los resultados obtenidos y si son diferentes de los esperados intenta dar una explicación.

En este paso el sistema interacciona con el entorno y tiene una forma de evaluar sus decisiones en el mundo real. Esto hace posible que el sistema juzgue la bondad

de las soluciones que propone y así pueda en el futuro anticipar y evitar los errores que ya ha cometido antes.

También se puede obtener información sobre los resultados que va a tener la solución comprobándola con un modelo, mediante simulación o simplemente como resultado de la interacción con el usuario.

El proceso de evaluación incluye explicación de diferencias entre lo esperado y lo que ha ocurrido, justificación de diferencias, previsión de salidas... Además como resultado de la evaluación puede ser necesaria más adaptación o reparación de la solución propuesta.

2.4.6 Almacenamiento

El caso actual se debe almacenar en la librería de casos para poder usarlo en el futuro. En el caso se debe incluir la descripción del problema, solución propuesta, resultado obtenido y cualquier información que pueda ser útil en el futuro.

Esta fase del almacenamiento consta de dos pasos:

- *Elegir cómo se va a indexar el caso*. Los índices se deben seleccionar de forma que en el futuro el caso sea recuperado cuando resulte útil. Esto implica que los índices deben ser predictivos y el razonador debe anticipar la importancia que va a tener el caso en el futuro.

- *Insertar el caso en la memoria*, y si es necesario reorganizarla.

2.5 El Razonamiento Basado en Casos y el Aprendizaje

Uno de los aspectos más interesantes del RBC es que el sistema es capaz de aprender y mejorar su comportamiento conforme aumenta su experiencia.

Se puede decir que con el RBC el sistema aprende de dos formas.

- El razonador es más eficiente según aumenta su experiencia porque *recuerda soluciones antiguas y las adapta en lugar de tener que derivarlas cada vez a partir de cero*. Por este motivo se puede utilizar el RBC para mejorar la eficiencia de otros métodos de razonamiento. Se puede construir un sistema híbrido que utilice Razonamiento Basado en Reglas (RBR) y RBC. De forma que cuando se tiene una experiencia parecida el RBC proporciona una respuesta buena en poco tiempo, y si no, el sistema basado en reglas deriva la respuesta. Así, la próxima vez que se plantee un problema similar, ya estará disponible una experiencia relativamente buena que el RBC podrá usar para construir de forma rápida una solución.

- La otra forma en que podemos considerar que el RBC aprende es porque *aprovecha la experiencia para obtener soluciones mejores*. Cuando el RBC comete un error lo incorpora a la librería de casos y puede tenerlo en cuenta en el futuro para no repetirlo. Además según aumenta su experiencia se amplía la librería de casos y podrá recuperar casos que sean más parecidos al problema actual, y por tanto obtener mejores soluciones.

Los sistemas expertos basados en reglas son diseñados por el autor y se ponen a funcionar sin que el sistema pueda evolucionar. Si se instalan varias copias en sitios diferentes, después de un tiempo todas las copias serán iguales y se comportarán de igual modo. Añadir más conocimiento al sistema en forma de reglas es una labor lenta y engorrosa. Hay que extraer las reglas que se quieren añadir y después comprobar su completitud.

Sin embargo en el RBC el sistema aprende progresivamente según va funcionando, es decir *evoluciona*. Es por esto que *la adquisición de conocimiento nuevo se hace de forma automática y no supone ningún esfuerzo adicional*. Así, pasado un tiempo, diferentes copias de la misma aplicación habrán tenido experiencias diferentes, habrán evolucionado de manera diferente y se comportarán de forma diferente. Esto no es malo. De hecho es lo que nos sucede a los hombres: cada uno aprende de acuerdo a sus propias experiencias. Además es bueno porque cada copia se habrá especializado en resolver los problemas que se plantean en su entorno. Además esto permite utilizar el sistema como una memoria corporativa que

acumula la experiencia que tienen los empleados de la organización y ponerla al servicio de todos.

Tradicionalmente dentro de la comunidad de la Inteligencia Artificial se identifica aprendizaje con la capacidad de construir generalizaciones que sean ampliamente aplicables. Sin embargo dentro del RBC la fuente principal de aprendizaje es la acumulación de casos nuevos. Las generalizaciones suponen sólo una parte y se usan principalmente para disminuir el tamaño de la librería de casos y mejorar la eficiencia, o cuando se hibrida con otros métodos como el RBR.

Tener conocimiento concreto tiene dos ventajas.

- *Tener conocimiento sobre una experiencia implica que ya tenemos una solución completa que podemos usar directamente o con alguna adaptación*. Sin embargo si tenemos conocimiento general hay que empezar a derivar la solución desde cero. Esto es lento y complejo, y si hay interacción entre las partes del problema, se complica aún más.

- *El conocimiento sobre casos concretos permite tratar las excepciones de forma adecuada*. El conocimiento general no es capaz de reflejarlas o necesita usar mecanismos complejos, y en el RBC es tan simple como incorporar casos que sean ejemplos de esas excepciones.

Por supuesto, esto no quiere decir que el conocimiento general no sea bueno. Sin embargo la experiencia en Inteligencia Artificial ha demostrado que los métodos generales son ampliamente aplicables y compactan mucho conocimiento, pero el uso de conocimiento específico del dominio suele producir resultados mejores con menos esfuerzo. Como contrapartida almacenar conocimiento específico tiene unos requerimientos de memoria de almacenamiento mucho mayores.

Se puede construir, por ejemplo, un sistema híbrido con RBC y RBR. De forma que en las situaciones generales se aplica RBR, y en las excepciones o donde no se disponga de conocimiento general, se aplica RBC.

Esto precisamente es lo que hemos hecho en [CCJL03], donde se propone en primer lugar aprender un conjunto de reglas difusas maximales mediante una Rejilla de Repertorio Difusa con el doble objetivo de obtener un sistema preciso y

fácilmente interpretable. A continuación se refina el conjunto de reglas, permitiendo añadir excepciones a aquellas reglas que entren en conflicto en alguna región con otras reglas (excepciones generales). Si en alguna zona sigue existiendo un conflicto especialmente delicado de resolver, se propone usar excepciones puntuales, es decir, no cubrir esa región del espacio por ninguna regla y utilizar entonces RBC con el método de los ε–entornos y la heurística propuestos en los apartados 4.3.1 y 4.3.3. Además excepciones puntuales permiten conservan la interpretabilidad de las respuestas, ya que cualquier experto entenderá que se justifique una salida porque determinados casos son muy parecidos a la entrada. De esta forma se integra un sistema de clasificación Basado en Reglas Difusas con un sistema Basado en Casos, permitiendo obtener los beneficios de ambos enfoques, manteniendo la interpretabilidad del sistema y obteniendo una gran precisión en las fronteras de decisión en las zonas conflictivas. En [CCJL03] hemos aplicado este sistema a un problema de tasación de propiedades.

2.6 Ventajas e Inconvenientes del RBC

El Razonamiento Basado en Casos proporciona múltiples ventajas:

- Puede proponer soluciones de forma rápida, sin tener que derivarlas desde cero.
- Puede proponer soluciones en dominios complejos donde no hay un modelo claro o en dominios que el razonador no comprende totalmente.
- Se puede evaluar y obtener soluciones aunque no exista ningún método algorítmico para ello.
- Puede utilizar la experiencia previa para prevenir problemas potenciales y no repetir errores cometidos en el pasado.
- Los casos ayudan al razonador a concentrarse en las características más importantes del problema.

- Los casos son útiles para interpretar conceptos que no están claramente definidos.
- No necesita una elicitación de conocimiento como en los sistemas basados en reglas, porque no requiere un modelo explícito del dominio del problema. Además, para los expertos es más cómodo y sencillo contar situaciones concretas (casos) que intentar dar un conjunto de reglas que explique el comportamiento del dominio.
- El sistema es fácil de mantener una vez desarrollado. Aprende incorporando nuevos casos y crece de forma progresiva, sin necesidad de actualizaciones. Además permite que se especialice en el dominio en que trabaja.
- Se adapta a la organización y ambiente en que trabaja, mediante el aprendizaje de casos nuevos.
- Puede utilizarse como almacén de conocimiento de una organización, lo que permite difundir experiencias y conocimiento experto por toda la organización y entrenar a personal nuevo.
- Utiliza la experiencia para resolver problemas (como hacen los expertos), para aprender, prevenir soluciones malas...
- Razona de forma parecida a los hombres, por lo que resulta más fácil comprender las soluciones y justificaciones que proporciona.
- Es capaz de responder a situaciones excepcionales y poco comunes.
- Parte de soluciones globales, por lo que no es necesario descomponer el problema en subproblemas, resolver los distintos subproblemas y después unir las soluciones parciales evitando las interacciones que puedan surgir.
- Permite razonar con conocimiento incierto e impreciso.
- Permite razonamiento por analogía.
- Permite tratar con valores ausentes y utilizar razonamiento por defecto con facilidad

Por supuesto el Razonamiento Basado en Casos también tiene desventajas y críticas:

- El razonador puede tener tendencia a utilizar los casos recuperados a ciegas, sin comprobar si esas soluciones son válidas en la situación nueva.
- Si los casos almacenados no son representativos y tienen un sesgo, pueden influir demasiado en el razonamiento y obtendremos soluciones que no son buenas.
- A menudo no se recuerdan los conjuntos de casos más apropiados para razonar, sobre todo cuando la gente o el sistema es inexperto.
- No tiene un fundamento teórico sólido en que apoyarse ni una semántica bien definida, pero la experiencia demuestra que funciona.

De todas formas hay que tener claro que el RBC tampoco es la panacea universal que permite resolver cualquier problema. En dominios donde exista un modelo bien definido, los sistemas basados en reglas, probablemente funcionarán muy bien. Y además están ya muy estudiados y son una tecnología madura que ofrece buenos resultados.

La calidad del RBC va a depender de cinco factores:

1. La experiencia que tiene.
2. Su habilidad para comprender situaciones nuevas a partir de otras previas.
3. Su capacidad para realizar adaptación.
4. Su capacidad para realizar evaluación y reparación.
5. Su habilidad para integrar experiencias nuevas en la memoria.

El Razonamiento Basado en Casos es una metodología de razonamiento relativamente nueva con muchas expectativas de futuro, pero que está madurando rápidamente. Ya hay disponibles herramientas de desarrollo comerciales que facilitan

gran parte del trabajo de desarrollo y permiten construir aplicaciones que usan RBC de forma rápida y fiable [Wat96].

Todavía hay mucho por hacer. Se deben desarrollar herramientas que permitan planificación, diseño, evaluación y adaptaciones complejas y sean a la vez robustas; mejorar las metodologías de recolección de casos; herramientas de mantenimiento de la librería de casos; manejar grandes librerías de casos; selección de índices de forma automática; crear vocabularios de propósito general; caracterizar nuevos tipos de adaptación; extraer casos de entornos continuos. También se puede construir sistemas basados en casos que sean altamente interactivos, y otros que sirvan como memoria corporativa.

Otro campo muy amplio se abre cuando se intenta combinar el RBC con otras técnicas de razonamiento para combinar lo mejor de ambos mundos: razonamiento basado en reglas, basado en modelos o redes neuronales, por ejemplo. Tampoco hay que olvidar por supuesto su relación con la psicología y la influencia que los avances en este campo pueden tener en el RBC. También los avances en el campo de la Inteligencia Artificial permiten comprender mejor los modelos cognitivos. Además como los sistemas basados en casos razonan de forma parecida a los hombres, pueden ser una herramienta muy útil de entrenamiento y aprendizaje.

En cualquier caso parece claro que todavía hay mucho trabajo por hacer y que al Razonamiento Basado en Casos le espera un futuro bastante halagüeño en los próximos años.

Capítulo 3

Clasificación y Razonamiento Basado en Casos

3.1 El problema de la Clasificación

En un problema de clasificación, se parte de un conjunto finito de ejemplos, y para cada ejemplo se tiene un conjunto de observaciones de algunas características relevantes y la clasificación correcta.

El objetivo es relacionar las observaciones y las clases, y así poder determinar la clase a la que pertenece cualquier objeto dado a partir de los valores de sus atributos. De esta forma, cuando se presenta un caso nuevo al sistema, éste tiene como información el conjunto de valores que presentan los atributos de ese caso. Y a partir de ese conjunto de observaciones, es capaz de determinar correctamente la clase a la que pertenece el caso.

Como ejemplo de problema de clasificación podemos considerar cualquiera en el que deba asignarse a un objeto una categoría, una clase o una valoración. El campo es muy amplio, pero por citar algunos ejemplos, incluye diagnóstico médico, clasificación de animales, detección de errores en dispositivos electricos y/o

mecánicos, ayuda a la toma de decisión (concesión de créditos, evaluación de riesgos,...), o incluso dada la posición de las piezas en una partida de ajedrez determinar si teoricamente ganan blancas, negras o son tablas.

Formalmente podríamos decir que existe una serie de objetos o datos $x \in U$, y que disponemos sólo de un subconjunto de esos datos o ejemplos $BC \subset U$ y un conjunto de d clases o clasificaciones posibles $C = \{C_1, C_2, \cdots, C_d\}$. Para cada ejemplo se tiene n valores para los atributos $A_1, A_2, \ldots, A_n$, que en principio pueden ser numéricos continuos, numéricos discretos o simbólicos; y una clasificación $c \in C$. En este trabajo vamos a considerar sólo valores numéricos. Esto en realidad no supone ninguna pérdida de generalidad porque siempre se puede establecer una biyección entre el conjunto de valores simbólicos y un conjunto de valores numéricos, y trabajar con los números en lugar de con los símbolos. Así vamos a poder definir nuestros objetos $x \in U$ como un vector de valores reales $x = (x_1, \ldots, x_n)$, con $x_i \in D_i$, donde D_i es el i-ésimo dominio, $x \in D$, y $D = D_1 \times \cdots \times D_n$ es el dominio de los posibles casos.

A partir de esta información, cuando se presenta un caso nuevo $y = (y_1, y_2, \ldots, y_n)$, el sistema debe decidir la clase c_y de entre el conjunto de clases posibles $\{C_1, C_2, \ldots, C_d\}$ a la que pertenece este caso.

Este trabajo se va a centrar en la situación en la que se tiene el conjunto finito de clases y ante un caso nuevo el sistema decide la clase a la que pertenece de entre ese conjunto finito de clases. También vamos a considerar que los atributos van a tomar valores numéricos precisos (bien continuos o discretos) y sin ruido. Pero cada atributo A_i va a tomar los valores dentro de un rango conocido $[a_i, b_i]$ con $a_i, b_i \in \mathrm{R}$ en el caso continuo, y en el caso discreto (numérico o simbólico) de entre un conjunto finito de valores posibles $\{V_i^1, \cdots, V_i^{m_i}\}$ con $m_i \in \mathrm{N}$ el número de valores que puede tomar el atributo A_i.

Se puede plantear el problema de la clasificación de forma aún más general, de manera que primero se debe determinar cuantas y cuales son las clases que deben utilizarse para la clasificación, y después se debe clasificar los distintos casos dentro de alguna de esas clases. Este problema se conoce con el nombre de "clustering", y tradicionalmente se han utilizado métodos estadísticos para resolverlo. Pero en este

trabajo no lo vamos a considerar.

3.2 Algunos Antecedentes Históricos

Para realizar las tareas de clasificación se puede utilizar una gran variedad de métodos. Tradicionalmente por ejemplo se han empleado métodos estadísticos, pero en este trabajo nos vamos a centrar en el campo de la Inteligencia Artificial (I.A.).

En general (y en particular también dentro de la I.A.) se suele dividir la clasificación en dos etapas. Se considera que en una primera etapa se realiza aprendizaje con algunos ejemplos y después se utiliza ese conocimiento adquirido para realizar clasificación de casos nuevos.

Tradicionalmente en la I.A. se entiende el aprendizaje como la adquisición de conocimiento estructurado, por ejemplo en forma de conceptos o reglas de producción. Pero en este trabajo vamos ser menos restrictivos y vamos a considerar que un sistema aprende o realiza aprendizaje cuando mejora su comportamiento.

A continuación se muestra el funcionamiento de algunos métodos que realizan aprendizaje. Esta lista de métodos no debe considerarse, ni mucho menos, exhaustiva, sino símplemente ilustrativa de algunas ideas, que en opinión del autor, son interesantes, y que han surgido dentro del área de la I.A. en los últimos años. Realizar un análisis exhaustivo de todos los métodos utilizados en I.A. para realizar aprendizaje es una tarea extensa que queda fuera del ámbito de este trabajo. Símplemente se muestra el funcionamiento de algunos métodos a título ilustrativo. Debe tenerse en cuenta que la inclusión de estos métodos y no otros, puede considerarse en última instancia casi arbitraria, ya que no existe una razón especial para realizar esta "mini–selección" de métodos de aprendizaje y no otra.

3.2.1 Clasificación mediante Arboles de Decisión

Los árboles de decisión o de clasificación fueron popularizados por Quinlan [Qui86]. Uno de los primeros y probablemente el más famoso es el ID3. Es el que describiremos en este apartado, y a lo largo del trabajo se identificará Arboles de Decisión con

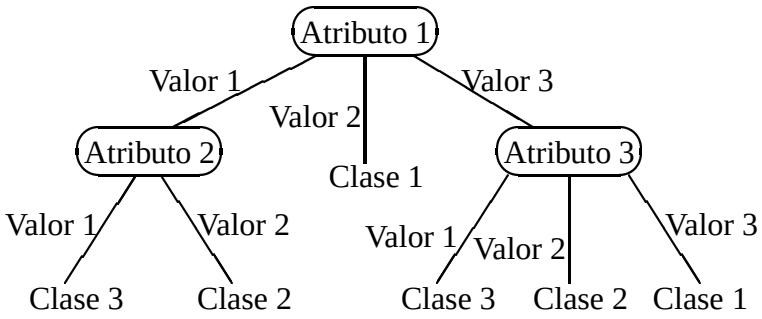


Figura 3.1: Árbol de Clasificación típico

el método ID3 a no ser que se especifique lo contrario de forma explícita o se hable de la familia de métodos de clasificación mediante Árboles de Decisión. En esta familia de métodos de aprendizaje el conocimiento adquirido a partir de los datos de entrenamiento se representa en forma de un árbol de decisión o clasificación.

El método ID3 tiene la limitación de que cada atributo puede tomar un conjunto discreto de valores mutuamente excluyentes. Posteriormente se han seguido desarrollando algorítmos de clasificación basados en árboles de decisión. Se han mejorado los resultados de ID3 y reducido restricciones y limitaciones. Como ejemplo más famoso de estos trabajos cabe citar el algorítmo C4.5 [Qui93].

Un árbol de clasificación es un árbol que tiene cada nodo etiquetado con un atributo, cada rama etiquetada con un valor del atributo asociado al nodo del que procede, y cada hoja etiquetada con el nombre de una clase. En un árbol de decisión las hojas son nombres de clases, los demás nodos representan comprobaciones sobre los atributos, con una rama para cada resultado posible. La estructura de un arbol de clasificación típico se muestra en la figura 3.1.

El mecanismo de inferencia para realizar la clasificación es el siguiente: cuando se nos presenta un objeto nuevo que queremos clasificar, comenzamos en el nodo raíz del árbol, evaluamos la comprobación y tomamos la rama apropiada según el resultado. Este proceso continúa hasta que llegamos a un nodo hoja, y entonces decimos que el objeto pertenece a la clase que aparece en ese nodo hoja. Así, siempre se recorre el árbol de clasificación partiendo del nodo raíz, comprobando el atributo que indican los nodos que nos encontramos y siguiendo por las ramas

etiquetadas con el valor del objeto nuevo para ese atributo, hasta llegar a un nodo hoja donde encontramos la clasificación que buscamos.

La tarea de aprendizaje consiste en obtener un árbol que permita clasificar correctamente objetos atendiendo al valor de sus atributos. Pero existen muchos árboles que clasifican correctamente los ejemplos de entrenamiento. ¿Cuál debemos elegir? Pues uno que también clasifique correctamente todos los ejemplos que se le puedan presentar al sistema. Pero a priori no tenemos esa información. Es necesario hacer algunas suposiciones, y aquí entra en juego el *Principio de Economía la Ockham* (también conocido como "Navaja de Ockham"), que se suele enunciar como: "el mundo es inherentemente simple" o "no hay que multiplicar los entes sin necesidad", es decir, que no hay que suponer la existencia de más entidades de las estrictamente necesarias para explicar los hechos. Aplicado a árboles de decisión el principio de Ockham nos diría que el árbol de decisión más simple o menor que clasifica correctamente los ejemplos de entrenamiento es el que tiene mayor probabilidad de clasificar correctamente objetos desconocidos.

Por tanto, un posible algorítmo consiste en generar todos los árboles de decisión posibles que clasifican correctamente el conjunto de entrenamiento y seleccionar el más sencillo de acuerdo al principio de economía de Ockham. Pero el número de árboles que deberíamos generar es muy grande y este sistema resulta inviable.

Para construir el árbol se puede utilizar el algoritmo ID3 [Qui86]. La idea consiste en elegir atributos e ir haciendo particiones sucesivas del conjunto de ejemplos de entrenamiento en función de los valores que presentan para esos atributos. Así hasta que llegamos a obtener particiones en las que todos los ejemplos de una misma partición pertenecen a la misma clase. Cuando posteriormente se intenta clasificar un objeto nuevo se comprueba a qué partición pertenece y se le asigna la clase que tienen todos los ejemplos de esa partición.

Más formalmente podemos decir que comenzamos con el nodo raíz conteniendo todos los ejemplos de entrenamiento. En cada nodo, el procedimiento detiene la expansión cuando todos los ejemplos que quedan son de la misma clase o representan la misma decisión. En este caso, el nodo se convierte en una hoja y se etiqueta con esa clase. Si se debe expandir el nodo, entonces se elige uno de los atributos

que no ha sido utilizado todavía para que en el nodo se compruebe ese atributo. El atributo elegido podrá tomar un conjunto finito de valores o bien se hace una partición de su dominio. En cualquier caso tendremos una partición del conjunto de ejemplos según el valor que toman en ese atributo. Se crea tantos nodos hijos como particiones tengamos, cada uno unido por un arco etiquetado con uno de los valores del atributo y conteniendo los ejemplos que toman ese valor. Y se repite el proceso para los nuevos nodos.

Para elegir el atributo que se debe comprobar se utilizan criterios heurísticos que deben intentar que el árbol resultante sea sencillo. Hay que elegir los atributos que sean más discriminantes. Un método basado en la teoría de la información se ha mostrado especialmente útil. En particular se trata de elegir aquel atributo que proporcione mayor ganancia de información.

Formalmente podemos calcular la cantidad de información de un nodo P como:

$$I_P = \sum_{i=1}^{d} p_i \times \log_2 p_i \tag{3.1}$$

donde $C = \{C_1, C_2, \cdots, C_d\}$ es el conjunto de decisiones y p_i es la probabilidad de la decisión i-ésima C_i en el nodo.

Para cada atributo no utilizado todavía, se calcula la cantidad de información de todos los nodos que se construirían si se eligiera ese atributo y se calcula $E(A)$ (información esperada del árbol que tiene como raíz al atributo) como la media de la información contenida en todos los nodos hijo ponderada por su probabilidad. Es decir, si el atributo A puede tomar v valores entonces genera v nodos $P_1, P_2, \cdots, P_v$ hijos del nodo P, cada nodo P_i con n_i ejemplos. La probabilidad de uno de los nodos P_i es la proporción de ejemplos de P que pertenecen al nodo, es decir $\frac{n_i}{n}$ donde $n = n_1 + n_2 + \cdots + n_v$. Por tanto podemas calcular la cantidad de información esperada del árbol que tiene como raíz al nodo A como:

$$E(A) = \sum_{i=1}^{v} \frac{n_i}{n} \times I_{P_i} \tag{3.2}$$

Se elige el atributo que proporciona mayor ganancia de información, es decir, que maximiza

$$gan(A) = I_P - E(A) \tag{3.3}$$

Nombre	Pelo	Estatura	Peso	Protector Solar	Resultado
María	rubio	media	ligero	no	Quemaduras
Josefa	rubio	alta	medio	sí	Nada
José	castaño	baja	medio	sí	Nada
Dolores	rubio	baja	medio	no	Quemaduras
Emilio	pelirrojo	media	pesado	no	Quemaduras
Pedro	castaño	alta	pesado	no	Nada
Juan	castaño	media	pesado	no	Nada
Juana	rubio	baja	ligero	sí	Nada

Tabla 3.1: Ejemplo de Datos de Entrenamiento para ID3

El algorítmo se muestra en la figura 3.2.

Como ejemplo, si realizamos el entrenamiento con los datos de la tabla 3.1, obtenemos el árbol de clasificación que aparece en la figura 3.3.

En este proceso de construcción del árbol nos podemos encontrar con algunos problemas. Puede suceder que el nodo *P* no contenga ningún ejemplo de alguno de los valores del atributo seleccionado *A*. En este caso nos encontraríamos con un nodo hoja que no contiene ningún ejemplo y la clasificación falla si algún objeto llega a esa hoja. También es posible que en un nodo ya hayamos usado todos los atributos pero todavía haya ejemplos de varias clases en una misma partición. En esta situación, nos encontramos con ejemplos que tienen todos los atributos iguales pero pertenecen a distinta clase. Esto se puede deber a varias causas:

- Necesitamos usar más atributos para ser capaces de distinguir unos objetos de otros (bien porque hemos considerado pocos atributos o bien porque son poco predictivos). La solución consiste en elegir con cuidado los atributos que consideramos de los objetos, o al menos tomar algunos más.

- Nuestros ejemplos tienen “ruido”, es decir, alguno de los valores de los atributos o la clase son erróneos. La solución puede consistir en usar algún criterio de parada. Por ejemplo, cuando en un nodo el porcentaje de objetos de una clase alcance un determinado umbral se dice que es un nodo hoja y se etiqueta

Dado un nodo P y el conjunto BC de casos de entrenamiento

Inicialmente considerar el árbol con un solo nodo, y ese nodo raíz contiene todos los ejemplos de entrenamiento BC.

Hasta que todos los ejemplos en cada nodo hoja sean de la misma clase:

1. Elegir un nodo hoja P donde haya ejemplos de más de una clase.
2. Calcular I_P, la cantidad de información del nodo P.
3. Considerar el conjunto A de los atributos que no aparecen en el camino que va desde el nodo raíz al nodo hoja P. Es decir, los que no se han utilizado todavía para llegar a obtener esta partición.
4. Para cada atributo $A \in A$ calcular E(A), la cantidad de información esperada del árbol que contiene como raíz a A.
5. Seleccionar el atributo $A \in A$ que proporciona mayor ganancia de información, es decir, que maximiza $gan(A) = I_P - E(A)$.
6. Reemplazar el nodo P por un nodo etiquetado con el atributo A, crear tantos nodos hijos de P como valores pueda tomar el atributo A, etiquetar cada arco que une el nodo P con un nodo hijo suyo con uno de los valores que puede tomar el atributo A, y colocar en el nodo hijo los ejemplos del nodo P que tienen ese valor para el atributo A.

Figura 3.2: Algorítmo de construcción del Árbol de Decisión con ID3

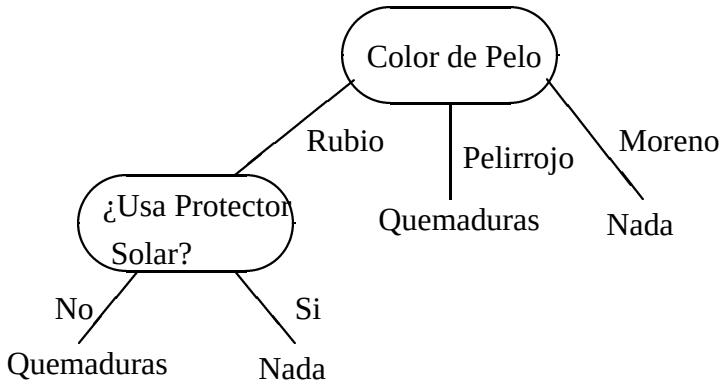


Figura 3.3: Árbol de Clasificación Resultado con ID3

con esa clase.

- Estamos en un ambiente impredecible, donde no se puede predecir la clase de un objeto dados cualquiera de sus atributos. En esta situación no se puede realizar tareas de clasificación y por lo tanto no tiene sentido aprender árboles de decisión.

Además hay que tener en cuenta que si el dominio de alguno de los atributos es continuo (no es simbólico ni finito), entonces necesitaremos hacer una discretización en intervalos. La elección de los intervalos normalmente es de difícil justificación, el sistema no muestra un comportamiento gradual y el árbol resultante no resulta comprensible. Otro problema adicional se presenta cuando consideramos que el valor de algunos de los atributos puede ser desconocido.

Los árboles de decisión tienen problemas cuando se trabaja con incertidumbre e imprecisión, cuando los valores de los atributos de los ejemplos tienen ruido, cuando los atributos toman valores continuos y se debe realizar una discretización, o cuando el valor de algún atributo no es conocido.

Una forma de abordar estos problemas consiste en combinar los árboles de decisión con una representación difusa de los valores de los atributos.

Los árboles de decisión difusos son básicamente como los árboles de decisión normales. Al igual que antes, el árbol tiene nodos para comprobar atributos, pero ahora las ramas tienen asociadas etiquetas lingüísticas (o subconjuntos difusos del

dominio del atributo) que representan las distintas alternativas. Además, ahora los nodos hoja pueden tener asociadas una o varias clases y sus valores de certeza.

Por tanto, básicamente lo que se ha hecho es transformar los atributos en variables lingüísticas y añadir valores de certeza a las clasificaciones.

El mecanismo de inferencia es similar al utilizado en los árboles de decisión normales. Pero ahora al evaluar un atributo puede tener valores de pertenencia mayores que cero en más de una de las etiquetas lingüísticas. Por tanto, éstas ya no forman una partición del conjunto de ejemplos del nodo y ahora es posible que en algunos nodos debamos explorar varias ramas. El resultado es que se recorre en paralelo una parte del árbol, y no sólo un camino como se hacía cuando la evaluación era única.

Se deben decidir tres operaciones [UOH$^+$94]:

- Operación para agregar los valores de pertenencia a las alternativas de las ramas (cómo encaja el objeto en un camino desde el nodo raíz a un nodo hoja, dado el grado de pertenencia a las etiquetas lingüísticas).
- Operación para agregar los valores de pertenencia a un camino y la certeza de la clase del nodo hoja.
- Operación para agregar certezas de la misma clase provenientes de distintos caminos.

En [UOH$^+$94] se utiliza el producto en las dos primeras y la suma en la tercera (con una normalización cuando la certeza total sobrepasa la unidad). Y por último se elige la clase que presenta una mayor certeza.

La construcción del árbol de decisión difuso se puede realizar de forma muy parecida a ID3 [UOH$^+$94]. Pero para seleccionar el atributo en lugar de calcular la ganancia de información a partir de las probabilidades de los datos de ejemplo, construye las probabilidades a partir de los valores de pertenencia de los ejemplos (porque ahora se trabaja con conjuntos difusos). Además se utiliza también algunos umbrales que hacen que el algoritmo pare cuando en un nodo casi todos los ejemplos son de la misma clase o cuando ya hay muy pocos ejemplos.

Otra alternativa a la inclusión explícita de valores de certeza consiste en adaptar el método del cálculo del centro de gravedad usado en conjuntos difusos para resolver los conflictos [Jan93].

3.2.2 Clasificación mediante el uso de Reglas

En particular, este apartado se va a centrar en la utilización de reglas para representar el conocimiento que adquiere el sistema, y su posterior utilización en las labores de clasificación.

Las reglas son un mecanismo bastante potente de representación de conocimiento, que se vienen utilizando con éxito desde hace años. En general una regla tiene la forma "**SI** *antecedente* **ENTONCES** *consecuente*", donde el antecedente es una conjunción o disyunción de predicados, el consecuente es un hecho o afirmación y significa que si se verifica el antecedente entonces se cumple el consecuente.

En general, para los problemas de clasificación vamos a considerar que las reglas tienen la forma:

$$\textbf{SI} \quad A_{i_1} \text{ es } L_1 \text{ y } A_{i_2} \text{ es } L_2 \text{ y } \cdots \text{ y } A_{i_h} \text{ es } L_h \qquad \textbf{ENTONCES} \quad y \text{ es } L \tag{3.4}$$

donde $h \leq n$, $A_{i_j} \in \{A_1, A_2, \cdots, A_n\}$ (es uno de los atributos), L_j es un valor o conjunto de valores que puede tomar el atributo A_{i_j}[1], y es "la clase", y $L \in \{C_1, C_2, \cdots, C_d\}$ es la clasificación que hace esta regla. La conjunción de antecedentes se puede expresar mediante la descomposición en varias reglas con el formato anterior.

Para realizar la clasificación se utilizan las reglas que se conocen y los atributos del ejemplo que se debe clasificar como hechos conocidos. Mediante un mecanismo de inferencia, a partir de esos hechos y esas reglas se llega a deducir la clase a la que pertenece el ejemplo.

Probablemente el problema más complejo sea el del aprendizaje de las reglas. Éste se puede plantear desde muchos puntos de vista, entre otros elicitación de conocimiento de un experto, generalización de ejemplos del conjunto de entrenamiento en reglas que no tienen contraejemplos, asignación de un grado de peso

[1] Con este formato general se puede considerar también la utilización de reglas difusas para la clasificación sin más que considerar que L_j es conjunto difuso o una etiqueta lingüística

o confianza en una regla de acuerdo a la probabilidad de que se verifiquen sobre el conjunto de entrenamiento, algoritmos genéticos y árboles de decisión.

La elicitación de conocimiento de un experto es una labor larga, tediosa y compleja que requiere un gran esfuerzo por parte del experto y del Ingeniero de Conocimiento. Además se trata de un método no automático y no lo vamos a tratar en este trabajo.

El problema del aprendizaje de las reglas se puede plantear como un problema de búsqueda, donde se tiene el conjunto de ejemplos y se intenta buscar un conjunto de reglas que clasifique bien los casos que se plantean al sistema. Como el espacio de búsqueda es muy grande, se aborda utilizando técnicas de búsqueda aproximadas pero que proporcionan buenos resultados. En particular, para el aprendizaje de reglas se ha utilizado técnicas de búsqueda basadas en Algoritmos Genéticos.

Además, para el aprendizaje de reglas se puede realizar el aprendizaje mediante otros métodos que utilizan formalismos diferentes de representación de conocimiento, y después transformar esa representación a reglas. En este sentido se analiza el uso de Arboles de Decisión.

3.2.2.1 Transformación de Arboles de Decisión en Reglas

Podemos ver que un árbol de decisión es capaz de representar las reglas de clasificación que hemos aprendido. Pero además, dado un árbol de decisión se puede representar esas mismas reglas de clasificación mediante reglas de la forma SI-ENTONCES.

Una vez que hemos construido el árbol de decisión, es relativamente sencillo convertirlo en un conjunto de reglas equivalentes. Sólo hay que seguir los caminos desde el nodo raíz hasta todos los nodos hoja en el árbol de decisión, almacenar las comprobaciones y los resultados como antecedentes de la regla y la clasificación que hace el nodo hoja como consecuente. Como resultado obtendremos tantas reglas como nodos hoja tenía el árbol de clasificación. Cada regla tendrá como antecedente una conjunción de comprobaciones de valores de atributos (tantas como la longitud del camino seguido desde el nodo raíz hasta alcanzar el nodo hoja), y

como consecuente una clasificación.

Una vez que se ha obtenido el conjunto de reglas, se puede intentar simplificar cada una de las reglas y después eliminar aquellas que no son útiles. Para simplificar una regla se puede intentar eliminar alguno de sus antecedentes sin cambiar lo que hace la regla sobre los ejemplos de entrenamiento. Es decir, se simplifica las reglas individuales eliminando los antecedentes que no tienen importancia. Esto se puede hacer por ejemplo construyendo lo que los estadísticos llaman una *tabla de contingencia*, que indica el grado en que un resultado es contingente a una propiedad. Posteriormente, para eliminar reglas se puede sustituir aquellas reglas que tienen la clasificación más común por una regla por defecto que se activa si no es aplicable ninguna de las otras. En este punto es interesante hacer notar que no nos vamos a encontrar con reglas que subsumen a otras porque todas proceden del mismo árbol de decisión y supondría que al recorrer el árbol de decisión desde el nodo raíz hacia un nodo hoja (regla más específica) hemos pasado antes por otro nodo hoja (que correspondería a la regla más general). Por este motivo es imposible antes de hacer la simplificación de las reglas, y además ésta tampoco altera la situación.

También nos podemos plantear partir de un árbol de decisión difuso y transformarlo en un conjunto de reglas difusas. La forma de hacerlo es análoga a como se transforman los árboles de decisión normales en reglas de la forma IF-THEN. Los árboles de decisión difusos asocian etiquetas lingüísticas a las ramas para representar las distintas alternativas. Por eso al almacenar las comprobaciones y los resultados como antecedentes de la regla ya obtenemos una regla difusa debido a que los resultados son etiquetas lingüísticas.

La clasificación que hace el nodo hoja nos da el consecuente de la regla, y el valor de certeza de la clasificación lo podemos utilizar por ejemplo como valor de certeza de la regla. Si en este nodo se ha almacenado más de una clasificación entonces se puede descomponer en varias reglas (tantas como clasificaciones haga el nodo), cada regla tendría una de las clasificaciones como consecuente y su factor de certeza sería el factor de certeza de esa clasificación. Por lo tanto, el número de reglas difusas que obtendremos será la suma de las clasificaciones que haya en los nodos hoja del árbol de clasificación difuso.

3.2.2.2 Aprendizaje de Reglas mediante Algoritmos Genéticos

Los algoritmos genéticos están inspirados en los procesos evolutivos que ocurren en la naturaleza y que hacen que las especies vayan evolucionando y mejorando su adaptación al medio. De forma análoga se pretende partir de una población de soluciones y realizar una "evolución" que tiene algunos paralelismos con la que se realiza en la naturaleza, para obtener poblaciones que tengan soluciones cada vez mejores al problema que nos planteamos. Hay que mencionar que los algoritmos genéticos se engloban dentro de las técnicas aproximadas, aunque pueden combinarse con otras técnicas para mejorar sus resultados (algoritmos genéticos híbridos).

Hay muchas variantes de algoritmos genéticos, pero todos comparten unos mecanismos de funcionamiento fundamentales:

- Operan sobre una población de individuos que inicialmente se genera de forma aleatoria (aunque se pueden combinar con otras técnicas y partir de una población inicial con mejores individuos).
- Cambian en cada iteración la población de individuos siguiendo los pasos:
    1. Evaluación de los individuos de la población para calcular su "adaptación" al medio (es decir, lo buena que es la solución que representa cada individuo).
    2. Selección de un conjunto de individuos de la población actual para realizar la reproducción sobre la base de su conveniencia o adaptación relativa.
    3. Recombinación de los individuos "padres" para formar una nueva población usando los operadores de cruce y mutación que realizan las funciones equivalentes a la "reproducción" y "mutación" que se produce en la naturaleza.

Los individuos que resultan después de estos tres pasos son los que forman la siguiente población, y el proceso se itera hasta que el sistema deja de mejorar.

Dejar de mejorar se puede entender como realizar un número de iteraciones

determinado, que el mejor individuo de la población deje de mejorar, o que la población en media no mejore.

Cada individuo se representa generalmente mediante una cadena de bits (equivalente a cromosomas en la naturaleza) que codifica los valores de las variables que intervienen en el problema. Los operadores de cruce y mutación están adaptados a esas cadenas de bits (o cromosomas) y las manipulan para obtener nuevas cadenas de bits que codifican nuevas soluciones al problema que están bien adaptadas.

Asociada a los operadores genéticos de cruce y mutación existe una probabilidad de cruce y una probabilidad de mutación que mide la probabilidad de que un individuo sufra una mutación o participe en un cruce.

Además, durante cada iteración el algoritmo mantiene una población de soluciones, y cada individuo se evalúa mediante una función que nos da una medida de su adaptación (o lo que es equivalente, lo buena que es la solución que codifica). Esta medida de adaptación se utiliza para calcular la probabilidad de reproducción de los individuos, de forma que tengan más probabilidad de reproducirse los individuos mejor adaptados (es decir, que se transmitan de una generación a la siguiente las características de los individuos que representan las mejores soluciones).

En general, dentro de los Algoritmos Genéticos se suele hablar de dos o tres enfoques distintos para abordar el problema del aprendizaje de reglas:

- *Método de Pittsburgh*. Una forma natural de abordar el problema consiste en representar todo un conjunto de reglas como un individuo. Ese individuo se representará como una cadena de 0's y 1's o será una cadena sobre algún alfabeto. Así, se ve la población como conjuntos de reglas candidatos, y se usa la selección y los operadores genéticos para producir una generación nueva de conjuntos de reglas. Este enfoque fue desarrollado por De Jong y se conoce con el nombre de método de Pittsburgh.

- *Método de Michigan*. Otro enfoque del problema consiste en considerar los miembros de la población como reglas individuales y toda la población forma el conjunto de reglas que queremos aprender. Este enfoque fue desarrollado por Holland y se conoce con el nombre de método de Michigan.

- *Programación Evolutiva*. También se puede distinguir un tercer enfoque basado en programación evolutiva, donde se incorpora conocimiento específico del dominio. En la programación evolutiva se utiliza:
  - una representación de los individuos de alto nivel que resulta más natural al problema, en lugar de la codificación como una cadena de 0's y 1's,
  - operadores genéticos específicos, adecuados a la representación elegida.

  No obstante en el problema del aprendizaje de reglas se puede considerar que la distinción es muy sutil y englobar este tercer método dentro del método de Pittsburgh.

3.2.3 Teoría del Ejemplar Generalizado Anidado (NGE)

Salzberg [Sal91] describe una familia de algorítmos de entrenamiento denominada Ejemplar Generalizado Anidado (NGE) (en inglés *Nested Generalized Exemplar*). En NGE los objetos se almacenan como puntos en un espacio Euclídeo n-dimensional, donde cada dimensión corresponde a un atributo de los ejemplos. El aprendizaje consiste en la generalización de varios de esos ejemplos de entrenamiento en un hiperrectángulo que los incluye. Los hiperrectángulos son paralelos a los ejes y se pueden anidar unos dentro de otros a cualquier nivel de profundidad para representar excepciones a los hiperrectángulos más exteriores. Como resultado se almacenan hiperrectángulos que son generalizaciones de varios ejemplos junto con puntos aislados que son ejemplos no generalizados. Para referirnos tanto a unos como a otros se utiliza el término ejemplar (en inglés *exemplar*).

Cuando se debe clasificar un ejemplo nuevo se calcula la distancia de ese ejemplo a todos los ejemplaress almacenados. Si el ejemplo está contenido en un hiperrectángulo entonces la distancia entre ambos es cero, en otro caso la distancia entre el ejemplo nuevo y un hiperrectángulo se calcula como la distancia euclídea del ejemplo al lado más cercano del hiperrectángulo. Al ejemplo nuevo se le asigna la clasificación del ejemplar (ejemplo o hiperrectángulo) que se encuentra a menor distancia.

Además a cada ejemplar almacenado se le asocia un peso propio que representa

la fiabilidad o probabilidad de que haga predicciones correctas. Simplemente mide la frecuencia con que el ejemplo ha hecho clasificaciones correctas en el pasado y se va modificando según va realizando buenas o malas predicciones sobre los ejemplos nuevos. Este peso se puede entender también como "en esta región del espacio la fiabilidad de mi predicción es *n*".

Utilizando la notación de Salzberg, dado E el ejemplo nuevo y H el ejemplar almacenado (que puede ser un punto o un hiperrectángulo), se calcula la distancia D_{EH} entre ambos como:

$$D_{EH} = w_H \sqrt{\sum_{i=1}^{m} \left(w_i \frac{dif_i}{max_i - min_i} \right)^2}$$

donde w_H es el peso asociado a H, w_i es el peso asociado a la característica i, min_i y max_i son los valores mínimo y máximo de esa característica, y m es el número de características que se conocen sobre E.

dif_i mide la diferencia entre E y H en la característica i-ésima y se calcula como:

$$dif_i = E_i - H_i \qquad \text{si } H \text{ es un punto}$$

$$dif_i = \begin{cases} E_i - H_{max} & \text{si } E_i > H_{max} \\ H_{min} - E_i & \text{si } E_i < H_{max} \\ 0 & \text{en otro caso} \end{cases} \qquad \text{si } H \text{ es un hiperrectángulo}$$

donde E_i es el valor de la i-ésima característica del ejemplo E. Si H es un punto entonces H_i es el valor de la i-ésima característica del punto H, y si H es un hiperrectángulo entonces H_{min} y H_{max} son los valores mínimo y máximo del intervalo de valores que presenta H para la i-ésima característica.

La función de distancia va ajustando sus pesos w_i para mejorar su comportamiento según se van realizando clasificaciones correctas y erróneas siguiendo un esquema de "aprendizaje por refuerzo" (en inglés *Reinforcement Learning*).

3.3 Uso del RBC en Problemas de Clasificación

Deseamos diseñar un sistema Basado en Casos que incorpore el conjunto de ejemplos *BC* como su Base de Casos, y que a partir de esos ejemplos o casos decida a

qué clase pertenece cada nuevo caso que deba clasificar.

Para realizar esto vamos a asumir que si dos casos son muy parecidos, va a ser muy probable que se clasifiquen de la misma forma. En la situación extrema, si tenemos dos casos e_1 y e_2, que en la Base de Casos aparecen clasificados respectivamente como c_1 y c_2 (lo podemos notar $(e_1,c_1),(e_2,c_2) \in BC$), podemos asumir que si $e_1 = e_2$, entonces $c_1 = c_2$ (ambos se clasifican igual). De no ser así estaríamos ante un problema en el que no es predecible la clase a la que pertenecen los casos (un caso en un momento pertenece a una clase y ese mismo caso en otro momento pertenece a otra clase distinta). De igual forma, vamos a suponer que si dos casos son muy parecidos entonces la clasificación que van a tener, cuando menos, es parecida.

En capítulos anteriores hablamos de Razonamiento Basado en Casos en general, considerándo los sistemas de la forma más general posible, y para afrontar cualquier tipo de problema. Ahora nos hemos centrado en problemas de clasificación y no se utilizan muchas de las posibilidades que nos brinda el RBC.

El comportamiento de un sistema típico de RBC para problemas de clasificación se va a reducir a los siguientes pasos:

1. Se plantea un caso e que debe ser clasificado.
2. Se selecciona de la Base de Casos BC un conjunto de casos $K = \{(e_1,c_1), (e_2,c_2), \cdots, (e_k,c_k)\}$ tal que $e_1, e_2, \cdots, e_k$ son los casos "más similares" a e.
3. Se combina las clasificaciones de los distintos casos recuperados para obtener una clasificación c para e.

En una primera impresión pueda parecer que se trata de un Sistema Basado en Casos "muy aligerado". Pero en estos pasos todavía se plantean los aspectos más importantes del Razonamiento Basado en Casos:

- El tamaño y la estructura de la Base de Casos
- La noción de similitud

- El problema de la recuperación de casos

- La noción de transformación de una solución.

Para problemas de clasificación la *transformación de la solución* que se suele utilizar es la identidad, y la *recuperación de casos* se suele reducir a buscar y devolver el caso más similar en la Base de Casos. Es decir, lo más usual es que directamente se proporciona como clasificación del nuevo caso la que tiene el caso conocido que más se le parece (se toma $c = c_1$).

Por tanto, el conocimiento que normalmente se utiliza para resolver el problema de clasificación está contenido en la Base de Casos y en el concepto de similitud que utilicemos.

Respecto del Conocimiento contenido en la Base de Casos debemos decir que para que se pueda clasificar correctamente los casos nuevos que se planteen, es primordial que dispongamos de suficientes ejemplos, que éstos sean representativos de lo que sucede en sus proximidades y que estén repartidos cubriendo todas las zonas del universo U (no hay sesgo en la muestra). Para trabajar vamos a suponer que si disponemos del suficiente número de ejemplos y éstos son elegidos aleatoriamente, entonces se van a cumplir las condiciones anteriores y la Base de Casos va a aportar el Conocimiento necesario para realizar clasificaciones correctas.

Aquí estamos considerando que si dos ejemplos son parecidos, entonces se clasificarán de la misma o parecida forma. No es necesario disponer de un modelo general explícito, claro y preciso sobre el dominio del problema. Implicitamente estamos asumiendo que los atributos seleccionados aportan suficiente información y que los casos contienen suficiente conocimiento sobre el comportamiento en ese dominio.

El Concepto de similitud lo vamos a tratar aparte en el apartado siguiente. En el capítulo 4 se volverá a retomar el tema de la noción de similitud, la recuperación de casos y la transformación de soluciones, pero ya para plantear diversas formas de concretar estos conceptos.

3.4 El concepto de Similitud: Medidas de Distancia y Similitud

Dado un caso x que debemos clasificar, ¿cual es el caso más "similar" a x de entre los casos que conocemos (contenidos en la Base de Casos BC)? Esta es una pregunta clave que debemos hacernos al construir cualquier sistema que utilice Razonamiento Basado en Casos. La respuesta va a tener una importancia crucial en la fase de recuperación del razonador, y por tanto en el comportamiento de todo el sistema.

En cualquier caso, la respuesta no va a ser única, y va a depender en gran medida de lo que entendamos por similar, el dominio en que estemos trabajando y de la información de que podamos disponer. En este trabajo no nos centramos en ningún dominio particular, sino que nos planteamos la situación en que vamos a poder poner a trabajar al sistema en un problema de clasificación, en principio en un dominio cualquiera. Por tanto no vamos a contar con información sobre el dominio. La única información de que vamos a disponer es símplemente el valor de los atributos de una serie de ejemplos y su clasificación. Es decir, una situación típica de "Machine Learning".

El concepto de similitud lo podemos representar de varias formas [Rit92]:

1. Un predicado binario $SIM(x,y) \subseteq U \times U$, representando "$x$ e y son similares"
2. Un predicado binario $DISSIM(x,y) \subseteq U \times U$, representando "$x$ e y son no similares"
3. Una relación ternaria $S(x,y,z) \subseteq U^3$, representando "y es al menos tan similar a x como z lo es a x"
4. Una relación cuaternaria $R(x,y,u,v) \subseteq U^4$, representando "y es al menos tan similar a x como v lo es a u"
5. Una función $sim(x,y) : U \times U \rightarrow [0,1]$ midiendo el grado de similitud entre x e y.

6. Una función $d(x,y) : U \times U \rightarrow$ R midiendo la distancia entre x e y.

Todas representan similitud, pero algunas pueden ser más expresivas que otras (de hecho aparecen en orden de menor a mayor grado de expresividad). Por ejemplo, la relación $S(x,y,z)$ permite definir el concepto "y es más parecido a x" mientras que $SIM(x,y)$ no. Estas formas de representar similitud casi siempre son simétricas, y pueden, en principio, ser reflexivas y transitivas, o no cumplir ninguna de estas propiedades.

Los axiomas básicos que normalmente se piden a una medida de similitud *sim* son:

1. $sim(x,x) = 1$ (reflexiva)
2. $sim(x,y) = sim(y,x)$ (simétrica)

Para la medida de distancia $d(x,y)$ hay que decir que no puede tomar valores negativos y debe ser reflexiva ($d(x,x) = 0$). Pero no tiene por qué cumplir otras propiedades como la simétrica o la desigualdad triangular, y puede ser $d(x,y) = 0$ para $x \neq y$. Por tanto, desde el punto de vista matemático, la medida de distancia d no tiene por qué ser una métrica.

A veces se llega más lejos incluso. Por ejemplo en [RA95] se propone un tipo de métrica de similitud local llamada AASM (asymmetric anisotropic similarity metric) donde la medida de similitud depende del punto de espacio desde donde se toma la distancia, y además es asimétrica (no se cumple la propiedad de simetría: $d(x,y) \neq d(y,x)$).

Como no disponemos de información específica del dominio del problema y solo contamos con valores de atributos, vamos a definir el concepto de *similitud* basándonos en el concepto de *vecino más cercano*. Además esto es muy usual al utilizar RBC para resolver problemas de clasificación. Por tanto el problema se va a reducir ahora al *Principio del Vecino más Cercano* (en inglés *Nearest Neighbour Principle*, o simplemente NNP).

Podemos enunciar este principio de dos formas equivalentes. Una utilizando medidas de similitud y otra utilizando medidas de distancia:

- Dado un caso actual x_0 y una medida $sim(x,y) : U \times U \rightarrow [0,1]$ de similitud entre casos, se selecciona un caso (x,c) de la Base de Casos *BC* tal que $sim(x_0,x)$ es maximal.

- Dado un caso actual x_0 y una medida $d(x,y) : U \times U \rightarrow$ R de distancia entre casos, se selecciona un caso (x,c) de la Base de Casos *BC* tal que $d(x_0,x)$ es minimal.

En este trabajo se va a utilizar esta última definición el Principio del Vecino más Cercano. Se ha preferido así porque al movernos en dominio numéricos, permite incorporar directamente funciones de distancia para determinar cual es el caso más parecido. Y la noción de distancia y las funciones de distancia están bastante arraigadas entre nosotros y resultan más intuitivas. El desarrollo posterior del trabajo utilizando medidas de similitud hubiera sido totalmente análogo. De hecho, dada una medida de distancia se puede definir fácilmente la medida de similitud equivalente y viceversa (en el sentido de distancia antes expresado).

Una vez que se tiene una medida de distancia, se puede obtener la medida de similitud definida como [Rit92]:

$$sim(x,y) = \begin{cases} 1 - \frac{d(x,y)}{max} & \text{si } d(x,y) \in [0,max] \\ 1 - \frac{d(x,y)}{1+d(x,y)} & \text{si } d(x,y) \text{ no está acotada} \end{cases} \tag{3.5}$$

Se dice que dos medidas *d* y *sim* son equivalentes si y solo si existe una función biyectiva $f : rango(d) \rightarrow rango(sim)$, tal que $f(0) = 1$ y $sim(x,y) = f(d(x,y))$. A menudo se suelen utilizar las funciones anteriores $f(z) = 1 - \frac{z}{1+z}$ si d no está acotada y $f(z) = 1 - \frac{z}{max}$ si *d* alcanza un mayor elemento *max*.

El Principio del Vecino más Cercano, como tantas otras cosas dentro del campo de la I.A., no tiene una justificación teórica sobre su buen comportamiento. Símplemente se ha comprobado de forma experimental que funciona bien.

Se suele utilizar la notación $x = NN(x_0, BC, d)$, para indicar que considerando la distancia *d*, *x* es el vecino más cercano de x_0 de entre los ejemplos que hay en la Base de Casos *BC*. Aquí se pone de manifiesto claramente que dado un caso x_0 *la elección de su vecino más cercano depende de* dos cosas:

- El estado actual de la Base de Casos *BC*.
- La medida o función de distancia que se considere.

Respecto del *estado actual de la Base de Casos* (o símplemente la Base de Casos), vamos a considerar que en principio es un parámetro que no podemos controlar. El sistema tiene un determinado conjunto de ejemplos inicial y después, con la experiencia, puede ir acumulando más ejemplos. Lo que sí vamos a poder controlar es la *función de distancia* que se va a utilizar para determinar cual es el vecino más cercano.

En una primera parte nos vamos a centrar en estudiar cómo pueden ser las funciones de distancia y a continuación proponer varias y estudiar su comportamiento con Bases de Casos reales que se utilizan dentro de la comunidad científica.

En una segunda parte vamos a estudiar qué sucede con la Base de Casos. A primera vista puede parecer preferible que cuantos más casos contenga mejor. Pero si es demasiado grande se pueden ralentizar y dificultar las tareas de recuperación de casos. Puede ser bueno en lugar de almacenar todos los casos conocidos, almacenar sólo los casos que realmente nos aportan información nueva no contenida en otros casos. Así se puede lograr mantener una Base de Casos relativamente pequeña (mejorando la recuperación) y tal vez conseguir mantener niveles similares de acierto en las tareas de clasificación. Aquí se va a estudiar qué criterios se pueden utilizar para no almacenar todos los casos, qué medidas de distancia obtienen buenos resultados con esos criterios, y el impacto que tiene reducir el tamaño de la Base de Casos sobre el comportamiento del sistema.

3.5 Funciones Clásicas de Distancia

Ya hemos visto que el conocimiento que normalmente se utiliza para resolver problemas de clasificación usando RBC está contenido en la Base de Casos y en el concepto de similitud que utilicemos. Si consideramos que dos casos similares van a tener valores de atributos parecidos, podemos definir la similitud entre casos utilizando una función de distancia. Así el concepto de similitud lo vamos a definir

asumiendo el *Principio del Vecino más Cercano* y por tanto el comportamiento de un sistema típico de RBC para problemas de clasificación se reduce a los siguientes pasos:

1. Se plantea un caso x que debe ser clasificado.
2. Se selecciona un caso (y, c_y) de la Base de Casos BC tal que y es el caso "más similar" a x. Es decir, dada una función de distancia d se selecciona el caso y que se encuentra a menor distancia de x: $d(x,y) \leq d(x,z) \qquad \forall z \in BC$.
3. Se transforma la clasificación c_y de y en una clasificación c_x para x.

Todo el peso de la noción de similitud recae sobre la función de distancia que elegimos, y por tanto esta elección va a resultar vital para que el Razonador Basado en Casos realice bien las tareas de clasificación.

Desde el punto de vista de la Geometría, una distancia o una métrica en un conjunto C es una aplicación $d : C \times C \rightarrow \mathrm{R}^+$ que cumple las siguientes propiedades:

1. $d(x,y) = 0 \Leftrightarrow x = y$
2. $d(x,y) = d(y,x)$
3. $d(x,z) \leq d(x,y) + d(y,z)$

Hay que tener en cuenta que lo único que necesitamos para utilizar el Principio del Vecino más Cercano es una función que nos indique el grado de similitud entre dos puntos, es decir, si se parecen los valores de sus atributos. Está claro que una función de distancia definida desde el punto de vista de las Matemáticas cumple esta misión, pero no es necesario exigir tanto. Por eso en este capítulo se van a relajar las propiedades que se exigen a las funciones y vamos a ver algunas "funciones de distancia" que no lo son desde el punto de vista de la Geometría, pero que nos sirven porque reflejan la similitud entre puntos del espacio.

En este capítulo se van a ver con más detalle las funciones de distancia que se pueden utilizar para definir "similitud". En primer lugar vamos a definir de forma genérica las funciones de distancia que utilizaremos, y veremos que básicamente las vamos a poder dividir en dos tipos:

- Función de Distancia Global
- Función de Distancia Local

atendiendo a si la medida de distancia se comporta de forma homogenea en todo el espacio o si depende de la zona del espacio que consideremos.

A continuación pasaremos a ver algunas funciones de distancia, unas incorporarán aprendizaje y otras no. En el capítulo 4 veremos algunos métodos para definir similitud que están basados en funciones de distancia.

3.5.1 Una Función de Distancia Genérica

Sea U el conjunto de ejemplos o puntos con que trabajamos. Sean $x,y \in U$ dos de esos puntos. Para cada uno tenemos n valores para los atributos $A_1,A_2,\ldots,A_n$, que en principio pueden ser numéricos continuos, numéricos discretos o incluso simbólicos; y una clasificación $c \in C$. En este trabajo vamos a considerar sólo valores numéricos. Esto en realidad no supone ninguna pérdida de generalidad porque siempre se puede establecer una biyección entre el conjunto de valores simbólicos y un conjunto de valores numéricos, y trabajar con los números en lugar de con los símbolos. Así vamos a poder definir nuestros puntos x e $y \in U$ como dos vectores de valores reales $x = (x_1,x_2,\ldots,x_n)$, $y = (y_1,y_2,\ldots,y_n)$, con $x_i,y_i \in D_i$ y $x,y \in D$, donde D_i es el i-ésimo dominio y $D = D_1 \times D_2 \times \cdots \times D_n$ es el dominio de los posibles puntos.

Cada uno de estos dominios D_i va a estar acotado, dentro de un rango conocido $[a_i,b_i]$ si el atributo A_i puede tomar valores continuos, o $\{V_i^1,\cdots,V_i^{m_i}\}$ en el caso discreto.

Podemos definir una función genérica $d(x,y) : D \times D \rightarrow \mathrm{R}$ que mide la distancia entre los puntos x e y como:

$$d(x,y) = Agr(w_1,w_2,\cdots,w_n,d_1(x_1,y_1),d_2(x_2,y_2),\cdots,d_n(x_n,y_n)) \tag{3.6}$$

donde:

- $d_i : D_i \times D_i \rightarrow \mathrm{R}$ es una medida de *distancia sobre una variable*, en el dominio i-ésimo que mide la distancia o diferencia que hay entre dos puntos de dicho dominio. Se usa para medir la distancia entre x_i e y_i, las componentes i-ésimas de x e y. Esta medida cumple la propiedad simétrica, es decir $d_i(x_i, y_i) = d_i(y_i, x_i) \qquad \forall x_i, y_i \in D_i$. Se suele calcular como:

$$d_i(x_i, y_i) = \begin{cases} |x_i - y_i| & \text{si } D_i \text{ es continuo} \\ 0 & \text{si } D_i \text{ es discreto y } x_i = y_i \\ 1 & \text{si } D_i \text{ es discreto y } x_i \neq y_i \end{cases}$$

- $w_i \in [0,1]$ es un peso asociado al dominio i-ésimo D_i que sirve para ponderar la importancia relativa de los distintos dominios. Se suele definir el *vector de pesos* $w = (w_1, w_2, \cdots, w_n)$ de tal manera que $\sum_{i=1}^{n} w_i = 1$.

- $Agr : \mathrm{R}^{2n} \rightarrow \mathrm{R}$ es una *Función de Agregación* que agrega los distintos pesos w_i y distancias parciales d_i sobre cada uno de los dominios D_i para obtener una única medida de total para el dominio D. Suele consistir en una función de combinación del peso y la distancia parcial de cada dominio (normalmente una potenciación y un producto) y una función de agregación de los valores de los diferentes dominios (normalmente un sumatorio y una raíz)

Esto quiere decir que podemos obtener diferentes funciones de distancia utilizando diferentes Distancias sobre una Variable y diferentes formas de combinar estas distancias: distintas Funciones de Agregación y distintos Vectores de Pesos.

El que hayamos realizado una definición de función de distancia con salidas acotadas en el intervalo [0,1] no resta generalidad. En algunas ocasiones puede tenerse una función no acotada [0,+∞), pero tendrá esta misma forma, y a efectos del estudio que se realiza en este trabajo resulta equivalente desde un punto de vista conceptual.

En lugar de utilizar la definición (3.6) vamos a utilizar otra notación más intuitiva y sencilla para definir una función de distancia genérica:

$$d(x,y) = Agr_{i=1}^{n} w_i \bigotimes d_i(x_i, y_i) \tag{3.7}$$

donde $\bigotimes$ es cualquier operador que permita combinar el peso w_i y el valor de la distancia parcial d_i asociada a un dominio. Normalmente se suele utilizar el operador producto.

Esta definición de distancia genérica debe considerarse simplemente como una notación "menos cargada" de la definición formal (3.6).

Como ejemplo podemos considerar:

$$n = 2 \qquad D_1 = D_2 = \mathrm{R} \qquad w_1 = w_2 = 1$$

$$d_i(x_i, y_i) = |x_i - y_i| \text{ para } i = 1, 2$$

$$Agr(w_1, w_2, d_1(x_1, y_1), d_2(x_2, y_2)) = \sqrt{\sum_{i=1}^{2} w_i \times d_i(x_i, y_i)^2}$$

que nos define la Distancia Euclídea usual en R^2:

$$d(x, y) = \sqrt{\sum_{i=1}^{2} (x_i - y_i)^2} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

En (3.6) y (3.7) hemos definido una *Función de Distancia Global* que va a permitirnos determinar cuando dos casos van a ser parecidos. Como situación extrema tendremos $d(x, y) = 0$ cuando ambos casos sean iguales, es decir, lo más similares posible. Según aumenta la distancia vamos teniendo casos menos similares.

Hemos mencionado antes que este tipo de distancias es global porque se comporta de igual forma en todo el espacio D. Ejemplos de este tipo de distancias son las distancias Euclídea, del Máximo, de la Suma, y en general todas las derivadas de las métricas L^n, donde se define $d(x, y) = \sqrt[n]{\sum_{i=1}^{n} d_i(x_i, y_i)^n}$.

En ocasiones puede resultar interesante tener en cuenta el contexto. Esto da lugar a las llamadas Funciones de Distancia Locales, que son aquellas que dependen del punto en el espacio de entrada D desde el que se mide la distancia. Usando métricas locales se pueden expresar fórmulas condicionales del tipo "Si el atributo A es mayor que 70% entonces usar sólo los atributos B y C para calcular la similitud" [RA95].

Teniendo esto en cuenta podemos definir una *Función de Distancia Local* genérica de la siguiente manera:

$$d(x,y) = Agr(w_1(x), w_2(x), \cdots, w_n(x), d_1(x_1,y_1), d_2(x_2,y_2), \cdots, d_n(x_n,y_n)) \quad (3.8)$$

donde ahora el peso asociado al dominio i-ésimo D_i es una función $w_i(x) : D \rightarrow [0,1]$ que depende del primer punto que se considera al medir la distancia, y las medidas de distancia parciales en las componentes no tienen por qué cumplir la propiedad simétrica ($d_i(x_i,y_i) \neq d_i(y_i,x_i)$).

Por tanto en general las distancias locales no cumplen la propiedad simétrica, es decir $d(x,y) \neq d(y,x)$, porque puede ocurrir que $w_i(x) \neq w_i(y)$ y también que $d_i(x_i,y_i) \neq d_i(y_i,x_i)$.

Igual que antes, también vamos a utilizar otra notación más sencilla e intuitiva pero menos formal:

$$d(x,y) = Agr_{i=1}^{n} w_i(x) \bigotimes d_i(x_i,y_i) \quad (3.9)$$

3.5.2 Funciones de Distancia Típicas de la Geometría

En este apartado vamos a considerar varias funciones de distancia globales. Es decir, al principio se define la función de distancia y ésta no cambia durante el tiempo en que se utiliza. Ninguna de ellas implementa ningún tipo de aprendizaje durante el funcionamiento del sistema.

En primer lugar se van a considerar las funciones de distancia típicas de la Geometría. A continuación consideraremos esas mismas funciones de distancia pero ponderando el peso de cada atributo por su correlación con la clase. Por tanto todas las funciones de distancia que se indican a continuación están basadas e influenciadas por las distancias típicas de la Geometría:

- Distancia Euclídea:

$$d_{Euclidea}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.10}$$

- Distancia del Máximo:

$$d_{Max}(x,y) = \max_{i\in\{1,\cdots,n\}} |x_i - y_i| \tag{3.11}$$

- Distancia de la Suma:

$$d_{Suma}(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{3.12}$$

- Distancia Euclídea con Correlación:

$$d_{EuclCorr}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2 \times |Corr_i|} \tag{3.13}$$

- Distancia del Máximo con Correlación:

$$d_{MaxCorr}(x,y) = \max_{i\in\{1,\cdots,n\}} |x_i - y_i| \times |Corr_i| \tag{3.14}$$

- Distancia de la Suma con Correlación:

$$d_{SumaCorr}(x,y) = \sum_{i=1}^{n} |x_i - y_i| \times |Corr_i| \tag{3.15}$$

donde $Corr_i$ representa la correlación del atributo i–ésimo con la clase.

Las distancias con correlación están basadas en las distancias usuales. Pero en lugar de ser totalmente ciegas en cuanto a la importancia de los atributos, tienen en cuenta la correlación estadística de cada uno de los atributos respecto de la clase.

Hemos supuesto que cada D_i está acotado dentro de un rango conocido $[a_i, bi]$. En la práctica podemos considerar incluso $D_i = [a_i, bi]$ sin perder generalidad. Esto nos lleva a otro inconveniente: si el rango del dominio de un atributo es mayor, entonces la distancia sobre esa variable puede alcanzar valores mayores, y por tanto tendrá más importancia que otras sobre la distancia total. Para contrarrestar este efecto no deseado, lo que se suele hacer es normalizar antes las distancias parciales de los distintos atributos dividiendolas por la amplitud del intervalo, para obtener resultados siempre en [0,1]. Reuniendo todo esto:

3.6 Medidas de Distancia basadas en bandas

Como ya se ha comentado anteriormente, una forma bastante frecuente y efectiva de definir una relación de similitud es mediante una función de distancia. Pero si consideramos sólo las distancias usuales, entonces no podemos obtener algunos tipos de relaciones de similitud que resultan bastante naturales. En esta sección vamos a analizar algunos de estos tipos de relaciones de similitud con el objetivo de encontrar medidas de distancia no usuales que permitan obtener algunos de esos tipos de relaciones de similitud naturales que no pueden aprehender las distancias usuales.

Dado un ejemplo x, ¿cuál es el caso más similar a x de entre todos los casos conocidos?, o, ¿cuál es el grado de similitud entre dos objetos x e y? La noción de similitud se puede expresar formalmente de muchas formas [Rit92]. Probablemente la manera más intuitiva es como una función $sim(x,y) : U \times U \rightarrow [0,1]$ que mide el grado de similitud entre x e y, de tal forma que $sim(x,y) = 1$ representa que x e y son totalmente similares (pueden ser el mismo objeto o no, pero son completamente similares en el sentido que se está considerando); $sim(x,y) = 0$ representa que x e y son no similares; y $sim(x,y) \in (0,1)$ representa que x e y tienen cierto grado de similitud. Esta medida de similitud $sim(x,y)$ debe ser reflexiva ($sim(x,x) = 1$), y normalmente es simétrica, ($sim(x,y) = sim(y,x)$), pero puede o no cumplir otras propiedades como la desigualdad triangular, y por supuesto es posible que $sim(x,y) = 1$ con $x \neq y$.

A menudo es difícil asignar un grado de similitud $sim(x,y) : D \times D \rightarrow [0,1]$ entre dos objetos x e y, representados mediante dos vectores $(x_1, x_2, \ldots, x_n)$ e $(y_1, y_2, \ldots, y_n)$ de n valores. En cambio suele resultar más sencillo definir medidas parciales de similitud para cada atributo $sim_i(x_i, y_i)$. Pero, ¿cómo se puede combinar esa información para obtener un único valor que represente la similitud total entre x e y? Una forma efectiva y bastante utilizada para superar estas dificultades consiste en emplear distancias, porque nos resultan familiares y, desde un puntos de vista práctico, es sencillo transformar una medida de distancia en una medida de similitud.

Se han realizado bastantes estudios sobre similitud ([PLA96] [Rit92] [Rit95]),

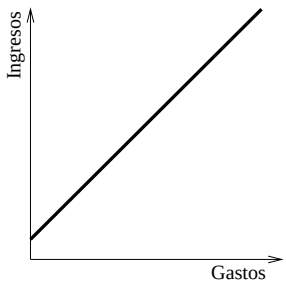


Figura 3.4: Compañías con beneficios iguales.

y sobre las distancias y los métodos basados en distancias (por ejemplo [RA95] [Wet94]). Este apartado se va a centrar en la relación entre la noción de similitud y la noción de distancia, y la forma de definir funciones de distancia y similitud útiles. Es interesante el sentido "suave" de este tipo de funciones debido a la salida gradual que proporcionan (entre 0 y 1 para la similitud, y entre 0 y $+\infty$ para la distancia), y por tanto su relación con otros campos como los conjuntos difusos [BM97].

Si consideramos sólo las distancias normales de la Geometría, entonces no se pueden obtener algunos tipos de relaciones de similitud que resultan bastante naturales. Por ejemplo, en problemas de clasificación conocemos el valor de algunas características y una clase de algunos ejemplos y el objetivo es encontrar la clase correcta para casos nuevos. A menudo la clave está en la proporción o la diferencia entre atributos. Por ejemplo, si conocemos los ingresos y gastos de algunas compañías y la clase son los beneficios de la compañía, los ejemplos que pertenecen a la misma clase (y en ese sentido son similares) se encuentran a lo largo de una recta (fig. 3.4). Otra situación frecuente es cuando tenemos un conjunto de notas o valores numéricos y la clase que se asigna es la media ponderada, por ejemplo $\frac{1}{3}v_1 + \frac{2}{3}v_2$. Estos son sólo dos ejemplos de un número muy amplio de problemas donde los ejemplos se encuentran agrupados en bandas, las distancias geométricas fallan y resulta más adecuado emplear otro tipo de medidas de distancia o similitud.

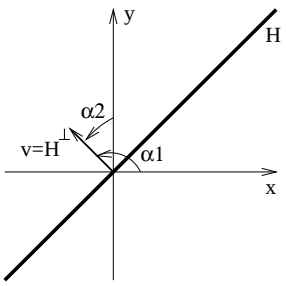


Figura 3.5: Definición de una banda en R^2.

3.6.1 Distancia basada en bandas: $d_{\alpha,ancho}(x,y)$

Las distancias usuales de la Geometría son útiles en gran número de situaciones, pero a menudo es más adecuado emplear otro tipo de distancias, tal y como hemos comentado anteriormente (por ejemplo fig. 3.4). En un trabajo previo [LC01], hemos presentado una función de distancia (aunque en realidad es sólo una medida de distancia) que agrupa los puntos de acuerdo a bandas a lo largo de un hiperplano H en R^n (una recta en R^2) $d_{\alpha,ancho}(x,y) : R^n \times R^n \rightarrow R_0^+$ como

$$d_{\alpha,ancho}(x,y) = ancho \left| \sum_{i=1}^{n} \cos\alpha_i (x_i - y_i) \right| \tag{3.16}$$

donde $\alpha = (\alpha_1, \cdots, \alpha_n) \in [0, 2\pi]^n$ es el conjunto de ángulos formados entre los ejes y el vector unitario $v = (v_1, \cdots, v_n)$ que es perpendicular al hiperplano deseado H (fig. 5.2). $ancho \in R_0^+$ controla la anchura de la banda de puntos que se encuentran a una distancia dada (valores menores implica que los puntos se acercan a H), y $|.|$ es la función valor absoluto en R. Además v cumple que $\sum_{i=1}^{n} v_i^2 = 1$, por lo tanto existen sólo n grados de libertad en la elección de los parámetros. Nótese que $(x_i - y_i)$ puede ser menor, igual o mayor que 0. Se usa R^n por simplicidad, y para dominios simbólicos $(x_i - y_i)$ representa la distancia parcial entre x_i e y_i en ese dominio.

Esta distancia cumple $d(x,y) \geq 0$, $d(x,y) = d(y,x)$, y $d(x,z) \leq d(x,y) + d(y,z)$ $\forall x,y,z \in D$; pero $d(x,y) = 0 \not\Rightarrow x = y$ $\forall x,y \in D$. Por lo tanto, esta distancia no es una métrica desde el punto de vista de las Matemáticas, sino una pseudo–métrica. Además, la relación binaria $xRy \Leftrightarrow d(x,y) = 0$ cumple las propiedades reflexiva,

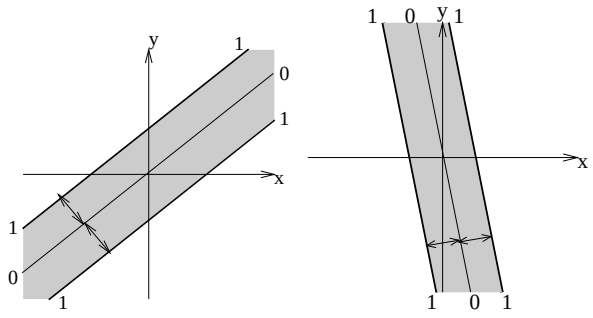


Figura 3.6: Ejemplos de bandas a lo largo de hiperplanos en R^2.

simétrica y transitiva: *R* es una relación de equivalencia que divide el conjunto original en clases. La figura 3.6 muestra un par de ejemplos de este tipo de distancias.

En lugar de restringir el vector de pesos α a valores en $[0,2\pi]^n$ de forma que $\sum_{i=1}^{n}\cos^2\alpha_i = 1$, también proponemos que, dado $w = (w_1,\cdots,w_n) \in R^n$, definir la función de distancia $d_w(x,y) : R^n \times R^n \rightarrow R_0^+$ como:

$$d_w(x,y) = \left|\sum_{i=1}^{n} w_i(x_i - y_i)\right| \tag{3.17}$$

Lema 1 *Para cualquier* $w \in R^n$*, existe un conjunto de valores* $\alpha \in [0,2\pi]^n$ *que cumple* $\sum_{i=1}^{n}\cos^2\alpha_i = 1$ *y un valor ancho* $\in R_0^+$ *tal que* $d_w(x,y) = d_{\alpha,wide}(x,y)$ $\forall x,y \in R^n$

Lema 2 *Para cualquier ancho* $\in R_0^+$ *y cualquier* $\alpha \in [0,2\pi]^n$ *que cumpla la restricción* $\sum_{i=1}^{n}\cos^2\alpha_i = 1$*, existe un conjunto de valores* $w \in R^n$ *tal que* $d_w(x,y) = d_{\alpha,wide}(x,y) \quad \forall x,y \in R^n$

Demostración Lema 1 *Dado* $w \in R^n$.

Si $w = (0,0,\ldots,0)$*, podemos elegir ancho* $= 0$ $(\in R_0^+)$*, y cualquier* $\alpha \in [0,2\pi]^n$ *que cumpla* $\sum_{i=1}^{n}\cos^2\alpha_i = 1$ *(por ejemplo* $\alpha_1 = 0$ *y* $\alpha_i = \frac{\pi}{2}$ $i = 2,\ldots,n$*:* $\alpha_1 = 0 \wedge \alpha_i = \frac{\pi}{2}$ $i = 2,\ldots,n \Rightarrow \cos\alpha_1 = 1 \wedge \cos\alpha_i = 0$ $i = 2,\ldots,n \Rightarrow \cos^2\alpha_1 = 1 \wedge \cos^2\alpha_i = 0$ $i = 2,\ldots,n \Rightarrow \sum_{i=1}^{n}\cos^2\alpha_i = 1$*). Entonces*

$d_w(x,y) = |\sum_{i=1}^{n} w_i(x_i - y_i)| = |\sum_{i=1}^{n} 0(x_i - y_i)| = |\sum_{i=1}^{n} 0| = 0$

$0 = 0\,|\sum_{i=1}^{n} \cos\alpha_i (x_i - y_i)| = d_{\alpha,ancho}(x,y)$

Si $w \neq (0,0,\cdots,0)$*, podemos elegir* $ancho = \sqrt{\sum_{j=1}^{n} w_j^2}$ *y* α *tal que* $cos\alpha_i = \frac{w_i}{\sqrt{\sum_{j=1}^{n} w_j^2}}$*. Entonces*

$\sqrt{\sum_{j=1}^{n} w_j^2} \geq 0 \Rightarrow wide \in \mathrm{R}_0^+$,

$|w_i| = \sqrt{w_i^2} \leq \sqrt{\sum_{j=1}^{n} w_j^2} \Rightarrow \frac{|w_i|}{\sqrt{\sum_{j=1}^{n} w_j^2}} \leq 1 \Leftrightarrow -1 \leq \frac{w_i}{\sqrt{\sum_{j=1}^{n} w_j^2}} \leq 1 \Leftrightarrow$

$\cos\alpha_i \in [-1,1]$ *por tanto podemos encontrar* $\alpha_i = \arccos \frac{w_i}{\sqrt{\sum_{j=1}^{n} w_j^2}} \in [0, 2\pi]$.

Además $\sum_{i=1}^{n} \cos^2 \alpha_i = \sum_{i=1}^{n} \frac{w_i^2}{\sum_{j=1}^{n} w_j^2} = \frac{\sum_{i=1}^{n} w_i^2}{\sum_{j=1}^{n} w_j^2} = 1$*, por lo que tanto ancho como* α *cumplen sus restricciones,*

$$d_{\alpha,ancho}(x,y) = wide\,|\sum_{i=1}^{n} \cos\alpha_i (x_i - y_i)| = \sqrt{\sum_{j=1}^{n} w_j^2} \left| \sum_{i=1}^{n} \frac{w_i}{\sqrt{\sum_{j=1}^{n} w_j^2}} (x_i - y_i) \right|$$

$$= \sqrt{\sum_{j=1}^{n} w_j^2} \left| \frac{1}{\sqrt{\sum_{j=1}^{n} w_j^2}} \sum_{i=1}^{n} w_i (x_i - y_i) \right| \overset{\sqrt{\sum_{j=1}^{n} w_j^2} \geq 0}{=} \left| \frac{\sqrt{\sum_{j=1}^{n} w_j^2}}{\sqrt{\sum_{j=1}^{n} w_j^2}} \sum_{i=1}^{n} w_i (x_i - y_i) \right| =$$

$|\sum_{i=1}^{n} w_i (x_i - y_i)| = d_w(x,y)$ *por lo tanto, con esos valores de ancho y* α_i *se cumple que* $d_w(x,y) = d_{\alpha,ancho}(x,y) \ \ \forall x,y \in \mathrm{R}^n$.

Demostración Lema 2 *Dado* $ancho \in \mathrm{R}_0^+$ *y* $\alpha \in [0,2\pi]^n$ *tal que* $\sum_{i=1}^{n} \cos^2 \alpha_i = 1$ *podemos elegir* $w_i = ancho\ \cos\alpha_i$ *(*$\in \mathrm{R}$*). Entonces*

$d_w(x,y) = |\sum_{i=1}^{n} w_i (x_i - y_i)| = |\sum_{i=1}^{n} ancho\ \cos\alpha_i (x_i - y_i)| =$

$|ancho \sum_{i=1}^{n} \cos\alpha_i (x_i - y_i)| \overset{ancho \in \mathrm{R}_0^+}{=} ancho\,|\sum_{i=1}^{n} \cos\alpha_i (x_i - y_i)| = d_{\alpha,ancho}(x,y)$

por lo tanto, con esos valores de w_i *se cumple que* $d_w(x,y) = d_{\alpha,wide}(x,y) \ \ \forall x,y \in \mathrm{R}^n$.

El significado de estos lemas debe interpretarse en el sentido de que dados *n* valores reales, siempre existe un conjunto de *n* ángulos con cosenos son proporcionales a esos valores reales (y viceversa).

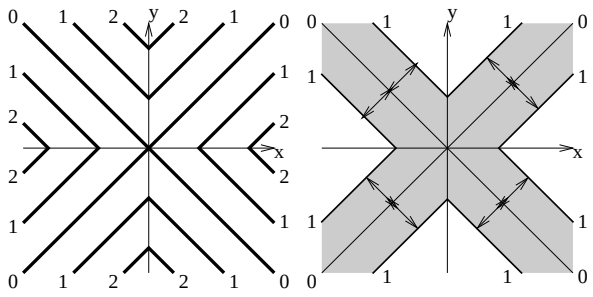


Figura 3.7: Ejemplo de distancia $d_w^{|\times|}(x,y)$.

3.6.2 Distancias $d_w^{\times}(x,y)$ y $d_w^{|\times|}(x,y)$

Dado $w = (w_1, \cdots, w_n) \in \mathrm{R}^n$, también son interesantes las dos distancias que se definen a continuación:

$$d_w^{\times}(x,y) = \sum_{i=1}^{n} w_i \, |x_i - y_i| \tag{3.18}$$

$$d_w^{|\times|}(x,y) = \left| \sum_{i=1}^{n} w_i \, |x_i - y_i| \right| \tag{3.19}$$

Si los valores $w_i > 0 \quad \forall i = 1, \ldots, n$ entonces las dos distancias son un tipo de distancia de la Suma ponderada, pero si algún $w_i \leq 0$ entonces se obtienen distancias no usuales. Por ejemplo, si consideramos $w_1 = -1$ y $w_2 = 1$ en R^2 se obtiene la distancia $d_w^{|\times|}(x,y)$ que aparece en la figura 3.7. A la izquierda se muestra la distancia a la que se encuentran algunos puntos del (0,0), y a la derecha los puntos con valores $d_w^{|\times|}(x,y) \in [0,1]$.

Si consideramos los mismos valores $w_1 = -1$ y $w_2 = 1$ se obtiene la distancia $d_w^{\times}(x,y)$ que se muestra en la figura 3.8.

$d_w^{|\times|}(x,y)$ está restringida a valores positivos o cero, pero $d_w^{\times}(x,y)$ puede también devolver valores negativos. En este punto es interesante observar que $d_w^{\times}(x,y)$ puede separar el espacio en dos regiones siguiendo una plantilla en forma de X, de acuerdo a los puntos con valor de distancia positivo o negativo (fig. 3.9).

Con estas distancias se puede expresar por ejemplo el conjunto de movimientos legales de un álfil en el juego de ajedrez. Se asigna distancia 0 a todas las casillas a

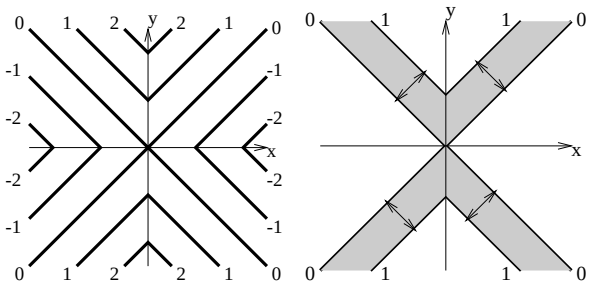


Figura 3.8: Ejemplo de distancia $d_w^{\times}(x,y)$.

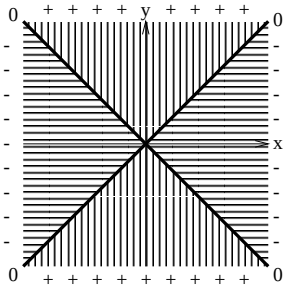


Figura 3.9: Regiones positivas y negativas de $d_w^{\times}(x,y)$.

las que el álfil podría mover en la próxima jugada, y en ese sentido, podría definirse el conjunto de casillas similares a la que ocupa el álfil.

3.7 Conclusiones del Capítulo

El capítulo comienza presentando qué se entiende por problema de clasificación y fijando la notación que se usará posteriormente. En los problemas de clasificación se parte de un conjunto finito de ejemplos cada uno con un conjunto de observaciones de algunas características relevantes y la clasificación correcta. El objetivo es relacionar las observaciones y las clases, y así poder determinar, a partir de los valores de los atributos, la clase a la que pertenece cualquier objeto dado.

A continuación se muestra el funcionamiento de algunos métodos que realizan aprendizaje para tareas de clasificación. Esta lista de métodos no debe considerarse, ni mucho menos, exhaustiva, sino que símplemente se muestra los principios en que se bansan algunos métodos a título ilustrativo. Debe tenerse en cuenta que la inclusión de estos métodos y no otros, puede considerarse en última instancia casi arbitraria, ya que algunos de los métodos mencionados podrían no estar y podrían haberse incluidos otros que no se encuentran presentes. Se describe someramente el funcionamiento básico de los árboles de clasificación (clásicos y difusos), uso de reglas, la transformación de árboles de decisión en reglas, el aprendizaje de reglas mediante algoritmos genéticos, y la teoría del ejemplar generalizado anidado (NGE).

Se muestra cómo debe ser un sistema de Razonamiento Basado en Casos que realice tareas de clasificación, y surge rápidamente la necesidad de definir similitud entre ejemplos.

Se muestran las dificultades para definir funciones de similitud. Se estudia la relación entre el concepto de similitud y el de distancia, y cómo se pueden definir medidas de similitud a partir de medidas de distancia. Es bastante interesante el comportamiento suave de este tipo de funciones debido a su salida gradual.

Se analiza cómo debe ser una función de distancia en general y cómo para definir similitud no es necesario que la función de distancia cumpla todas las propiedades

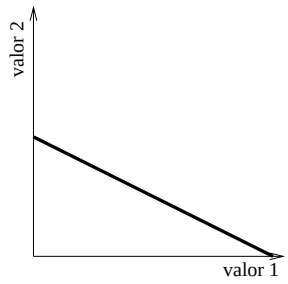


Figura 3.10: Puntos con la misma media ponderada.

que exige la Geometría. Por tanto, para definir similitud entre objetos, las medidas de distancia deben entenderse en sentido amplio, sin limitarlas obligándolas a cumplir propiedades que no son necesarias para nuestros propósitos.

Se describen rápidamente las funciones de distancia típicas de la Geometría y algunas variantes que vamos a emplear en los experimentos. A continuación se introducen algunas medidas de distancia nuevas que no son funciones de distancia desde el punto de vista de la Geometría: $d_{\alpha,ancho}(x,y)$, $d_w^{\times}(x,y)$ y $d_w^{|\times|}(x,y)$.

La medida de distancia $d_{\alpha,ancho}(x,y)$ asigna valores de distancia (y de similitud) iguales a lo largo de hiperplanos en R^n. Es preferible en dominios donde los ejemplos similares están agrupados formando bandas, por ejemplo cuando tenemos un conjunto de notas o valores numéricos y la clase se asigna de acuerdo con la media ponderada de esos valores (por ejemplo la figura 3.10 muestra puntos donde la clase se asigna como $\frac{1}{3}v_1 + \frac{2}{3}v_2$).

Las medidas $d_w^{\times}(x,y)$ y $d_w^{|\times|}(x,y)$ asignan la distancia siguiendo un patrón en forma de X. Con estas distancias se puede expresar, por ejemplo, el conjunto de movimientos legales de un álfil en el juego del ajedrez, porque permiten asignar distancia 0 a todas las casillas donde el álfil puede mover en la próxima jugada.

Podemos concluir que las medidas de distancia propuestas $d_{\alpha,wide}(x,y)$, $d_w^{\times}(x,y)$, y $d_w^{|\times|}(x,y)$, son útiles en dominios donde la información está agrupada de acuerdo a patrones diferentes de la noción habitual de proximidad. En general esto mismo ocurre con las medidas de distancia no usuales, incluso aunque no sean verdaderas distancias desde un punto de vista matemático. En ese tipo de dominios, las distancias típicas de la geometría no se comportan de manera adecuada y otras métodos

como por ejemplo los k–NN elegirán puntos que no son los más indicados. Todavía es necesario realizar mucho trabajo con este tipo de medidas de distancia, como por ejemplo la estimación de los valores de los parámetros o la definición de nuevas distancias no usuales.

Capítulo 4

Métodos de Clasificación Basados en Distancias

En el capítulo 3 se vieron distintas formas de definir la distancia que hay entre dos casos dados. También vimos cómo la medida de distancia nos define implicitamente similitud. Así podíamos decir que, dada la medida de distancia d, el caso más similar a otro dado es aquel que se encuentra a menor distancia.

En este capítulo vamos a ver métodos basados en medidas de distancia que sirven para clasificar. En el más sencillo se va a recuperar sólo un caso, pero en los más complejos se va a recuperar más de uno y después se va a combinar la información de esos casos para proponer una clasificación. Por tanto el comportamiento general de RBC que se usa en este capítulo queda reflejado en el siguiente algoritmo:

1. Se plantea un caso e que debe ser clasificado.

2. Dada una función de distancia d se selecciona de la Base de Casos BC un conjunto de casos $K = \{(e_1, c_1), (e_2, c_2), \cdots, (e_k, c_k)\}$ tal que $e_1, e_2, \cdots, e_k$ son los casos "más similares" a e. Donde k puede ser un número preestablecido a priori, variable o dependiente de alguna función.

3. Se combina las clasificaciones de los distintos casos recuperados para obtener una clasificación *c* para *e*.

Todo el peso de la noción de similitud recae sobre la función de distancia que elegimos, y por tanto esta elección va a resultar de vital importancia para que el Razonador Basado en Casos realice bien las tareas de clasificación. Además se debe determinar de alguna forma cuántos ejemplos se deben recuperar, y cómo combinar la información que aportan esos ejemplos para asignar una clase al caso nuevo.

4.1 Método del Vecino más Cercano (NN ó 1-NN)

Este método de clasificación consiste en la aplicación directa del Principio del Vecino más Cercano. Se conoce bajo las denominaciones de Vecino más Cercano (en inglés *Nearest Neighbour*), NN ó 1-NN [FH51] [CH67] [Das91]. Cuando intentamos clasificar un caso nuevo *e*, directamente le asignamos la clase del caso (o punto) que se encuentra a una menor distancia de acuerdo con una función o medida de distancia. Normalmente esta medida de distancia es la distancia Euclídea, pero puede ser cualquier función que proporcione una medida de disimilitud (valores bajos indican que los casos son similares) [Rit92] [Rit95]. De manera ideal esta función debería proporcionar una medida de distancia entre clases, pero en la práctica no conocemos la clase del caso nuevo (eso es exáctamente lo que queremos conocer), y usamos en cierto modo un principio de localidad: si los atributos de dos casos son similares, entonces probablemente sus clases también serán similares. Pero más allá de consideraciones teóricas, el hecho es que sabemos experimentalmente que este método funciona y realiza clasificaciones correctas.

El método del Vecino más Cercano suele tener algunos problemas. En primer lugar, si pudieramos encontrar la función óptima de distancia entonces obtendríamos un clasificador óptimo, pero encontrar esa función es equivalente al problema original, que es no trivial o imposible. Sin embargo, se han realizado algunos intentos para encontrar una medida de distancia óptima bajo algunas suposiciones [HT96] [LW97] [GB97]. En segundo lugar, una vez que aceptamos que en general no podemos conocer la medida óptima de distancia, debemos asumir que la clase del vecino

más cercano a menudo no será la clase correcta, es decir, dependiendo de la base de casos, algunos puntos violan el principio de localidad en que se basa este método: 1–NN tiene bastantes problemas para clasificar los puntos aislados y los datos con ruido.

En la literatura científica se han estudiado algunas variantes del método 1–NN, incluyendo las series IBx [AKA91] [Aha92] para reducir las necesidades de almacenamiento e incrementar la tolerancia al ruido; la teoría del "Ejemplar Generalizado Anidado" (*Nested Generalized Exemplar* o símplemente NGE)) [Sal91] donde se emplean hiperrectángulos en lugar de puntos; la "Métrica de Diferencia de Valores" (*Value Difference Metric* o VDM) [SW86], la "Métrica de Diferencia de Valores Modificada" (*Modified Value Difference Metric* o MVDM) [CS93], la "Métrica de Diferencia de Valores Heterogénea" (*Heterogeneous Value Difference Metric* o HVDM) [WM97] y la "Métrica de Diferencia de Valores Simplificada" (*Simplified Value Difference Metric* o SVDM) [Dom97] que derivan estadísticamente la medida de distancia para atributos simbólicos basándose en la similitud total de la clasificación de todos los ejemplos para cada valor posible de cada atributo.

Otras medidas de distancia interesantes son la *Asymmetric Anisotropic Similarity Metric* (AASM) [RA95] y la *Local Asymmetrically Weighted Similarity Metric* (LASM) [RA99] que definen medidas de distancia locales que varían a lo largo del espacio y son asimétricas.

4.2 Método de los *k* Vecinos más Cercanos (*k*-NN)

Los métodos de los *k* Vecinos más Cercanos (*k Nearest Neighbors o k–NN*) realizan la clasificación teniendo en cuenta los *k* puntos más cercanos al caso nuevo, en lugar de usar solo un punto [Wet94] [WD95]. Aquí el principio de localidad puede ser más flexible en el sentido de que si la clase del vecino más cercano no es la correcta, quizás sí lo sea la clase de algunos de los otros vecinos más cercanos y podamos realizar la clasificación correcta. Por eso se presupone que la elección de la medida de distancia es menos crucial en *k*–NN. Pero ahora aparecen dos cuestiones nuevas: la elección del valor adecuado de *k* (¿cuántos casos se deben tener en cuenta?), y

la combinación de la información de esos *k* vecinos más cercanos. Con respecto a la combinación de la información de los *k* vecinos más cercanos, la forma más directa consiste en asignar la clase de la mayoría, es decir, la moda de entre las clases de los *k*–NN. Se pueden realizar muchas combinaciones diferentes, pero la evidencia experimental recomienda usar el Voto Ponderado de los *k*–NN o *Weighted Vote k–NN* [Wet94] (nosotros también hemos verificado este hecho en experimentos preliminares): considerar la distancia de los *k* puntos, y ponderar la importancia o peso de cada caso inversamente por la distancia que lo separa del caso nuevo. Por eso he tomado como referencia en todas las comparaciones el Voto Ponderado de los *k*–NN. De aquí en adelante, y mientras no se indique lo contrario, cuando use simplemente el término *k*–NN me referiré a esta variante.

El valor óptimo de *k* depende de la base de casos, la medida de distancia y la forma en que se combina la información de los *k* vecinos. *k* se puede estimar por Validación Cruzada [WK91] [Wet94] [WD95]. También es posible, aunque poco frecuente, calcular un valor local de *k* para cada punto nuevo, en lugar de usar un único valor de *k* para todo el espacio [Wet94] [WD94].

k–NN puede tener problemas relacionados con la elección del valor de *k*. Si *k* es demasiado grande o el caso nuevo se encuentra en una región poco poblada, *k*–NN tendrá en cuenta puntos lejanos que probablemente no son muy relevantes. Si *k* es demasiado pequeño o el caso nuevo se encuentra en una región densamente poblada, *k*–NN ignorará algunos puntos cercanos que probablemente aporten información importante. Usando el Voto Ponderado de los *k*–NN se pueden reducir las consecuencias negativas de una mala elección de *k*, pero todavía puede haber problemas si el valor de *k* varía a lo largo del espacio.

En la literatura se han estudiado diferentes variantes de los *k*–NN, incluyendo los *k surrounding neighbors* o *k*–SN [ZYY97] que seleccionan *k* ejemplos cercanos al punto nuevo y que además se encuentran "bien distribuidos" alrededor de este punto. También se han realizado esfuerzos para reducir el número de características que se deben considerar y tomar en cuenta solo las relevantes (*feature selection*), principalmente mediante reducción global de atributos (*global dimension reduction* [HT96] [BL97] [AD91], o incluso mediante una selección local de características [Dom97].

Otros artículos fusionan los *k*–NN y las redes neuronales [BV92], o se centran en la noción de similitud en lugar de usar distancias, como Plaza et al. [PLA96] que usa una asignación basada en entropía muy interesante, pero existe una clara relación entre la noción de similitud y la distancia [Rit92] [Rit95] [LC01]. Otra idea interesante es el *Racing Algorithm* [MM97] que descarta rápidamente los modelos menos prometedores y también se puede utilizar para encontrar características relevantes.

4.3 Las Variantes Propuestas de *k*–NN

Se proponen algunas variantes que difieren de *k*–NN [Wet94] [WD95] en tres características básicas:

C1: Imponer un umbral de distancia ε en el conjunto K de los *k* vecinos más cercanos.

C2: Siempre que K esté vacío, usar *k*–NN.

C3: Dado de antemano un conjunto de medidas de distancia, seleccionar la medida de distancia de ese conjunto que mejores resultados proporciona usando el método 1–NN con el mismo conjunto de datos.

Ahora podemos manejar estas tres características de manera independiente para realizar comparaciones y estudiar al utilidad de cada una de ellas. Combinando C1, C2 y C3 obtenemos las 6 primeras variantes que se muestran en la tabla 4.1. C2′ denota que siempre que K esté vacío, se usa 1–NN en lugar de *k*–NN. Como veremos más adelante, es útil hacer esta distinción especial. Pero procedamos ahora a presentar más formalmente cada una de las características.

4.3.1 Característica C1: ε–entornos

Para superar las dificultades de los clasificadores *k*–NN se propone el punto de vista opuesto: para clasificar un caso nuevo *e*, se debe establecer un entorno alrededor de

Tabla 4.1: Variantes propuestas sobre el método básico de los *k*–NN.

Código	Características	Método de clasificación
A		*k*–NN
B	C1	ε–entorno
C	C3	*k*–NN Heur
D	C1 & C2	ε–entorno^{k-NN}
E	C1 & C3	ε–entorno Heur
F	C1 & C2 & C3	ε–entorno^{k-NN} Heur
D1	C1 & C2′	ε–entorno^{1-NN}
F1	C1 & C2′ & C3	ε–entorno^{1-NN} Heur

ese punto y tener en cuenta los puntos que están dentro de ese entorno, en lugar de seleccionar un número fijo de puntos más cercanos. De este modo siempre se tiene garantizado considerar solo puntos cercanos, independientemente de la densidad de puntos de la base de casos: cuando el punto nuevo e se encuentra en un área poco poblada del espacio se considerarán menos puntos que si se encuentra en una región densamente poblada.

Se elige un valor real fijo ε como el radio del entorno, así controlamos su tamaño. Se define el ε–*entorno* de un caso e como

$$K = \{e\prime \in S \text{ tal que } d(e\prime, e) \leq \varepsilon\} \tag{4.1}$$

y el ε–*entorno de la clase* C_i de un caso e como

$$K_i = \{e\prime \in S \text{ tal que } clase(e\prime) = C_i \text{ y } d(e\prime, e) \leq \varepsilon\} \tag{4.2}$$

donde S es el conjunto de datos. En la medida de influencia del ε–entorno de una clase es importante considerar la distancia y números de casos del ε–entorno que pertenecen a esa clase. Se definen las *medidas de influencia* o *cardinales exponenciales* de un ε–entorno K y K_i respectivamente como:

$$|K|_d = \sum_{e\prime \in K} \exp(-\alpha d(e, e\prime)) \qquad \text{donde } \alpha = \frac{4}{\varepsilon^2} \tag{4.3}$$

$$|K_i|_d = \sum_{e\prime \in K_i} \exp(-\alpha d(e, e\prime)) \qquad \text{donde } \alpha = \frac{4}{\varepsilon^2} \tag{4.4}$$

Dadas estas definiciones, se verifica claramente que

$$\sum_{i=1}^{n} |K_i|_d = |K|_d \tag{4.5}$$

donde n es el número de clases o clasificaciones posibles.

Para clasificar un caso nuevo e se calcula la medida de influencia del ε–entorno de cada clase y se asigna a e la clase con mayor medida de influencia.

En trabajos previos hemos observado que es deseable el comportamiento suave de la función exponencial desde puntos cercanos (y probablemente relevantes) hasta los más alejados (y probablemente menos importantes) Este comportamiento suave también ha sido estudiado y empleado en otras áreas, como Psicología Cognitiva, Estadística o Redes Neuronales.

Dadas las definiciones anteriores podemos decir que un ejemplo e será un *punto aislado* si $K = e$, o equivalentemente si $|K|_d = 1$; y e será un *punto interno* si $|K_i|_d = |K|_d$, donde $clase(e) = C_i$, o equivalentemente si $|K_j|_d = 0 \; \forall j \neq i$. Claramente, con estas definiciones todos los puntos aislados son internos.

También se puede definir que un ejemplo e será un *punto de l–ruido* si $|K_i|_d = 1$, siendo $clase(e) = C_i$, y $|K|_d \geq l$. Es decir, en el entorno K, e es el único ejemplo que hay de su clase, y existen al menos $l-1$ puntos de clase distinta a la de e.

4.3.2 Características C2 y C2′: ε–entornos^{k-NN} y ε–entornos^{1-NN}

Si el valor de ε es demasiado pequeño, puede ocurrir que K esté vacío, y por tanto el caso nuevo será “no–clasificado”. Una manera de solucionar este problema consiste en elegir la clase del vecino o los vecinos más cercanos siempre que K esté vacío, es decir, cuando el ε–entorno “falla” entonces debemos clasificar usando k–NN, que siempre proporciona vecinos más cercanos y una clasificación. También se ha estudiado usar 1–NN cuando K está vacío.

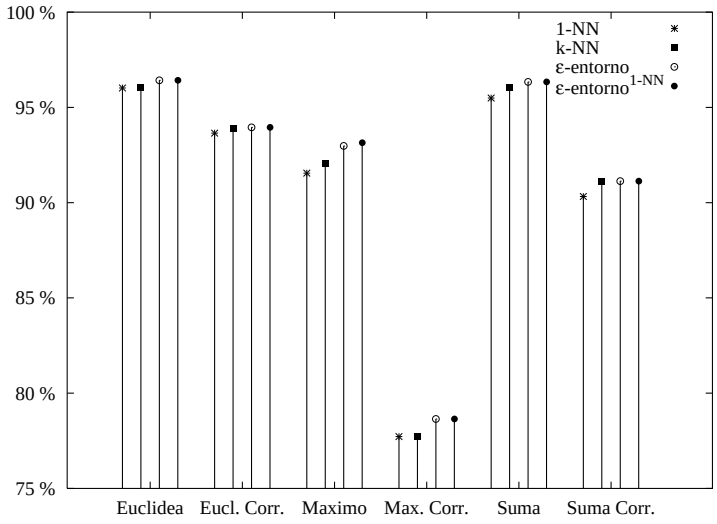


Figura 4.1: Acierto de los métodos 1–NN, *k*–NN, ε–entorno y ε–entorno^{1-NN} con cada medida de distancia en la base de casos Letter Recognition. La relación entre el acierto de 1–NN y el acierto de los otros métodos es clara.

4.3.3 Característica C3: Heurística para seleccionar la medida de distancia

Como era esperable, si hay características irrelevantes (LED24, Waveform–40, y Bandas), el mejor comportamiento se obtiene usando ponderación de atributos (*feature weighting*) que tenga en cuenta información sobre la importancia o utilidad de los atributos (por ejemplo mediante la correlación con la clase). En general se pueden usar otros métodos como la Información Mutua (*Mutual Information* [Wet94] [WD95], o ponderar la ganancia de información basándose en la entropía [DvdBW97]. Wettschereck [WAM97] ha realizado una revisión de métodos de ponderación de atributos bastante completa y útil. También se han probado técnicas de ponderación local [HT96] [AMS97a] [AMS97b], [Fri94], incluyendo algoritmos de ponderación de clases [HC97]

Se aprecia que existe cierta relación entre el acierto que obtiene 1–NN con diferentes distancias y el acierto de otros métodos de clasificación basados en esas distancias (fig. 4.1). Esta idea también es apoyada por algunas bases de casos, donde

una distancia obtiene resultados extremadamente pobres (en comparación con las otras distancias) y lo mismo le ocurre al resto de métodos que usan esas mismas distancias. Este hecho nos llevó a pensar que el método del vecino más cercano (1-NN) junto con distintas medidas de distancia podría ser utilizado para extraer información útil sobre las bases de casos y la distribución de puntos a lo largo del espacio. Por estos motivos, y considerando que los métodos 1–NN son relativamente rápidos, se propone la siguiente heurística: *entrena y prueba métodos 1–NN con un conjunto de medidas de distancia y utiliza el método basado en distancias deseado (k–NN,* ε*–entorno,* ε*–entorno*$^{k-NN}$*...) con la distancia que ofrece mayor porcentaje de acierto con 1–NN.*

4.4 Los Experimentos

Se ha usado un gran número de bases de casos para estudiar el comportamiento de los diferentes clasificadores, y se han realizado las pruebas usando 10 Validación Cruzada (10–CV) [WK91]. Se han probado todos los clasificadores con exactamente los mismos ejemplos y se ha realizado una prueba *t*–Student pareada con dos colas con un nivel de significación del 95% para comparar los resultados de los distintos clasificadores.

4.4.1 Los Clasificadores

Para estudiar la importancia de la función de distancia en este tipo de métodos se consideran seis medidas de distancia diferentes: las tres distancias básicas de la Geometría (Euclídea, Máximo y Suma) y sus variantes con correlación, donde la importancia de cada atributo se pondera por la correlación de ese atributo con la clase.

Los clasificadores (Voto Ponderado) *k*–NN, ε–entorno y ε–entorno^{k-NN} tienen uno o dos parámetros que controlan su comportamiento. Para estimar el valor de los parámetros se ha realizado una prueba 10–CV con cada conjunto de entrenamiento. En los clasificadores *k*–NN se han considerado los valores impares 1, 3, 5,..., 49

para el parámetro k; en los métodos ε–entorno y ε–entorno^{k-NN} se han considerado 30 valores en el intervalo [0,2] para el parámetro ε. En los métodos ε–entorno^{k-NN} se han considerado los valores impares 1, 3, 5,..., 19 para su parámetro k[1]; y se ha distinguido el caso especial ε–entorno^{1-NN} donde $k = 1$.

En resumen, se ha probado un gran número de combinaciones y se ha empleado una gran cantidad de tiempo para obtener estos resultados.

4.4.2 Las Bases de Casos

Se han empleado en total las 68 bases de casos que se presentaron en la sección 1.2.1: 18 bases bastante utilizadas del UCI–Repository [BM98], una versión reducida de 1,000 ejemplos de la Granada Handwritten Digits (tabla 1.1), y 49 bases sintéticas. Las bases del UCI–Repository son usadas con frecuencia en la literatura científica, lo que facilita las comparaciones con los resultados experimentales obtenidos por otros clasificadores introducidos en otros artículos, y las bases sintéticas son útiles para estudiar los clasificadores en un entorno controlado.

Las bases de casos sintéticas se han construido *ad hoc* sobre el cuadrado unidad [0,1]×[0,1], cada una con 500 ejemplos. Como queremos estudiar la influencia de la distribución de clases de la base de casos, hemos considerado las bases: Bandas (5, 10 y 20), Gauss, Anillos con área constante (3, 6 y 9), Anillos con radio constante (3, 6 y 9), Senos (3, 6 y 9) y Cuadrados (2, 4, 6 y 8). En la figura 4.2 se muestran algunos ejemplos de estas bases, y en el apartado 1.2.1 se comenta con más detalle cada una de ellas.

Parece razonable que k–NN pueda tener dificultades si el valor óptimo de k no es constante a lo largo del espacio, es decir, si en algunas regiones k debe ser mayor que en otras. En esas condiciones es razonable que los métodos ε–entorno y ε–entorno^{k-NN} exhiban un comportamiento mejor porque en algunas regiones k–NN tomará en cuenta demasiados puntos o demasiado pocos, pero los ε–entornos considerarán solo los puntos relevantes (aquellos que están suficientemente cerca, pero no los alejados).

[1] No se debe confundir con el parámetro k de los k–NN. El sentido de k en los métodos k–NN y ε–entorno^{k-NN} es similar, pero los métodos son muy diferentes.

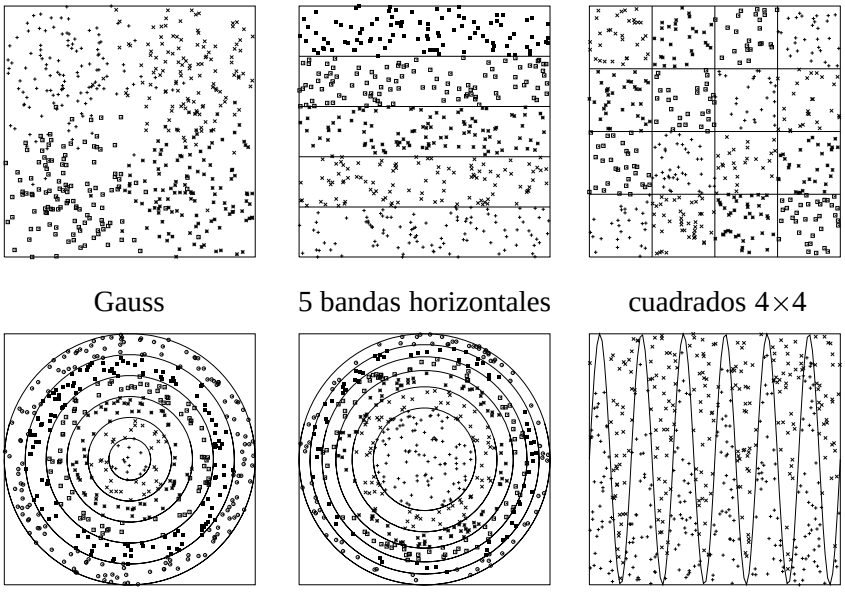
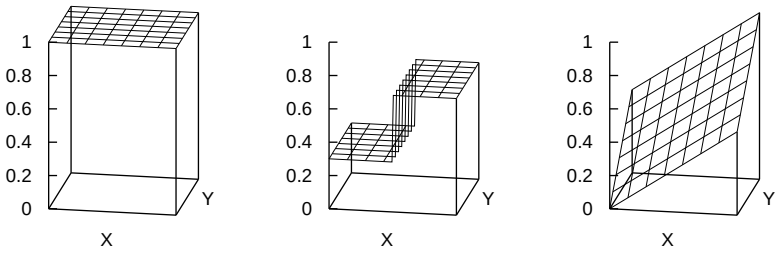


Figura 4.2: Algunas bases de casos sintéticas. Símbolos diferentes indican clases diferentes. Las líneas representan las fronteras de decisión.

Se han generado tres variantes de todas las bases de datos sintéticas (excepto por supuesto para la base Gauss) para estudiar cómo se ven afectados los métodos por el hecho de que la densidad de puntos varíe. Así podemos comprobar si los métodos *k*–NN se encuentran con problemas cuando el valor óptimo de *k* varía a lo largo del espacio. Las regiones densamente pobladas tienen más puntos y el valor óptimo de *k* será mayor. En estas tres variantes, los puntos se distribuyen con probabilidades diferentes a lo largo del espacio (fig. 4.3). Así se consigue un espacio con densidad de puntos diferente y por lo tanto un valor óptimo de *k* que varía. Estas variantes son:

- *Uniforme*: los puntos se distribuyen uniformemente a lo largo del espacio.
- *Mitad*: el 30% de los puntos se encuentran en la mitad izquierda del espacio ($x < 0.5$) y el restante 70% en la mitad derecha ($x \geq 0.5$): los puntos están distribuidos en dos regiones claramente diferenciadas.
- *Progresiva*: la probabilidad de aceptación de un punto es proporcional a la



a. Uniforme b. Mitad c. Progresiva

Figura 4.3: Distribución de puntos en las tres variantes de las bases de casos sintéticas.

suma de sus coordenadas ($x+y$): la densidad de puntos se incrementa progresivamente desde la esquina inferior izquierda hasta la superior derecha.

4.5 Análisis de los Resultados

Se han usado varias formas de comparar el comportamiento de los algoritmos para evitar que un punto de vista particular pudiera llevar a conclusiones erróneas. Por una parte, se ha calculado para cada clasificador su acierto medio y la mejora del acierto respecto del método *k*–NN básico a lo largo de todas las bases de casos (tabla 4.2). Para facilitar la comparación entre clasificadores, también se ha incluido una columna con la posición relativa de cada clasificador.

En la tabla 4.3 se muestra para cada clasificador su acierto medio desglosado en cada una de las tres categorías o grupos de bases de casos que estamos distinguiendo en este trabajo. Para facilitar la comparación entre clasificadores, también se ha incluido una columna con la posición relativa de cada clasificador en cada una de estas categorías.

Por otra parte, se ha aplicado una comparación pareada entre clasificadores usando un test *t*–Student con dos colas para construir un intervalo con un 95% de confianza para la diferencia en los porcentajes de acierto de los algoritmos en todas las bases de casos (tabla 4.4), en las bases de casos del UCI–Repository (tabla 4.5) y en las bases de casos sintéticas (tabla 4.6). En adelante, cuando use el término di-

Tabla 4.2: Acierto de los clasificadores en las pruebas con todas las bases de casos: acierto medio y mejora frente a *k*–NN. "Pos." indica la posición relativa de cada clasificador. F logra la mejor posición.

	Clasificador	Acierto Medio	Δ vs *k*–NN	Pos.
A	*k*–NN	85.65%	0.00%	8
B	ε–ball	85.75%	+0.10%	7
C	*k*–NN Heur	87.30%	+1.65%	3
D	ε–ball^{k-NN}	85.88%	+0.23%	5
E	ε–ball Heur	87.29%	+1.64%	4
F	ε–ball^{k-NN} Heur	87.39%	+1.74%	1
D1	ε–ball^{1-NN}	85.81%	+0.16%	6
F1	ε–ball^{1-NN} Heur	87.37%	+1.72%	2

ferencia estadísticamente significativa, o simplemente diferencia significativa, nos referiremos a que la diferencia es estadisticamente significativa de acuerdo a este test *t*–Student.

Estas tablas muestran que las diferentes formas de comparar el comportamiento de los algoritmos proporcionan resultados muy parecidos, y todas las variantes propuestas de *k*–NN mejoran al *k*–NN básico frecuentemente.

La conclusión más destacable es la utilidad de la característica C3: la heurística propuesta mejora el acierto en muchas bases de casos, y los mejores clasificadores son aquellos que la usan. Las tablas 4.2 y 4.4 muestran la gran mejora que se logra con C3 (compárese A frente a C, B frente a E, y D frente a F). Sin embargo, C1 y C2 proporcionan una mejora menor. C mejora a A, B y D; por lo tanto ¡es preferible usar C3 antes que C1 y C2 juntos!

Debe tenerse en cuenta que la característica C2 incluye a la C1. Añadir C2 (y C1) proporciona una mejora mayor que añadir sólo C1 (compárese B frente a D, y E frente a F), pero la influencia de C1 y C2 es similar.

Desde un punto de vista estadístico no hay diferencia significativa entre D y D1, y entre F y F1, pero los métodos que usan C2 mejoran sólo ligeramente a los méto-

Tabla 4.3: Acierto de los clasificadores en las pruebas con las bases de casos del UCI–Repository, Sintéticas y todas las bases de casos. "Pos." indica la posición relativa de cada clasificador en cada apartado.

		UCI		Sintéticas		Todas	
	Clasificador	Media	Pos.	Media	Pos.	Media	Pos.
A	*k*–NN	85.03%	7	85.89%	8	85.65%	8
B	ε–ball	84.95%	8	86.06%	7	85.75%	7
C	*k*–NN Heur	85.12%	6	88.14%	1	87.30%	3
D	ε–ball^{k-NN}	85.33%	3	86.09%	5	85.88%	5
E	ε–ball Heur	85.14%	5	88.11%	4	87.29%	4
F	ε–ball^{k-NN} Heur	85.48%	1	88.13%	2	87.39%	1
D1	ε–ball^{1-NN}	85.15%	4	86.07%	6	85.81%	6
F1	ε–ball^{1-NN} Heur	85.41%	2	88.13%	2	87.37%	2

dos que usan C2′. Sin embargo, hay una diferencia grande entre el esfuerzo computacional requerido por C2 y el requerido por C2′, porque C2 implica usar *k*–NN cuando el ε–entorno está vacío, es decir, hay que estimar el valor del parámetro *k*, mientras que C2′ usa directamente 1–NN. Por tanto, es recomendable usar C2′ en lugar de C2: se debe usar F1 en lugar de F y D1 en lugar de D.

E, F y F1 mejoran significantivamente al resto de clasificadores frecuentemente. No hay diferencias significativas entre ellos, pero F tiende a mejorar ligeramente a F1, y F1 mejora ligeramente a E. Sin embargo, F requiere mucho más esfuerzo computacional que F1 y E. Por lo tanto, en general es recomendable usar el método F1 (ε–entorno^{1-NN}Heur), porque ofrece casi los mismos resultados que F (ε–entorno^{k-NN}Heur) pero con muchos menos requerimientos de cálculo. F1 mejora significativamente al *k*–NN básico frecuentemente (15–53–0), mejora al *k*–NN básico muy frecuentemente (45–5–18), y proporciona una mejora global del 2.18% sobre *k*–NN. Esto también es cierto con las bases de datos del UCI–Repository, donde el método F1 logra respectivamente 3–16–0, 11–3–5 y una mejora global de 0.47% sobre *k*–NN.

En resumen, todas las características propuestas son útiles. Claramente, la ca-

Tabla 4.4: Comparación pareada de diferencias estadísticamente significativas entre clasificadores. Cada celda contiene respectivamente el número de victorias, empates y derrotas estadisticamente significativas entre el método de esa fila y el método de esa columna. E, F y F1 mejoran significativamente al resto de métodos frecuentemente.

	F1	D1	F	E	D	C	B
A	0–53–15	0–63–5	0–53–15	0–53–15	0–63–5	1–53–14	1–62–5
B	2–55–11	0–67–1	2–55–11	2–55–11	0–67–1	6–48–14	
C	3–58–7	14–47–7	3–58–7	3–58–7	14–47–7		
D	2–55–11	0–68–0	2–55–11	2–55–11			
E	0–68–0	11–55–2	0–68–0				
F	0–68–0	11–55–2					
D1	2–55–11						

racterística que más contribuye a mejorar los resultados es C3, seguida por C1 y C2, y, sin ninguna información previa, es recomendable usar el método ε–entorno^{1-NN} Heur (F1).

Las tablas 4.7[2] y 4.8 muestran los resultados que han obtenido los clasificadores con cada base de casos. Los resultados aparecen seguidos por un "$+$" (o un "$-$") si muestran un mejora (o degradación) estadísticamente significativa sobre el *k*–NN básico, de acuerdo con un test pareado *t*–Student con dos colas con un nivel de confianza del 95%.

Este trabajo se centra principalmente en la comparación de estos métodos para establecer sus puntos fuertes y débiles y hacer recomendaciones sobre su utilidad, y no en sus resultados concretos o si son mejores o peores que otras aproximaciones. Somos conscientes del hecho de que la comparación con otros métodos de clasifi-

[2] A pesar del hecho de que los resultados con la base de casos Cleveland pudieran parecer anormalmente bajos, se debe tener en cuenta que estos resultados se han obtenido considerando 5 clases, mientras que los experimentos con esta base normalmente se han concentrado en distinguir simplemente la presencia (valores 1,2,3,4) de la ausencia (valor 0) de enfermedad de corazón.

Tabla 4.5: Comparación pareada de diferencias estadísticamente significativas entre clasificadores en las bases de casos del UCI–Repository. Cada celda contiene respectivamente el número de victorias, empates y derrotas estadisticamente significativas entre el método de esa fila y el método de esa columna.

	F1	D1	F	E	D	C	B
A	0–16–3	0–16–3	0–16–3	0–16–3	0–16–3	1–17–1	1–15–3
B	0–19–0	0–18–1	0–19–0	0–19–0	0–18–1	3–15–1	
C	1–14–4	1–14–4	1–14–4	1–14–4	1–14–4		
D	0–19–0	0–19–0	0–19–0	0–19–0			
E	0–19–0	0–19–0	0–19–0				
F	0–19–0	0–19–0					
D1	0–19–0						

Tabla 4.6: Comparación pareada de diferencias estadísticamente significativas entre clasificadores en las bases de casos sintéticas. Cada celda contiene respectivamente el número de victorias, empates y derrotas estadisticamente significativas entre el método de esa fila y el método de esa columna. E, F y F1 mejoran significativamente al resto de métodos frecuentemente.

	F1	D1	F	E	D	C	B
A	0–37–12	0–47–2	0–37–12	0–37–12	0–47–2	0–36–13	0–47–2
B	2–36–11	0–49–0	2–36–11	2–36–11	0–49–0	3–33–13	
C	2–44–3	13–33–3	2–44–3	2–44–3	13–33–3		
D	2–36–11	0–49–0	2–36–11	2–36–11			
E	0–49–0	11–36–2	0–49–0				
F	0–49–0	11–36–2					
D1	2–36–11						

Tabla 4.7: Resultados de los clasificadores con las bases de casos del UCI–Repository. "+"/"−" representa mejora/degradación estadísticamente significativa sobre *k*–NN.

	A	B	C	D	E	F
IR	96.00%	96.00%	96.00%	96.00%	96.00%	96.00%
WI	97.19%	97.19%	96.63%	97.19%	95.51%	96.63%
PI	75.39%	73.31%	76.30%	73.31%	75.52%	75.52%
GL	71.50%	71.03%	72.90%	71.50%	73.36%	75.23%
CL	57.10%	58.42%	58.09%	58.42%	57.43%	57.43%
GD	96.70%	95.70%	96.70%	96.70%	95.70%	96.70%
SN	87.02%	87.02%	88.94%	87.98%	90.87%	90.38%
LD	65.22%	63.48%	65.22%	65.51%	63.48%	65.51%
ZO	96.04%	96.04%	97.03%	97.03%	96.04%	97.03%
TT	84.24%	82.57%−	78.18%−	84.24%	80.90%	80.90%
L7	74.48%	74.36%	74.48%	74.36%	74.36%	74.36%
L24	72.34%	73.80%+	73.74%+	73.80%+	73.52%+	73.52%+
W21	85.42%	85.04%	85.42%	85.04%	85.04%	85.04%
W40	84.54%	84.62%	85.40%	84.62%	84.46%	84.46%
SO	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
F1	80.68%	82.93%+	80.58%	82.93%+	82.93%+	82.93%+
F2	96.34%	96.62%	96.25%	96.62%	96.62%	96.62%
F3	99.44%	99.53%	99.44%	99.53%	99.53%	99.53%
LR	96.02%	96.42%+	96.02%	96.42%+	96.42%+	96.42%+

cación es importante, pero está más allá del ámbito de este trabajo. Se deja para trabajos futuros porque este trabajo se centra en los *k*–NN y ha sido materialmente imposible incluirlo aquí por razones de tiempo. Sin embargo, si el lector está interesado en la comparación con otros métodos de clasificación, creo que la metodología de los experimentos es suficientemente estándar para permitir consultar resultados de otros artículos y establecer comparaciones que probablemente serán adecuadas.

En cualquier caso, es conveniente dejar claro que estos clasificadores obtienen siempre un 100% de acierto con los datos del conjunto de entrenamiento porque conservan todos los casos que aprenden. En trabajos futuros se estudiarán mecanismos de olvido para conservar sólo un subconjunto de los ejemplos de entrenamiento. Entonces tendrá sentido estudiar el acierto con el conjunto de entrenamiento y el tamaño del conjunto conservado.

Tradicionalmente se ha supuesto que los métodos basados en distancias tienen problemas con el ruido y los atributos irrelevantes, pero los resultados de las bases LED7 y LED24, y Waveform–21 y Waveform–40 son comparables, a pesar del ruido y de los 17 y 19 atributos irrelevantes. Los resultados con estas bases de casos están cercanos al porcentaje de acierto de su clasificador Bayesiano óptimo (74% y 86% para los dominios LED display and Waveform respectivamente).

Tabla 4.8: Resultados de los clasificadores con las bases de casos sintéticas. "+"/"−" representa mejora/degradación estadísticamente significativa sobre *k*–NN.

	A	B	C	D	E	F
bandas 5u	96.20%	96.00%	99.40%+	96.00%	99.20%+	99.20%+
bandas 5h	95.40%	95.80%	98.80%+	95.80%	98.20%+	98.20%+
bandas 5p	95.40%	94.80%	98.20%+	94.80%	98.60%+	98.60%+
bandas 10u	88.20%	89.60%	97.00%+	89.60%	96.60%+	96.60%+
bandas 10h	89.80%	88.80%	97.20%+	88.80%	96.80%+	96.80%+
bandas 10p	87.80%	88.40%	95.80%+	88.40%	96.80%+	96.80%+
bandas 20u	72.60%	72.40%	93.80%+	72.60%	94.00%+	94.00%+
bandas 20h	71.60%	72.00%	93.20%+	72.00%	92.20%+	92.20%+
bandas 20p	71.60%	71.40%	91.80%+	71.60%	89.80%+	89.80%+
Gauss	87.60%	88.20%	86.60%	88.20%	86.40%	86.40%

(continúa en la página siguiente)

Tabla 4.8: Resultados de los clasificadores con las bases de casos sintéticas. "+"/"−" representa mejora/degradación estadísticamente significativa sobre *k*–NN.

	A	B	C	D	E	F
anillo a 3u	90.60%	91.40%	93.00%+	91.40%	92.20%	92.20%
anillo a 3h	91.40%	93.00%+	91.40%	93.00%+	93.00%+	93.00%+
anillo a 3p	93.80%	93.60%	93.60%	93.60%	93.80%	93.80%
anillo a 6u	80.80%	81.20%	80.80%	81.20%	81.20%	81.20%
anillo a 6h	80.00%	81.20%	80.00%	81.20%	81.20%	81.20%
anillo a 6p	82.80%	82.20%	82.40%	82.20%	81.80%	81.80%
anillo a 9u	70.00%	71.20%	70.00%	71.20%	69.60%	69.60%
anillo a 9h	70.60%	69.80%	70.20%	69.80%	69.80%	70.20%
anillo a 9p	72.60%	72.60%	72.60%	72.60%	72.60%	72.60%
anillo r 3u	95.40%	95.00%	96.20%	95.00%	95.60%	95.60%
anillo r 3h	95.80%	95.00%	95.20%	95.00%	95.60%	95.60%
anillo r 3p	94.20%	94.40%	94.60%	94.40%	94.60%	94.60%
anillo r 6u	88.00%	88.20%	88.20%	88.20%	88.60%	88.60%
anillo r 6h	89.80%	91.20%+	89.80%	91.20%+	91.20%+	91.20%+
anillo r 6p	85.60%	85.60%	84.80%	85.60%	85.80%	85.80%
anillo r 9u	82.60%	81.60%	84.40%	81.60%	82.80%	82.80%
anillo r 9h	78.20%	78.60%	79.20%	78.60%	79.60%	79.60%
anillo r 9p	77.00%	77.80%	78.20%	77.80%	79.20%	79.20%
senos 3u	93.40%	92.60%	93.40%	92.60%	92.60%	92.60%
senos 3h	91.40%	92.00%	90.60%	92.00%	90.80%	90.80%
senos 3p	93.00%	91.80%	94.40%	91.80%	93.80%	93.80%
senos 6u	85.80%	86.20%	86.00%	86.20%	86.60%	86.60%
senos 6h	83.60%	84.00%	84.20%	84.00%	84.80%	84.80%
senos 6p	83.80%	84.40%	84.60%	84.40%	85.00%	85.00%
senos 9u	75.80%	75.80%	75.80%	76.00%	75.80%	76.00%
senos 9h	78.40%	78.80%	78.60%	78.80%	78.60%	78.60%
senos 9p	76.60%	76.40%	79.80%+	77.20%	77.40%	77.40%
cuad. 2u	97.60%	97.20%	97.60%	97.20%	97.20%	97.20%
cuad. 2h	96.80%	97.00%	96.80%	97.00%	97.00%	97.00%
cuad. 2p	98.20%	98.60%	98.20%	98.60%	98.60%	98.60%

(continúa en la página siguiente)

Tabla 4.8: Resultados de los clasificadores con las bases de casos sintéticas. "+"/"−" representa mejora/degradación estadísticamente significativa sobre *k*–NN.

	A	B	C	D	E	F
cuad. 4u	93.80%	94.40%	94.40%	94.40%	94.20%	94.20%
cuad. 4h	93.80%	93.80%	93.60%	93.80%	93.40%	93.60%
cuad. 4p	93.20%	92.60%	93.60%	92.60%	92.00%	92.00%
cuad. 6u	87.20%	87.20%	86.40%	87.20%	87.00%	87.00%
cuad. 6h	85.80%	86.80%	87.80%	86.80%	87.80%	87.80%
cuad. 6p	84.60%	85.20%	84.60%	85.20%	85.20%	85.20%
cuad. 8u	75.80%	76.40%	75.80%	76.40%	76.40%	76.40%
cuad. 8h	80.80%	81.00%	82.40%+	81.00%	82.80%+	82.80%+
cuad. 8p	84.00%	83.80%	84.00%	83.80%	83.80%	83.80%

El valor óptimo de *k* depende de la base de casos misma y de la medida de distancia que se emplee. Por supuesto, si usamos el valor óptimo de *k* de la distribución uniforme con las distribuciones mitad y progresiva, entonces el porcentaje de acierto desciende. Pero, en estas últimas distribuciones, el valor óptimo de *k* es más cercano al valor óptimo en las regiones más densas. Según vamos modificando la distribución de puntos a lo largo del espacio, el valor óptimo de *k* cambia de tal modo que tiende a clasificar mejor las regiones densas y peor las poco pobladas. En otras palabras, *k*–NN prefiere conseguir la clase correcta en las regiones más densas a costa de las pocas pobladas, pero alcanzando un compromiso entre la población y el valor óptimo de *k* de las diferentes regiones. Lo mismo sucede con los métodos de los ε-entornos.

Otra cuestión interesante es determinar cuándo se deben usar los clasificadores *k*–NN y cuándo los ε–entornos. Se ha encontrado cierta evidencia sobre la debilidad de *k*–NN cuando la distribución de puntos no es constante a lo largo del espacio (y por tanto el valor óptimo de *k* varía). La tabla 4.9 muestra que *k*–NN mejora ligeramente a los ε–entornos en las variantes uniforme y progresiva, pero es inferior en la variante mitad (0–14–2). En la variante mitad, los métodos *k*–NN (A y C) reducen sus resultados con respecto a la variante uniforme, mientras que los métodos de los ε–entornos ven incrementado su porcentaje de acierto. Este hecho

Tabla 4.9: Acierto medio de los clasificadores con las variantes uniforme, mitad y progresiva. Δ muestra la variación con respecto a la variante uniforme.

		Unif.	Mitad		Prog.	
	Clasificador	Media	Media	Δ	Media	Δ
A	*k*–NN	85.86%	85.83%	–0.03%	85.89%	+0.03%
B	ε–entorno	86.03%	86.18%	+0.15%	85.85%	–0.18%
C	*k*–NN Heur	88.26%	88.06%	–0.20%	88.20%	–0.06%
D	ε–entorno^{k-NN}	86.05%	86.18%	+0.13%	85.91%	–0.14%
E	ε–entorno Heur	88.10%	88.30%	+0.20%	88.05%	–0.05%
F	ε–entorno^{k-NN} Heur	88.11%	88.34%	+0.23%	88.05%	–0.06%
D1	ε–entorno^{1-NN}	86.04%	86.18%	+0.14%	85.86%	–0.18%
F1	ε–entorno^{1-NN} Heur	88.10%	88.34%	+0.24%	88.05%	–0.05%

parece indicar que los métodos *k*–NN tienen más problemas que los ε–entornos si hay grandes áreas con una distribución de puntos muy diferente, y con un valor de *k* óptimo también bastante diferente, mientras que los ε–entornos prefieren este tipo de situaciones.

Por otra parte, *k*–NN es preferible si el valor óptimo de *k* es constante a lo largo del espacio (variante uniforme), o si varía progresivamente (variante progresiva), pero sin llegar a formar regiones claras con valores muy diferentes. *k*–NN Heur es preferible claramente en las bases de casos de las bandas, y los métodos de los ε–entornos en las bases de los anillos (tanto con área constante como de radio constante).

4.6 Conclusiones

En este capítulo se han analizado algunas variantes del método de clasificación *k*–NN, todas ellas basadas en la adición de hasta tres características. Se ha estudiado la utilidad de estas características y se ha realizado una comparación exhaustiva con *k*–NN (se han usado seis funciones de distancia diferentes, y se han probado los

clasificadores con 68 bases de casos). Las tres características propuestas se han revelado útiles, y las variantes propuestas tienden a superar los resultados del *k*–NN básico.

Como era de esperar, la elección de la medida de distancia es importante. Afecta de manera decisiva al rendimiento del clasificador, indepedientemente de la familia de clasificadores, y su elección depende de la base de casos. En las bases de datos con atributos irrelevantes es útil una función sensible al contexto. Esto puede lograrse usando algún método que proporcione información sobre la importancia relativa de los atributos, por ejemplo con la correlación de los atributos con la clase o con la Información Mutua.

Se observa que existe cierta relación entre el porcentaje de acierto obtenido por las distancias (1–NN) y el logrado por otros métodos de clasificación basados en esas mismas distancias. Considerando que los métodos 1-NN son relativamente rápidos, se propone la siguiente heurística: *entrena y prueba métodos 1–NN con un conjunto de varias medidas de distancia y utiliza el método de clasificación basado en distancias deseado (k–NN,* ε*–entorno,* ε*–entorno*$^{k-NN}$*...) con la medida de distancia que ofrece la mejor tasa de acierto con 1–NN.*

Esta heurística es usada por los mejores clasificadores y claramente incrementa su porcentaje de acierto. Para acelerar la selección de la medida de distancia (especialmente si proponemos un gran número de ellas) es posible usar las ideas de *Racing Algorithms* [MM97]: probar los modelos en paralelo, descartar rápidamente aquellos que son claramente inferiores usando límites estadísticos, y concentrar el esfuerzo computacional en diferenciar entre los mejores modelos.

El hecho de que esta heurística obtenga buenos resultados permite además elegir la distancia que mejores resultados ofrece con 1–NN para luego emplearla con otros clasificadores basados en distancias. Así, por ejemplo, si deseamos utilizar una medida de distancia que tiene uno o varios parámetros que controlan su comportamiento, podemos intentar optimizar esos parámetros usando un clasificador 1–NN, y posteriormente emplearla con otro clasificador basado en distancias.

La heurística no nos garantiza que esa sea la mejor distancia para usar en combinación con ese clasificador, sino que indica que esa distancia tenderá a ser una

buena distancia para usar con ese clasificador. Usar esta aproximación para seleccionar la medida de distancia puede aportar dos tipos de ventajas. Por una parte se puede reducir el tiempo necesario para estimar los parámetros de la distancia, debido principalmente al bajo tiempo de entrenamiento de un clasificador 1–NN. Por ejemplo, si vamos a utilizar el método *k*–NN o los ε–entornos, y elegimos la distancia empleando directamente esos métodos, también necesitamos estimar a la vez el valor del parámetro del método de clasificación (*k* o ε en este ejemplo). Este motivo puede ser importante si contamos con un tiempo limitado para la selección de la medida de distancia, o si es demasiado elevado el tiempo que se necesita si se emplea directamente el clasificador deseado. Por otra parte, elegir la distancia con 1–NN puede ser beneficioso porque al ser un clasificador más sencillo puede facilitar la realización de estudios teóricos sobre la distancia, sus propiedades y parámetros.

Si comparamos los resultados de los clasificadores, podemos observar que el método ε–entorno^{k-NN} Heur mejora significativamente a los métodos *k*–NN (15–53–0), *k*–NN Heur (7–58–3) y ε–entorno (11–55–2) frecuentemente. ε–entorno^{k-NN} Heur mejora significativamente al método *k*–NN en las bases de casos Led24, Letter Recognition, Solar Flare 1, todas las bases de las bandas (bandas 5, 10 y 20 variantes uniforme, mitad y progresiva), cuadrados 8×8 mitad, 6 anillos radios constante mitad y 3 anillos área constante mitad. ε–entorno^{k-NN} Heur mejora significativamente al método *k*–NN en las bases de casos Tic–Tac–Toe, Letter Recognition, Solar Flare 1, Solar Flare 2, 6 anillos radios constante mitad, 6 anillos radios constante progresiva y 3 anillos área constante mitad; y es significativamente peor en las bases WaveForm–40, bandas 20 progresiva y cuadrados 4×4 progresiva. ε–entorno^{k-NN} Heur mejora significativamente al método ε–entorno en todas las bases de casos de las bandas (bandas 5, 10 y 20 variantes uniforme, mitad y progresiva), cuadrados 8×8 mitad y 3 senos progresiva; y es significativamente peor en las bases Gauss y 3 senos mitad.

No existe diferencia significativa entre ε–entorno^{k-NN} Heur y ε–entorno Heur (0–68–0), aunque ε–entorno^{k-NN} Heur logra resultados ligeramente mejores y además es menos sensible a la elección de ε. En general, cuando no hay diferencia significativa desde un punto de vista estadístico, ε–entorno^{k-NN} es el método que logra mejores resultados con la mayoría de las bases de casos. Tampoco existe

diferencia significativa entre ε–entorno^{k-NN} Heur y ε–entorno^{1-NN} Heur, aunque el primero logra resultados ligeramente mejores y necesita mucho más tiempo.

El tiempo requerido por todos los clasificadores es similar, con la excepción de ε–entorno^{k-NN} y ε–entorno^{k-NN} Heur. Por eso, como regla general, y sin tener ninguna información a priori, es recomendable usar ε–entorno^{1-NN} Heur, el método que logra mejor tasa de acierto con un consumo de tiempo similar.

Se ha encontrado cierta evidencia sobre la debilidad de *k*–NN cuando la distribución de puntos no es constante a lo largo del espacio (y por lo tanto el valor óptimo de *k* varía). Los métodos *k*–NN tienen más problemas que los ε–entornos si existen grandes zonas del espacio con una distribución de puntos y un valor óptimo de *k* muy diferente. El método *k*–NN Heur es claramente preferible en las bases de casos de las bandas, y los ε–entornos en las bases de casos de los anillos, tanto de área como de radio constante.

En un futuro creemos que puede ser interesante el estudio de nuevas medidas de distancia que permitan ajustarse mejor a las peculiaridades de la base de casos. Se puede estudiar también mecanismos para reducir las necesidades de memoria con técnicas de compactación de información, técnicas de recuerdo y olvido selectivo (recordar los ejemplos útiles y olvidar los que aportan menos información), con el objetivo de recorrer el camino que conduce hacia métodos que muestren más características del Aprendizaje Basado en Ejemplos (*Instance–Based Learning* o IBL) [AKA91] [Aha92] y el Razonamiento Basado en Casos (*Case–Based Reasoning* o CBR) [Kol92] [Kol93] [AP94] [Aha98].

Capítulo 5

Métodos de Clasificación Basados en la Distancia de las Bandas

En este capítulo se van a desarrollar las ideas de las distancias basadas en bandas que se han introducido en la sección 3.6.1 para usarlas en problemas de clasificación.

Se han realizado muchos estudios sobre la similitud ([PLA96] [Rit92] [Rit95]), y sobre las medidas de distancia y los métodos de clasificación basados en distancias (por ejemplo [Wet94] [RA95] [RA99]).

Es frecuente definir el grado de similitud entre dos objetos x e y mediante medidas de distancia $d(x,y) : D \times D \rightarrow \mathrm{R}$, porque nosotros estamos familiarizados con ellas y desde un punto de vista práctico son fáciles de definir y muy intuitivas. Además, como se ha comentado en el apartado 3.4, dada una medida de distancia se puede definir fácilmente la medida de similitud equivalente y viceversa.

Como consecuencia, algoritmos de clasificación tan populares como 1–NN y k–NN [FH51] [CH67] [Das91] emplean medidas de distancia para buscar "los casos más similares" al caso nuevo, basándose en el valor de algunas características. Lo más frecuente en este tipo de algoritmos es emplear una de las distancias clásicas de la Geometría, sobre todo la distancia Euclídea. Dado un caso o punto nuevo e,

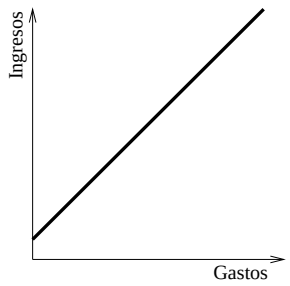


Figura 5.1: Compañía con beneficios iguales.

se seleccionan los puntos más cercanos a *e* de acuerdo a esa función de distancia y se propone la asignación de una clase basándose en las clases de sus puntos más cercanos y su distancia a *e*.

Si se consideran sólo las distancias usuales, entonces no se pueden obtener algunos tipos de relaciones de similitud (y distancia) que resultan bastante naturales, y es difícil afrontar problemas donde los ejemplos están agrupados de acuerdo a otros patrones o esquemas. Por ejemplo, si conocemos los ingresos y gastos de algunas compañías y la clasificación se realiza de acuerdo a los beneficios, los ejemplos que pertenecen a la misma clase (y en ese sentido son similares) están distribuido a lo largo de una recta (fig. 5.1). Este es sólo un ejemplo de un amplio conjunto de problemas donde los ejemplos están agrupados en bandas, y las distancias usuales fallan.

En este capítulo se propone un enfoque diferente del habitual:

- Se propone entender la función de distancia en sentido amplio. Se necesita una función que mida similitud o disimilitud entre objetos y que proporcione valores bajos para los casos con clases iguales o similares. Pero en general no es necesario que esta función cumpla todas las propiedades de una distancia geométrica. Por lo tanto en este capítulo se propone ser flexibles y emplear "funciones de distancia" en sentido amplio, entendidas como medidas de distancia que están adaptadas al problema específico que tratamos de resolver.

- Se propone emplear funciones de distancia locales e ir más allá de la idea de buscar los puntos más cercanos al caso nuevo *e*. Se propone una primera

fase de entrenamiento en la que cada punto aprende una banda o hiperplano que pasa por ese punto y se ajusta mejor a la distribución de puntos en sus alrededores. Entonces, dado un caso nuevo *e*, cada punto conocido puede proporcionar una distancia de cómo de lejos se encuentra él del caso *e* desde su punto de vista (de acuerdo a la banda). Esta aproximación el la opuesta de la habitual. La aproximación usual busca los puntos más cercano desde *e*, pero ahora cada punto "dice" cómo de lejano o cercano ve a *e*, teniendo en cuenta lo que sucede a su alrededor.

Este enfoque aporta varias ventajas:

- Puede ser visto como una medida de distancia local: cada punto conoce su propia medida de distancia, que tiene en cuenta las peculiaridades de la zona concreta donde se encuentra. En conjunto, la reunión de todas esas medidas locales proporciona una medida de distancia que puede variar a lo largo del espacio para ajustarse de manera adecuada a las características especiales de las diferentes regiones del espacio.

- Con muchos hiperplanos se puede aproximar localmente casi cualquier forma, ¡y nosotros tenemos un hiperplano asociado a cada punto!

- Cada punto puede aprender el hiperplano que minimiza localmente la distancia a los puntos de su clase, e incluso que maximiza la distancia a los puntos de otras clases. De esta manera los puntos tienden a asignar distancias menores a los puntos que se encuentran en las direcciones del hiperplano (la dirección de los puntos de su clase) y distancias mayores el la dirección perpendicular.

  Con el enfoque usual no se puede acercar o alejar puntos dependiendo de sus clases, porque el caso nuevo *e* todavía no tiene una clase (precisamente se está intentando asignar una clase a *e*).

- Además se puede tener mucho más conocimiento que simplemente con los puntos en bruto, y se puede intentar extraer información del conjunto de hiperplanos que han aprendido los puntos.

5.1 Un Algoritmo de Aprendizaje de Bandas o Hiperplanos

Las distancias usuales de la Geometría son útiles en gran número de situaciones, pero a veces resulta más adecuado emplear otro tipo de distancias. En la sección 3.6.1, ecuación (3.16), y en un trabajo previo [LC01], hemos presentado una medida de distancia, que agrupa los puntos de acuerdo a bandas a lo largo de un hiperplano H en R^n (una recta en R^2) $d_{\alpha,ancho}(x,y) : \mathrm{R}^n \times \mathrm{R}^n \rightarrow \mathrm{R}_0^+$ definida como:

$$d_{\alpha,ancho}(x,y) = ancho \left| \sum_{i=1}^{n} \cos\alpha_i (x_i - y_i) \right|$$

donde $\alpha = (\alpha_1, \cdots, \alpha_n) \in [0, 2\pi]^n$ es el conjunto de ángulos formados entre los ejes y el vector unitario $v = (v_1, \cdots, v_n)$ que es perpendicular al hiperplano deseado H (fig. 5.2). $ancho \in \mathrm{R}_0^+$ controla la anchura de la banda de puntos que se encuentran a una distancia dada (valores menores implica que los puntos se acercan a H), y $|.|$ es la función valor absoluto en R. Además v cumple que $\sum_{i=1}^{n} v_i^2 = 1$, por lo tanto existen sólo n grados de libertad en la elección de los parámetros. Nótese que $(x_i - y_i)$ puede ser menor, igual o mayor que 0. Se usa R^n por simplicidad, y para dominios simbólicos $(x_i - y_i)$ representa la distancia parcial entre x_i e y_i en ese dominio.

Esta distancia no es una métrica desde el punto de vista de las Matemáticas, pero es una pseudo–métrica porque $d(x,y) = 0 \not\Rightarrow x = y \quad \forall x,y \in D$. La figura 5.2 muestra un ejemplo de este tipo de distancias.

5.1.1 Aprendizaje de la Dirección de la Banda

Dejando a un lado la anchura de la banda, el objetivo es que cada punto de entrenamiento aprenda lo que está sucediendo a su alrededor. Y se pretende que cada punto p aprenda una banda o hiperplano que pase por p y se ajuste lo mejor posible a los puntos que lo rodean, es decir, el hiperplano H que minimice la distancia $d_{\alpha,ancho}$ a los puntos de su alrededor. Se define el *ajuste de un hiperplano* $H = v^{\perp}$

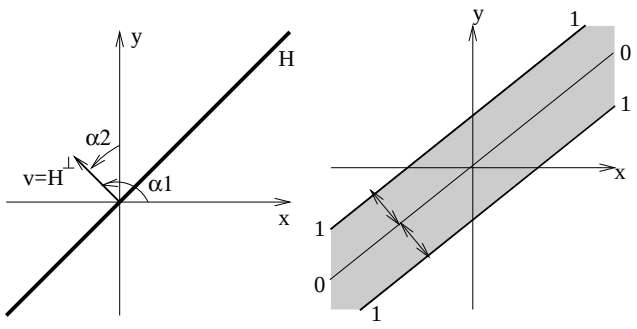


Figura 5.2: Definición de una banda en R^2 y un ejemplo de banda a lo largo de un hiperplano en R^2.

como

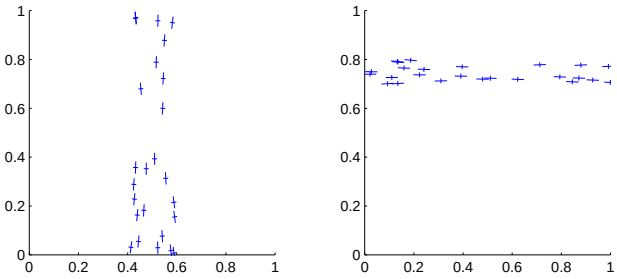
$$ajuste\_a_H = \sum_{i=1}^{n} \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right)^2 \tag{5.1}$$

donde x_0 es un sinónimo de p, el punto respecto del que se realizan los cálculos para aprender la dirección del hiperplano, P es el conjunto de puntos conocidos, N es el número de atributos o dimensiones, n es el número de puntos ($|P|$), x_{0j} es el j–ésimo atributo o coordenada de x_0, x_{ij} es el j-ésimo atributo del i–ésimo punto de P, y $v_j = \cos\alpha_j$ es el j–ésimo coseno director de H.

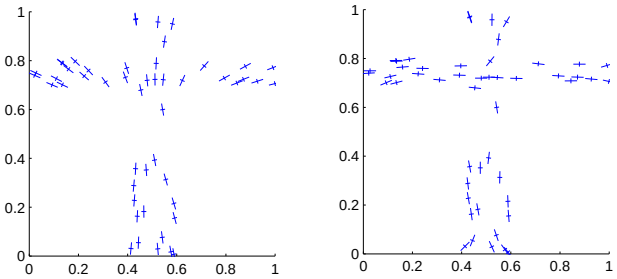
Ahora se puede expresar el hiperplano H que pasa por p y mejor se ajusta a los puntos de su alrededor como el hiperplano con menor valor de ajuste.

Se han realizado diversos experimentos para observar el comportamiento de esta manera de seleccionar los hiperplanos. Por ejemplo, se ha generado un conjunto de puntos aleatorios dentro de una banda horizontal de anchura 0.2 (fig. 5.3.a) y otro conjunto dentro de una banda vertical de anchura 0.3 (fig. 5.3.b). Los puntos aprenden razonablemente bien la dirección de la banda en que se encuentran. Pero si unimos estas dos bandas (los puntos de ambas), los puntos tienden a apuntar hacia el centro geométrico del conjunto de puntos, en lugar de aprender la dirección de la banda en que se encuentran (fig. 5.3.c).

En este ejemplo se muestra claramente que este método de determinar las bandas es demasiado global, y la dirección que selecciona cada punto está afectada por puntos demasiado alejados. Por tanto, se debe emplear un método más local. Para



(a) Banda Vertical (b) Banda Horizontal



(c) Unión con $ajuste\_a_H$ (d) Unión con $ajuste\_e_H$

Figura 5.3: Bandas que aprende cada punto de una banda vertical, una banda horizontal band y su unión en R^2. La posición de cada punto se encuentra en la intersección de las dos líneas que aparecen, y la dirección de su banda es la indicada por el segmento de mayor longitud.

conseguirlo se han propuesto y estudiado cuatro variantes locales de este método básico que tienen en cuenta la distancia de los puntos.

$$ajuste\_b_H = \sum_{x_i \in P} \frac{1}{d(x_i, x_0)} \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right)^2 \tag{5.2}$$

$$ajuste\_c_H = \sum_{x_i \in E(x_0, \varepsilon)} \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right)^2 \tag{5.3}$$

$$ajuste\_e_H = \sum_{x_i \in P} \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right)^2 e^{-\frac{4}{\varepsilon^2} d(x_i, x_0)} \tag{5.4}$$

$$ajuste\_f_H = \sum_{x_i \in E(x_0,\varepsilon)} \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right)^2 e^{-\frac{4}{\varepsilon^2} d(x_i,x_0)} \quad (5.5)$$

donde $E(x,\varepsilon) = \{x\prime \in P \text{ tal que } d(x\prime,x) \leq \varepsilon\}$ y $d(x_i,x_0)$ es la distancia Euclídea entre x_i y x_0, es decir, $\sqrt{\sum_{j=1}^{N}(x_{ij} - x_{0j})^2}$.

Estas variantes aprenden bien una única banda, y la fig. 5.3.d muestra el comportamiento de una de ellas con la unión de las dos bandas.

El método de la definición (5.2) no es sufientemente local, y los métodos de las definiciones (5.3), (5.4) y (5.5) tienen un parámetro real positivo ε que permite controlar con suavidad la localidad de estos métodos y variarla para obtener métodos que tienen un comportamiento que va desde muy local hasta muy global. Con un valor de ε alto, estos métodos exhiben un comportamiento similar a los que no usan la función *kernel* (métodos (5.1), (5.2) y (5.3)). Al reducir ε se obtienen bandas con un comportamiento cada vez más local Los tres métodos tienen un comportamiento muy parecido con los valores adecuados de ε, por tanto hemos elegido uno de ellos y nos hemos concentrado en su estudio.

Ahora tenemos un parámetro ε que controla el grado de localidad que el punto debe usar cuando aprenda la dirección de su hiperplano. Básicamente tenemos dos alternativas para elegir su valor. En primer lugar ε puede ser un valor real fijo, constante para toda la base de casos. Se puede proporcionar directamente ese valor o puede ser calculado, por ejemplo como resultado de alguna expresión que permita tener en cuenta las características de cada base de casos. O bien, ε puede ser un valor real diferente para cada punto, que se calcula basándose en los alrededores del punto, es decir, cada punto tienen su propio ε que recoge las características especiales de esa región.

En principio el valor de ε debe ser suficientemente grande para incluir la información de los alrededores del punto. Lo ideal sería incluir al *cluster* o grupo de puntos donde se encuentra ubicado el punto. Pero no debe ser demasiado grande para que no se vea afectado por puntos demasiado alejados, que son puntos aislados o pertenecen a otros grupos de puntos.

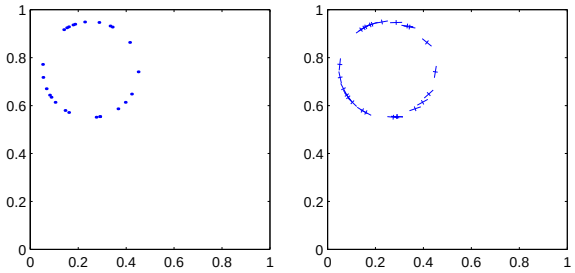
Hemos realizado algunos experimentos para encontrar un método que permita determinar de manera automática el valor de ε para cada punto, pero nos hemos

encontrado con más dificultades de las esperadas y los resultados no han sido concluyentes. Esta parte de la investigación está directamente relacionada con el *clustering*, es decir dado un conjunto de ejemplos determinar en primer lugar el número de conjuntos o clases en que están agrupados esos ejemplos, y posteriormente, dado un punto, determinar a cuál de esos conjuntos pertenece. Dado que este problema cae fuera del ámbito de nuestra línea principal de trabajo, y dadas las dificultades que hemos encontrado, hemos decidido dejar la determinación automática del valor de ε como una línea de trabajo que debe abordarse en un futuro. Aquí podrían usarse técnicas ya conocidas de *clustering* para determinar el tamaño de la nube donde se encuentra el punto, y elegir un valor de ε que recoja la información de los puntos de la nube pero sin llegar a incluir información de otras nubes de puntos. Otra posibilidad sería explorar alguna medida que recoja cómo influyen los puntos de la misma clase/nube y los de otras en el ajuste del hiperplano. Para así determinar un valor de corte ε que permita tener una influencia grande antes de acabar la nube donde se encuentra el punto y que reduzca notablemente la influencia mucho antes de alcanzar otras nubes de puntos.

En las figuras 5.4, 5.5, 5.6, 5.7 y 5.8 se muestra la dirección de las bandas que aprenden algunos puntos generados aleatoriamente sobre el cuadrado unidad $[0,1]\times[0,1]$. A la izquierda se muestra la posición que ocupa cada punto aleatorio, y a la derecha se muestra la dirección de la banda que aprende cada punto. Aquí se cada punto se encuentra en la intersección de dos segmentos de diferente longitud, y el segmento de mayor longitud muestra graficamente la dirección de la banda que ha aprendido ese punto.

En la figura 5.4.a se han elegido los puntos de manera aleatoria a lo largo de una circunferencia, y se han elegido valores de ε relativamente pequeños, lo que hace que la búsqueda de bandas sea relativamente local. En la mitad derecha de la figura podemos observar cómo se elige la dirección que lleva hacia los puntos más cercanos, y se "dibuja" o se aproxima localmente una circunferencia mediante una serie de pequeños segmentos. Si se amplia mucho el valor de ε nos encontraremos con que los puntos tienden a apuntar hacia el centro de la circunferencia. Profundizando en esta propiedad de las bandas, se puede por ejemplo aprender bandas con dos valores de ε: uno relativamente pequeño y otro relativamente grande (para el

(a) Puntos de una circunferencia



(b) Puntos de un círculo

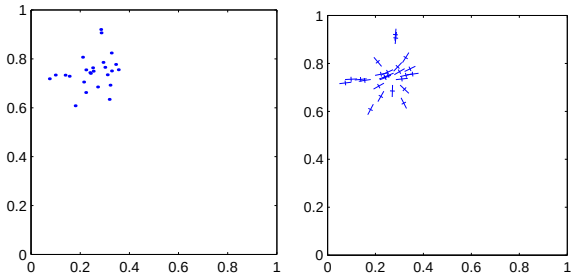


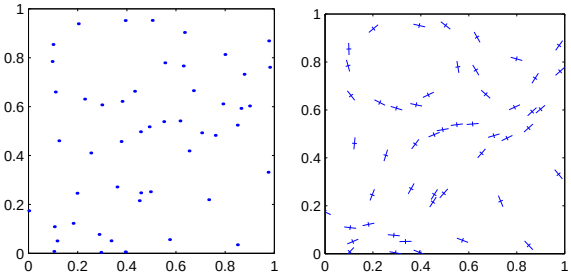
Figura 5.4: Bandas que aprenden puntos aleatorios de una circunferencia y un círculo. A la izquierda se muestra la posición de cada punto y a la derecha la dirección de cada banda (segmento de mayor longitud).

tamaño de la nube de puntos). Así podemos emplear el valor menor para reconocer siluetas o figuras, y el valor mayor reconocer dónde se encuentra el centro de las figuras, y combinando la información de ambos podemos estudiar más características, como grados de curvaturas, ubicación de los centros de curvatura de esas figuras...

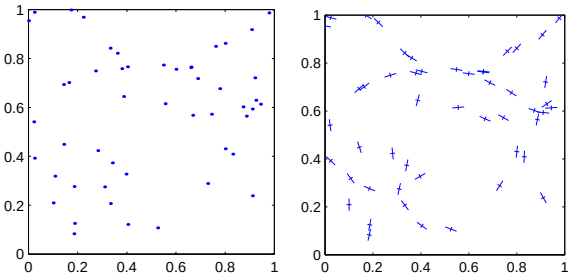
En la figura 5.4.b se han elegido los puntos de manera aleatoria en el interior de un círculo. La nube es relativamente compacta y puede observarse claramente cómo las bandas apuntan hacia el centro de la nube (con valores de ε suficientemente grandes).

En la figura 5.5 los puntos están generados aleatoriamente sobre el cuadrado unidad $[0,1]\times[0,1]$ sin ningún tipo de restricción. En esta figura se muestran tres ejemplos distintos en los que se han generado 50 puntos aleatorios. Si cada punto aprende la banda que mejor se ajusta a su entorno, tenderá a elegir direcciones que

(a) Puntos aleatorios



(b) Puntos aleatorios



(c) Puntos aleatorios

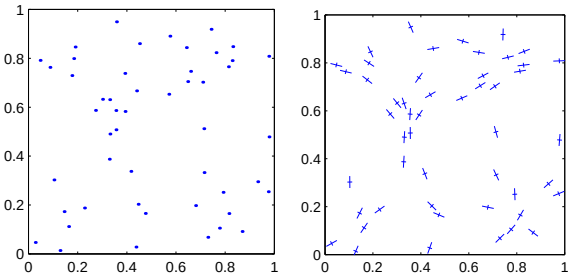


Figura 5.5: Bandas que aprenden puntos generados aleatoriamente. A la izquierda se muestra la posición de cada punto y a la derecha la dirección de cada banda (segmento de mayor longitud).

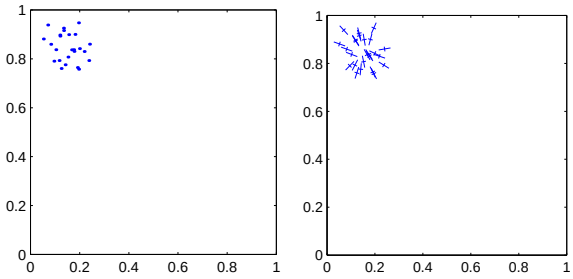
lleve o apunte hacia puntos cercanos. Está intentando encontrar bandas de puntos, o figuras que puedan formar los puntos. Esto es bastante parecido a las famosas pruebas de la Psicología en las que a un sujeto se le presenta una serie de láminas y debe decir las figuras o cosas que reconoce en las láminas. Las respuestas de dos individuos pueden ser totalmente diferentes, porque entra en juego la subjetividad. Si a un individuo le presentaramos las nubes de puntos que aparecen en la parte izquierda de los ejemplos de las figuras 5.5.a, 5.5.b y 5.5.c, y le pidiéramos que intentara encontrar formas o figuras que forman esos puntos, podría decirnos cualquier cosa, desde que no reconoce ninguna forma hasta imaginar cualquier figura.

Eso es precisamente lo que está haciendo el método de las bandas. Por ejemplo, en la figura 5.5.a, si observamos el gráfico de la derecha y las bandas de los puntos de la esquina superior izquierda, parece que indican que ahí existe una circunferencia o un óvalo, y parece también que hay una banda amplia e irregular que recorre la figura desde la esquina superior derecha hacia la esquina inferior izquierda. Si a un humano le proporcionamos el gráfico de la mitad izquierda, probablemente también reconocerá esa circunferencia y esa banda. ¡Y los puntos están generados de forma aleatoria!, aunque claro, quien soy yo para decirle que ahí no hay una circunferencia (o mejor dicho, que él no ve una circunferencia ahí).

Lo mismo sucede con las figuras 5.5.b y 5.5.c. En la figura 5.5.b puede reconocerse en la mitad superior una figura en forma de X aplastada, y en la esquina inferior izquierda, simplemente una nube de puntos un poco alargada. En la figura 5.5.c parece que los puntos se acumulan principalmente en el tercio superior e inferior, pero no en el centro; y parece que en el tercio superior se forma una especie de triángulo o embudo.

En la figura 5.6.a se han elegido los puntos de manera aleatoria en el interior de un cuadrado, y en la figura 5.6.b se han añadido algunos puntos aislados para estudiar cómo afecta su presencia. La nube es relativamente compacta y en la figura 5.6.a puede observarse que las bandas tienen el mismo comportamiento si se trata de puntos en el interior de un círculo o de un cuadrado: tienden a apuntar hacia el centro de la nube si el valor de ε es suficientemente grande. Si añadimos algunos puntos aislados (figura 5.6.b), éstos no afectan a la nube de puntos, y los puntos aislados pueden en principio tomar practicamente cualquier dirección,

(a) Puntos aleatorios de un cuadrado



(b) Puntos de un cuadrado y algunos aislados

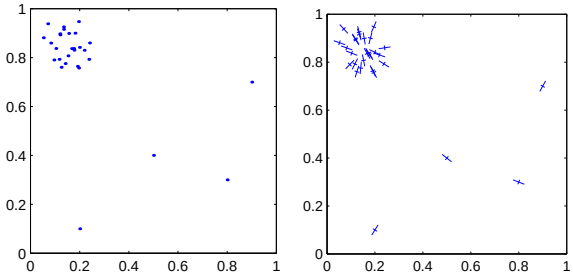


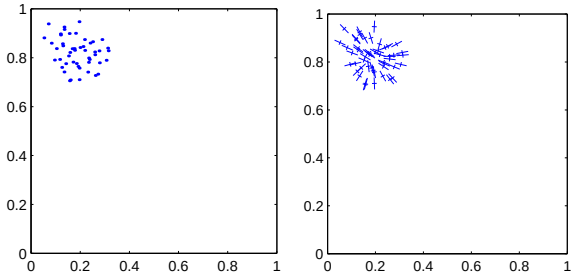
Figura 5.6: Bandas que aprenden puntos aleatorios de un cuadrado. A la izquierda se muestra la posición de cada punto y a la derecha la dirección de cada banda (segmento de mayor longitud).

aunque siempre estarán más influenciados por los puntos más cercanos. Una posible forma de distinguir si un punto está relativamente aislado podría ser mediante el valor de $ajuste\_e_H$ (ecuación (5.4)) que será mucho menor que si se encontrara integrado dentro de una nube de puntos más o menos compacta, aunque un punto alineado dentro de una banda muy estrecha también puede llegar a tener valores muy bajos de $ajuste\_e_H$.

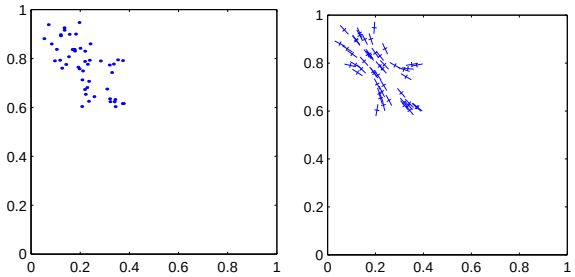
Otro experimento interesante es el que se muestra en la figura 5.7, donde se han elegido los puntos de manera aleatoria en el interior de dos cuadrados. Ahora en la figura 5.7.a los dos cuadrados están casi totalmente superpuestos, en la figura 5.7.b tienen una pequeña zona común, y en la figura 5.7.c están totalmente separados y alejados.

En la figura 5.7.a las dos nubes están superpuestas y los puntos eligen las bandas como si se tratase de una sola nube. Esto es bueno, porque realmente forman una

(a) Dos cuadrados superpuestos



(b) Dos cuadrados con una zona común



(c) Dos cuadrados alejados

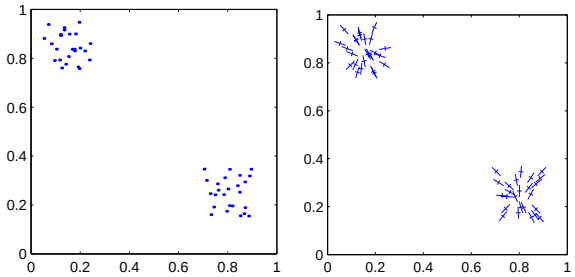


Figura 5.7: Bandas que aprenden puntos aleatorios de dos cuadrados. A la izquierda se muestra la posición de cada punto y a la derecha la dirección de cada banda (segmento de mayor longitud).

sola nube.

En la figura 5.7.b hemos desplazado una de las dos nubes, de forma que la zona común entre ambas sea pequeña. Aquí se produce un efecto curioso, parece que la nube de puntos se "estira", y se parece más a una banda de puntos. Se nota además que los puntos tienden a tomar la dirección de esa banda, aunque también se nota que algunos tienden a apuntar al dentro de la nube.

Si seguimos alejando las nubes, comienzan a comportarse como dos nubes de puntos independientes. En la figura 5.7.c se muestra claramente como los puntos de cada nube tienden a apuntar al centro de su nube.

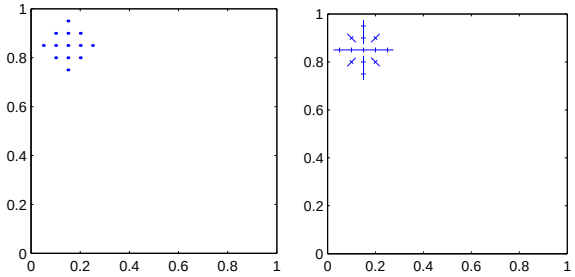
Casi cualquier observador humano también diría que en la figura 5.7.a hay una sola nube de puntos, en la figura 5.7.b hay una nube alargada, y en la figura 5.7.c hay dos nubes claramente diferenciadas.

En la figura 5.8 se han elegido los puntos de acuerdo a una "rejilla", es decir hemos elegido algunos puntos distribuidos periodicamente dentro de alguna figura. En la figura 5.8.a los puntos están dentro de un círculo (aunque parecen puntos de un cuadrado rotado, un rombo, o un diamante). En la figura 5.8.b los puntos están dentro de un cuadrado, y en la figura 5.8.c se han añadido algunos puntos aislados para estudiar su influencia.

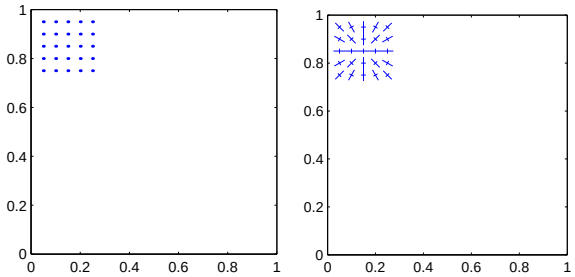
En los tres apartados los puntos tienden a apuntar hacia el centro de la nube. En principio el punto central puede elegir cualquier dirección porque tiene la misma influencia de puntos cercanos desde distintas direccciones. Aunque por la distribución de los puntos, tenderá a elegir una dirección paralela a uno de los ejes, porque ahí están los puntos más cercanos.

En la figura 5.8.c puede observarse que al incluir algunos puntos aislados alejados, su influencia sobre la dirección que eligen los puntos de la nube es muy baja o casi nula. Aunque es suficiente para desequilibrar la "neutralidad" del punto central de la nube por una u otra dirección, y hacer que el punto central apunte hacia los puntos aislados.

(a) Cuadrado rotado 45^o (“Rombo”)



(b) Cuadrado



(c) Cuadrado y puntos aislados

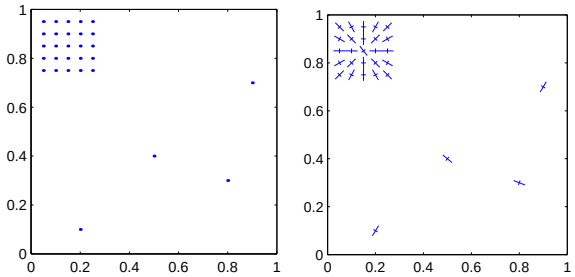


Figura 5.8: Bandas que aprende cada punto de un cuadrado. A la izquierda se muestra la posición de cada punto y a la derecha la dirección de cada banda (segmento de mayor longitud).

5.1.2 Extensión del Aprendizaje de la Dirección de la Banda a Problemas Multiclase

A la hora de elegir el valor de ε se puede penalizar aquellos valores que incluyan puntos de otras clases para que ε tienda a recoger sólo la información correspondiente a la nube de puntos en que se encuentra situado el ejemplo. En principio esto está bien si no vamos a considerar clases o no disponemos de esa información. Pero en un problema con múltiples clases puede resultar bastante útil que el valor de ε sea mayor para poder tener en cuenta puntos cercanos que pertenecen a otras clases. De esta forma al aprender la dirección del hiperplano podemos usar esta información para intentar "alejar", en la medida de lo posible, los puntos de otras clases, y no sólo "acercar" los de la misma clase.

Para determinar el hiperplano óptimo es deseable elegir la dirección de los puntos cercanos de la misma clase y huir de las direcciones donde hay puntos de otras clases. Para alcanzar este doble objetivo se propone que el hiperplano H que aprende cada punto minimice la distancia $d_{\alpha,ancho}$ a los puntos de los alrededores de la misma clase y maximice la distancia $d_{\alpha,ancho}$ a los puntos de otras clases que se encuentran en los alrededores. Este último objetivo es equivalente a minimizar la distancia entre los puntos de otras clases y la dirección perpendicular a H, es decir, minimizar la distancia a $v = H^{\perp}$. Por tanto se define el *ajuste multiclase* de un hiperplano $v = H^{\perp}$ como

$$\begin{aligned} ajuste\_multi_H \;=\; & \sum_{i=1}^{n_=} \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right)^2 e^{-\frac{4}{\varepsilon^2} d(x_i, x_0)} + \\ & F \sum_{i=1}^{n_{\neq}} \left(\sum_{j=1}^{N} (x\prime_{ij} - x_{0j})^2 - \left(\sum_{j=1}^{N} (x\prime_{ij} - x_{0j}) v_j \right)^2 \right) e^{-\frac{4}{\varepsilon^2} d(x\prime_i, x_0)} \end{aligned} \tag{5.6}$$

donde $F \in \mathrm{R}_0^+$, N es el número de atributos o dimensiones, n es el número de puntos, $n_=$ es el número de puntos de la misma clase, $n_{\neq}$ es el número de puntos de otras clases, x_{0j} es el j–ésimo atributo o coordenada de p, x_{ij} es el j–ésimo atributo del i–ésimo punto de la misma clase, $x\prime_{ij}$ es el j–ésimo atributo del i–ésimo punto que es de otra clase, y $v_j = \cos\alpha_j$ es el j–ésimo coseno director de H. Se debe hacer notar que $\sum_{i=1}^{n_=}$ sólo recopila información de los puntos de la misma clase, y $\sum_{i=1}^{n_{\neq}}$

hace lo propio con los de otras clases.

F es un valor real fijo que permite controlar la influencia que se da a los puntos de otras clases. Es decir, hemos introducido un parámetro nuevo que permite controlar la fuerza o el grado en que queremos que la banda evite las direcciones donde hay puntos de otras clases. Elegir $F = 0$ es equivalente a ignorar los puntos de otras clases y aprender el hiperplano solo con los puntos de la misma clase. Elegir $F = 1$ implica dar el mismo peso o importancia a elegir las direcciones donde se encuentran puntos de la misma clase y evitar aquellas donde están los puntos de otras clases. Elegir un valor muy alto para F implica que queremos que la banda "huya" a toda costa de las direcciones donde hay puntos de otras clases, aun a pesar de que el hiperplano tampoco apunte hacia direcciones donde hay puntos de la misma clase.

Ahora de nuevo se puede expresar el hiperplano H que pasa por p y se ajusta mejor a los puntos que rodean a p, como el hiperplano con menor valor de ajuste $ajuste\_multi_H$.

Encontrar el Hiperplano H que minimiza $ajuste\_multi_H$ es equivalente a encontrar el vector $v = H^{\perp}$ que minimiza $ajuste\_multi_H$. Este problema en principio puede parecer bastante complejo y lento de resolver, pero puede abordarse como un problema de minimización con restricciones donde se puede aplicar el método de los multiplicadores de Lagrange para obtener una solución mucho más directa. Así podemos transformarlo en resolver un sistema de N ecuaciones lineales con N incógnitas, mucho más sencillo de afrontar. El desarrollo se encuentra en el Apéndice A. Dado que en algunas bases de casos el número de atributos (y por lo tanto el número de ecuaciones e incógnitas) puede llegar a ser bastante elevado, en las pruebas que se han realizado hemos usado métodos numéricos de aproximación a la solución.

5.1.3 Elección de la Anchura de la Banda

Una vez elegido el valor de ε y el hiperplano $H = v^{\perp}$ para cada punto, estamos en condiciones de enfrentarnos al problema de la anchura de la banda. En realidad no interesa que las bandas sean infinitas como la mostrada en la fig 5.2, porque

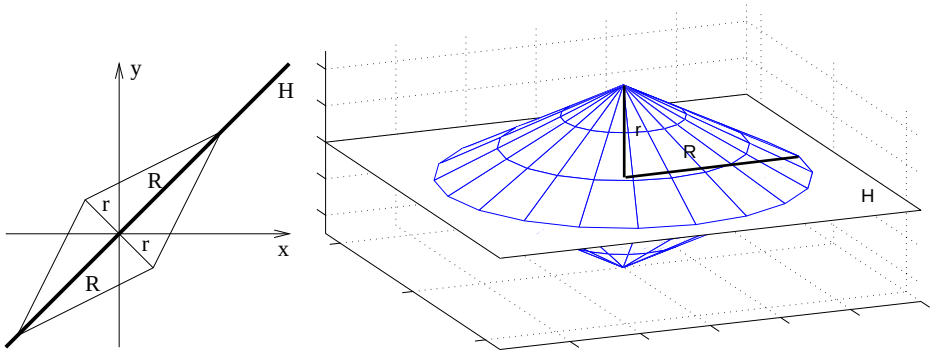


Figura 5.9: Anchura y longitud de una banda o hiperplano en R^2 y R^3.

en la inmensa mayoría de los problemas reales las bandas están limitadas, y no es deseable asignar un valor de distancia muy cercano a 0 a un punto alejado que simplemente está ahí "por casualidad".

Después de considerar distintas alternativas y realizar algunas pruebas preliminares nos hemos decantado por emplear dos parámetros reales r y R que controlan respectivamente la anchura y longitud de la banda o hiperplano. Se calcula la distancia entre un punto $p\prime$ y el hiperplano H que pasa por el punto p como:

$$d_{r,R}(p\prime,H) = \frac{d(p\prime,H)}{r} + \frac{d(p\prime,H^{\perp})}{R} \tag{5.7}$$

donde $d(p\prime,H)$ es la distancia (Euclídea) entre el punto y el hiperplano y $d(p\prime,H^{\perp})$ es la distancia (Euclídea) entre el punto y la dirección perpendicular al hiperplano (v). De esta forma, los puntos que se encuentran a una distancia dada forman un rombo en R^2, dos conos que comparten su base en R^3, y en general dos hiperconos en R^n (fig 5.9).

Dado el hiperplano H mediante el vector unitario $v = (v_1, v_2, \ldots, v_N)$ de sus cosenos directores (lo que es equivalente a proporcionar H mediante ecuaciones paramétricas $H \equiv v_1x_1 + v_2x_2 + \cdots + v_Nx_N = 0$), podemos expresar las distancias anteriores del punto $p\prime = (p\prime_1, p\prime_2 \cdots, p\prime_N)$ al hiperplano H y a su perpendicular como:

$$d(p\prime,H) = |v_1(p\prime_1 - p_1) + v_1(p\prime_2 - p_2) + \cdots + v_N(p\prime_N - p_N)| = |\sum_{i=1}^{N} v_i(p\prime_i - p_i)| \tag{5.8}$$

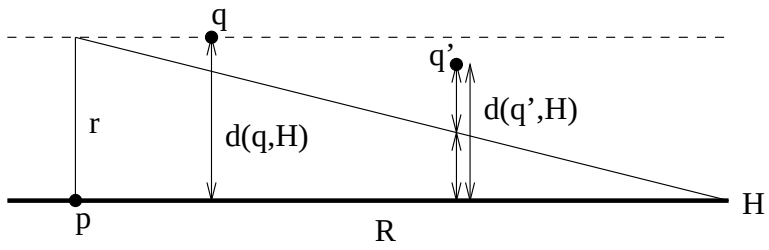


Figura 5.10: Elección de los radios de una banda o hiperplano en R^2.

$$d(p\prime, H^{\perp}) = \sqrt{||p\prime - p||^2 - d(p\prime, H)^2} = \sqrt{\sum_{i=1}^{N}(p\prime_i - p_i)^2 - \left(\sum_{i=1}^{N} v_i(p\prime_i - p_i)\right)^2} \tag{5.9}$$

donde $|.|$ es la función valor absoluto, y $||p\prime - p||$ es la norma (o módulo) del vector que va del punto p al $p\prime$. Con lo que la ecuación (5.7) puede reescribirse como

$$d_{r,R}(p\prime, H) = \frac{|\sum_{i=1}^{N} v_i(p\prime_i - p_i)|}{r} + \frac{\sqrt{\sum_{i=1}^{N}(p\prime_i - p_i)^2 - \left(\sum_{i=1}^{N} v_i(p\prime_i - p_i)\right)^2}}{R} \tag{5.10}$$

Después de estudiar y probar distintas alternativas, hemos decidido elegir el par de valores $< r, R >$ para cada punto de acuerdo al algoritmo que se muestra en la figura 5.11.

En primer lugar elegimos r de forma conservadora, sin incluir puntos de otras clases. Si hay puntos de otras clases a distancia (Euclídea) menor que ε, entonces se fija r como la distancia del punto más cercano de otra clase hasta el Hiperplano, en otro caso se fija $r = \frac{\varepsilon}{4}$.

Después agrandamos R, pero sin incluir puntos de otras clases. Si hay puntos de otras clases que están dentro de la banda infinita de radio r, entonces se calcula el valor de R para que el cono o rombo pase justo por la mitad de la distancia al Hiperplano de ese punto, en otro caso la banda tiene longitud infinita.

En la figura 5.10 se muestra gráficamente con un ejemplo cómo se elegirían los radios r y R de una banda o hiperplano en R^2.

Dado un conjunto de puntos P, un punto $p \in P$, un hiperplano H que pasa por p, una constante $\varepsilon \in \mathrm{R}_0^+$ y la clase de cada uno de los puntos.

Elegimos los valores r y R de acuerdo al siguiente algoritmo:

// Elegimos r de forma conservadora, para no incluir puntos de otras clases.

si $\exists q \in P$ tal que $d(q,p) \leq \varepsilon$ y clase(p)$\neq$clase(q)

entonces se selecciona $q \in P$ con distancia $d(q,p)$ menor tal que clase(p)$\neq$clase(q), y se fija $r = d(q,H)$.

en_otro_caso se fija $r = \frac{\varepsilon}{4}$

fin_si

// Agrandamos R de forma conservadora, sin incluir puntos de otras clases.

si $\exists q\prime \in P$ tal que $d(q\prime,H) < r$ y clase(p)$\neq$clase($q\prime$)

entonces se selecciona $q\prime \in P$ con distancia $d(q\prime,H)$ menor tal que clase(p)$\neq$clase($q\prime$), y se fija $R = \frac{r\ d(q\prime,H)}{r - \frac{d(q\prime,H)}{2}}$

en_otro_caso se fija $R = \infty$

fin_si

Figura 5.11: Algoritmo de elección de los radios r y R de las bandas

5.2 Los experimentos

Se ha usado un gran número de bases de casos para estudiar el comportamiento de los diferentes clasificadores, y se han realizado las pruebas usando 10 Validación Cruzada (10–CV) [WK91]. Se han probado todos los clasificadores con exactamente los mismos ejemplos y se ha realizado una prueba *t*–Student pareada con dos colas con un nivel de significación del 95% para comparar los resultados de los distintos clasificadores.

Para realizar la clasificación usando la distancia basada en las bandas se ha utilizado el método de ajuste de hiperplanos para problemas multiclase definido en (5.6). Aquí el aprendizaje de las bandas o hiperplanos tiene dos parámetros que controlan su comportamiento: ε y F. Con el primero de ellos se controla la localidad del método de aprendizaje, para permitir desde un aprendizaje muy local que casi únicamente tiene en cuenta los puntos más cercanos, hasta un aprendizaje global que también tiene en cuenta los puntos bastante alejados. Con el parámetro F se controla el grado en que se quiere evitar que el método seleccione direcciones donde se encuentran puntos de otras clases, y permite desde ignorar los puntos de otras clases hasta intentar evitarlos a toda costa.

Los métodos de clasificación que se han probado con la distancia de las bandas son 1–NN y *k*–NN. Con este último método necesitamos estimar también el valor del parámetro *k*, que controla el número de puntos más cercanos que se tienen en cuenta para realizar la clasificación. Para completar el estudio hubiese sido interesante analizar el comportamiento de los métodos de los ε–entornos introducidos en el capítulo anterior cuando emplean como distancia base la distancia basada en bandas. Pero por falta de tiempo no ha sido posible realizar estas pruebas, por lo que los resultados de que disponemos son con los métodos 1–NN y *k*–NN. En este sentido, a la hora de analizar el comportamiento de las distancias basadas en bandas frente a las clásicas de la Geometría, las comparaciones más indicativas serán las que realicemos con los resultados obtenidos por los métodos 1–NN y *k*–NN en las pruebas realizadas en el capítulo anterior.

Para estimar el valor de los parámetros se ha realizado una prueba 10–CV con cada conjunto de entrenamiento. Para el aprendizaje de la distancia basada en ban-

das se han considerado los valores 0, 1, 2, 4, 6, 8, 10, 15 para el parámetro F, y distintos valores reales para el parámetro ε, dependiendo de la base de casos. Así por ejemplo, en casi todas las bases sintéticas se han empleado para ε los valores 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1; y en muchas de las bases del UCI–Repository se han empleado los valores 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2. Pero el límite, así como los valores concretos probados, varían de una base a otra.

Para facilitar la comparación de resultados, en los clasificadores k–NN se han considerado los valores impares 1, 3, 5,..., 49 para el parámetro k, los mismos valores usados en las pruebas del capítulo anterior.

En resumen, se ha probado un gran número de combinaciones y se ha empleado una gran cantidad de tiempo para obtener estos resultados.

5.3 Los Resultados

Se han usado varias formas de comparar el comportamiento de los algoritmos para evitar que un punto de vista particular pudiera llevar a conclusiones erróneas. Por una parte, hemos calculado para cada clasificador su acierto medio, mejora del acierto respecto del método k–NN básico, y el acierto medio respecto del mejor clasificador a lo largo de todas las bases de casos (tabla 5.12), y el acierto medio desglosado según el tipo de base de casos (tablas 5.5 y 5.13. Para facilitar la comparación entre clasificadores, también hemos incluido columnas con la posición relativa de cada clasificador de acuerdo a cada una de estas medidas.

Por otra parte, se ha aplicado una comparación pareada entre clasificadores usando un test t–Student con dos colas para construir un intervalo con un 95% de confianza para la diferencia en los porcentajes de acierto de los algoritmos (tablas 5.1, 5.9, 5.10, y 5.11). En adelante, cuando use el término diferencia estadísticamente significativa, o simplemente diferencia significativa, nos referiremos a que la diferencia es estadisticamente significativa de acuerdo a este test t–Student.

Tabla 5.1: Comparación pareada de diferencias estadísticamente significativas entre clasificadores 1-NN con las distancias básicas. Cada celda contiene respectivamente el número de victorias, empates y derrotas estadisticamente significativas entre el método de esa fila y el método de esa columna.

	Suma C	Suma	Max C	Max	Eucl C	Eucl
bandas	36–24–8	29–36–3	38–23–7	32–35–1	33–31–4	28–36–4
Eucl	24–33–11	2–65–1	25–33–10	14–54–0	11–44–10	
Eucl C	22–37–9	9–42–17	26–36–6	18–37–13		
Max	21–30–17	0–55–13	23–28–17			
Max C	1–56–11	11–31–26				
Suma	23–33–12					

5.3.1 Resultados de 1–NN con la distancia de las bandas

En la tabla 5.1 se muestra la comparación pareada usando el test *t*–Student con clasificadores 1–NN usando las medidas de distancia típicas de la Geometría y la nueva distancia basada en bandas que se ha introducido en este capítulo. En las tablas 5.2 y 5.3 se desglosa esta comparación en las bases del UCI–Repository y las sintéticas.

En estas tres tablas puede observarse claramente cómo el método 1–NN cuando usa la distancia de las bandas mejora significativamente con mucha frecuencia los resultados que obtiene con el resto de distancias incluídas en el estudio. Este hecho puede observarse independientemente de la categoría que se considere: UCI–Repository, Sintéticas y Todas.

En la tabla 5.4 se muestra para cada clasificador su acierto medio y la mejora del acierto respecto del método 1–NN con la distancia Euclídea, a lo largo de todas las bases de casos. Para facilitar la comparación entre clasificadores, también se ha incluido una columna con la posición relativa de cada clasificador. Puede observase cómo el método 1–NN obtiene claramente el mejor acierto medio con la distacia de las bandas.

Tabla 5.2: Comparación pareada de diferencias estadísticamente significativas entre clasificadores 1-NN con las distancias básicas en las bases de casos del UCI–Repository. Cada celda contiene respectivamente el número de victorias, empates y derrotas estadisticamente significativas entre el método de esa fila y el método de esa columna.

	Suma C	Suma	Max C	Max	Eucl C	Eucl
bandas	6–11–2	8–9–2	9–9–1	10–8–1	6–10–3	7–10–2
Eucl	2–15–2	1–18–0	4–13–2	9–10–0	1–16–2	
Eucl C	3–14–2	2–16–1	5–14–0	9–10–0		
Max	1–10–8	0–10–9	2–9–8			
Max C	0–13–6	2–12–5				
Suma	2–14–3					

Tabla 5.3: Comparación pareada de diferencias estadísticamente significativas entre clasificadores 1-NN con las distancias básicas en las bases de casos sintéticas. Cada celda contiene respectivamente el número de victorias, empates y derrotas estadisticamente significativas entre el método de esa fila y el método de esa columna.

	Suma C	Suma	Max C	Max	Eucl C	Eucl
bandas	30–13–6	21–27–1	29–14–6	22–27–0	27–21–1	21–26–2
Eucl	22–18–9	1–47–1	21–20–8	5–44–0	13–28–8	
Eucl C	19–23–7	7–26–16	21–22–6	9–27–13		
Max	20–20–9	0–45–4	21–19–9			
Max C	1–43–5	9–19–21				
Suma	21–19–9					

Tabla 5.4: Acierto de los clasificadores en las pruebas con todas las bases de casos: acierto medio y mejora frente a la distancia Euclídea. "Pos." indica la posición relativa de cada clasificador. 1–NN con la distacia de las bandas logra la mejor posición.

Clasificador	Acierto Medio	Δ vs Eucl	Pos.
Euclídea	83.81%	0.00%	3
Euclídea con Correlación	83.53%	-0.28%	4
Máximo	80.65%	-3.16%	6
Máximo con Correlación	80.79%	-3.02%	7
Suma	83.83%	$+0.02\%$	2
Suma con Correlación	82.12%	-1.69%	5
Distancia Bandas	86.62%	$+2.81\%$	1

Tabla 5.5: Acierto de las distancias con un clasificador 1–NN en las pruebas con las bases de casos del UCI–Repository, Sintéticas y todas las bases de casos. "Pos." indica la posición relativa de cada clasificador de acuerdo a cada medida. La distancia basada en bandas logra unos resultados muy buenos.

	UCI		Sintéticas		Todas	
Clasificador	Media	Pos.	Media	Pos.	Media	Pos.
Euclídea	80.18%	4	85.22%	3	83.81%	3
Euclídea con Correlación	81.05%	2	84.49%	5	83.53%	4
Máximo	70.34%	7	84.64%	4	80.65%	6
Máximo con Correlación	77.86%	6	81.93%	7	80.79%	7
Suma	80.17%	5	85.25%	2	83.83%	2
Suma con Correlación	80.93%	3	82.59%	6	82.12%	5
Distancia Bandas	82.54%	1	88.20%	1	86.62%	1

Tabla 5.6: Número de bases en que la Heurística del capítulo anterior elige cada distancia con las bases de casos del UCI–Repository, Sintéticas y todas las bases de casos. La distancia basada en bandas logra los mejores resultados en más de la.

Clasificador	UCI	Sint.	Todas
Euclídea	4	6	10
Euclídea con Correlación	4	0	4
Máximo	0	1	1
Máximo con Correlación	1	6	7
Suma	3	9	12
Suma con Correlación	3	6	9
Distancia Bandas	12	27	39

En la tabla 5.5 se muestra el acierto medio de estos clasificadores desglosado según el tipo de base de casos de que se trate (UCI–Repository, Sintéticas y Todas). Puede observarse cómo con esta medida la distancia basada en bandas obtiene la primera posición en todas las categorías. En las bases del UCI–Repository logra un 1.49% más que la distancia Euclídea con Correlación. En las bases de casos sintéticas logra superar en casi un 3% a la segunda medida de distancia, la de la Suma. Globalmente logra ser la mejor distancia, un 2.81% mejor que la segunda distancia (Suma), por lo que sin información a priori sobre la base de casos es una distancia bastante recomendable.

En la tabla 5.6 se muestra el número de bases en que la heurística propuesta en la sección 4.3.3 elegiría cada una de las distancias básicas. Es decir, se muestra el número de bases en que cada una de las distancias con el método de clasificación 1–NN, y considerando sólo el conjunto de entrenamiento, logra los mejores resultados, de entre todas las distancias consideradas. Se muestra de forma detallada, por columnas, el número de bases de casos del UCI–Repository, Sintéticas y de todas las bases de casos, en las que logra ser la mejor distancia. En este punto me permito recordar que en total se han considerado 68 bases de casos: 19 del UCI–Repository y 49 sintéticas. La distancia basada en bandas exhibe el mejor comportamiento con diferencia: logra los mejores resultados de este conjunto de distancias en más

de la mitad de las bases de casos, tanto globalmente como en los apartados UCI–Repository y sintéticas. Por lo tanto, esta tabla vuelve a indicar que sin información a priori sobre la base de casos es recomendable utilizar la distancia basada en bandas.

En las tablas 5.7[1] y 5.8 puede observarse de manera detallada los resultados que han obtenido cada uno de los clasificadores 1–NN con estas distancias en cada una de las bases de casos utilizadas en la comparativa. Los resultados aparecen seguidos por un "$+$" (o un "$-$") si muestran un mejora (o degradación) estadísticamente significativa sobre la distancia Euclídea, de acuerdo con un test pareado *t*–Student con dos colas con un nivel de confianza del 95%.

Analizando estas tablas, cabe destacar que la distancia basada en bandas logra unos resultados muy superiores al resto de distancias en las bases de casos Tic–Tac–Toe (+21.92%) y WaveForm–21 (+8.10%), y bastante inferiores en Granada Digits (-10.80%), Led 24 (-9.16%) y Letter Recognition (-8.20%). Aunque globalmente la distancia basada en bandas obtiene en las bases de casos del UCI–Repository la primera posición, por delante de la distancia Euclídea con Correlación y Suma con Correlación (tablas 5.2 y 5.5). En la tabla 5.6 se muestra que la distancia basada en bandas logra los mejores resultados en 12 de las 19 bases del UCI–Repository. Cuando existen muchos atributos irrelevantes la distancia basada en bandas sufre una degradación superior a las distancias que usan Correlación (véase los resultados con las bases de casos Led 7 y Led 24, y WaveForm–21 y WaveForm–40), y una degradación muy inferior a las distancias que no usan Correlación. Por tanto, si la base de casos tiene atributos irrelevantes es especialmente importante usar de manera previa algún método para eliminarlos, y posteriormente realizar la clasificación propiamente dicha.

[1] A pesar del hecho de que los resultados con la base de casos Cleveland pudieran parecer anormalmente bajos, se debe tener en cuenta que estos resultados se han obtenido considerando 5 clases, mientras que los experimentos con esta base normalmente se han concentrado en distinguir simplemente la presencia (valores 1,2,3,4) de la ausencia (valor 0) de enfermedad de corazón.

Tabla 5.7: Resultados del método 1–NN con las medidas de distancias básicas y la distancia basada en bandas en las bases de casos del UCI–Repository. "+"/"−" representa mejora/degradación estadísticamente significativa sobre la distancia Euclídea.

	Eucl	Eucl C	Max	Max C	Suma	Suma C	Bandas
IR	95.33%	96.00%	96.00%	94.67%	94.00%	94.67%	96.67%
WI	94.94%	96.07%	94.94%	92.70%	95.51%	96.63%	98.31%
PI	69.92%	70.83%	68.36%	70.57%	69.53%	68.36%	72.79%+
GL	70.09%	68.69%	66.82%	65.89%	73.36%	72.90%	67.29%
CL	52.48%	54.13%	50.83%	55.12%	52.48%	56.44%	57.43%+
GD	96.70%	96.40%	64.60%−	73.30%−	96.30%	95.10%−	85.90%−
SN	87.02%	88.94%	79.33%−	84.62%	86.54%	87.50%	88.46%
LD	61.74%	60.00%	57.97%	57.39%	61.16%	60.00%	66.38%
ZO	96.04%	97.03%	74.26%−	91.09%−	96.04%	97.03%	96.04%
TT	75.57%	74.84%	56.37%−	76.51%	75.57%	76.72%	98.64%+
L7	60.02%	60.00%	59.88%−	60.00%	60.02%	60.00%	59.92%
L24	48.98%	63.08%+	10.20%−	62.98%+	48.98%	63.08%+	53.92%+
W21	77.06%	76.22%	76.38%	74.78%−	76.62%	75.86%	85.16%+
W40	73.24%	77.82%+	68.04%−	75.70%+	73.22%	76.98%+	83.52%+
SO	100.00%	97.87%	53.19%−	97.87%	100.00%	97.87%	100.00%
F1	73.55%	73.64%	73.26%	73.73%	73.64%	73.55%	74.77%+
F2	95.50%	95.40%	95.40%	95.40%	95.50%	95.31%	95.97%
F3	99.25%	99.34%	99.16%	99.34%	99.25%	99.34%	99.34%
LR	96.02%	93.65%−	91.55%−	77.72%−	95.49%−	90.32%−	87.82%−

Tabla 5.8: Resultados del método 1–NN con las medidas de distancias básicas y la distancia basada en bandas en las bases de casos sintéticas. "+"/"−" representa mejora/degradación estadísticamente significativa sobre la distancia Euclídea.

	Eucl	Eucl C	Max	Max C	Suma	Suma C	Bandas
bandas 5u	94.40%	97.00%+	94.00%	98.60%+	95.20%	98.60%+	96.20%
bandas 5m	94.60%	97.40%+	94.40%	99.00%+	95.20%	98.40%+	95.60%
bandas 5p	93.60%	96.00%	93.60%	97.20%	93.40%	97.60%+	96.20%
bandas 10u	87.80%	90.80%+	86.80%	94.80%+	87.00%	95.00%+	92.20%+
bandas 10m	88.60%	94.40%+	88.00%	96.80%+	89.00%	96.60%+	94.60%+
bandas 10p	85.80%	91.80%+	85.00%	93.80%+	86.60%	94.60%+	93.80%+
bandas 20u	72.60%	86.40%+	70.80%	92.20%+	74.00%	92.40%+	89.00%+
bandas 20m	71.60%	82.60%+	67.60%−	90.60%+	71.80%	90.00%+	87.80%+
bandas 20p	71.60%	83.60%+	69.60%	89.00%+	70.80%	90.20%+	85.60%+
Gauss	82.60%	82.40%	81.40%	80.00%	82.20%	78.20%−	84.60%
anillo a 3u	89.40%	89.40%	89.00%	88.80%	89.80%	89.80%	92.80%+
anillo a 3m	91.20%	90.60%	91.00%	84.80%−	91.20%	86.40%−	93.20%+
anillo a 3p	92.80%	93.00%	93.00%	91.80%	92.60%	90.00%−	94.40%
anillo a 6u	80.80%	78.20%	79.40%−	73.60%−	81.80%	74.40%−	89.60%+
anillo a 6m	79.80%	79.40%	79.80%	80.00%	80.40%	80.00%	87.00%+
anillo a 6p	80.60%	80.00%	80.20%	79.80%	80.80%	82.00%	87.60%+
anillo a 9u	70.00%	70.00%	70.80%	70.00%	68.80%	69.40%	77.40%+
anillo a 9m	70.60%	67.00%−	70.20%	52.40%−	70.40%	54.80%−	75.80%+
anillo a 9p	72.60%	61.60%−	69.20%−	52.00%−	69.80%−	55.20%−	79.80%+
anillo r 3u	94.80%	94.80%	95.20%	94.60%	94.20%	94.00%	97.40%+
anillo r 3m	94.40%	94.60%	95.20%	95.60%	94.40%	94.80%	96.00%
anillo r 3p	94.20%	94.40%	93.80%	94.20%	94.60%	94.40%	95.20%
anillo r 6u	86.40%	75.80%−	86.20%	51.80%−	86.40%	56.00%−	89.80%+
anillo r 6m	89.80%	89.20%	89.80%	89.20%	89.60%	88.40%	92.60%+
anillo r 6p	84.00%	84.60%	84.00%	84.80%	83.40%	83.40%	87.20%+
anillo r 9u	80.20%	80.20%	79.20%	77.80%−	81.20%	76.80%−	84.80%+
anillo r 9m	78.20%	77.80%	79.20%	75.40%	78.20%	76.80%	83.40%+
anillo r 9p	77.00%	78.20%	77.40%	76.20%	78.20%	75.80%	81.20%+
senos 3u	92.60%	91.40%	91.80%−	88.80%−	92.00%	89.20%−	91.20%

(continúa en la página siguiente)

Tabla 5.8: Resultados del método 1–NN con las medidas de distancias básicas y la distancia basada en bandas en las bases de casos sintéticas. "+"/"−" representa mejora/degradación estadísticamente significativa sobre la distancia Euclídea.

	Eucl	Eucl C	Max	Max C	Suma	Suma C	Bandas
senos 3m	90.60%	88.60%−	90.60%	86.80%−	90.60%	87.60%−	91.60%
senos 3p	91.60%	88.80%−	91.60%	87.20%−	92.20%	88.20%−	92.20%
senos 6u	85.80%	73.80%−	85.40%	62.20%−	86.60%	65.80%−	86.60%
senos 6m	83.60%	78.80%−	82.80%	74.00%−	84.20%	75.60%−	84.00%
senos 6p	83.80%	71.00%−	82.20%	63.40%−	84.20%	67.00%−	85.80%
senos 9u	75.80%	67.80%−	75.00%	58.80%−	74.40%	63.20%−	76.80%
senos 9m	78.40%	72.00%−	76.60%	62.80%−	78.60%	67.20%−	76.40%
senos 9p	76.60%	70.40%−	76.00%	63.60%−	76.60%	65.40%−	77.40%
cuad. 2u	96.80%	95.40%	96.40%	89.40%−	97.00%	88.80%−	97.00%
cuad. 2m	96.20%	95.80%	96.00%	96.20%	96.20%	95.60%	95.80%
cuad. 2p	98.60%	98.60%	98.60%	97.20%	98.80%	97.60%	98.60%
cuad. 4u	93.20%	91.40%−	93.00%	88.20%−	93.40%	87.80%−	92.40%
cuad. 4m	93.80%	90.80%−	92.40%−	81.40%−	93.60%	82.60%−	93.00%−
cuad. 4p	90.20%	90.80%	90.00%	91.00%	90.60%	90.60%	90.40%
cuad. 6u	85.80%	86.20%	86.40%	80.80%−	85.60%	82.40%−	86.60%
cuad. 6m	86.40%	85.40%	87.00%	85.20%	85.80%	84.80%	85.20%
cuad. 6p	84.60%	82.80%	83.60%	83.20%	84.40%	82.60%	82.80%
cuad. 8u	76.60%	75.60%	75.60%	69.40%−	75.80%	68.60%−	75.40%
cuad. 8m	80.80%	80.80%	79.80%	79.40%	82.40%+	81.00%	79.00%−
cuad. 8p	84.00%	82.40%	82.80%	80.80%−	83.20%	81.20%	82.60%

Respecto de las bases de casos sintéticas, la distancia basada en bandas logra globalmente los mejores resultados con diferencia (tabla 5.5). Cabe destacar que logra los mejores resultados en la base de casos de Gauss, en todas las bases de casos de los anillos con radio constante y anillos con área constante, y en la mayoría de las bases de casos de los senos. La explicación de estos resultados es debida a que localmente se puede aproximar mediantes rectas cualquier figura en el plano, y por ejemplo en las bases de casos de los anillos, se están aproximando localemnte esos anillos mediante bandas. Y desde luego, a juzgar por los resultados lo hace

relativamente bien.

Como era previsible, en las bases de casos de los cuadrados obtiene peores resultados, aunque la diferencia tampoco es demasiado grande con el resto de las distancias, y la distancia basada en bandas logra el mejor resultado en dos variantes de los cuadrados. Con las bases de casos de los cuadrados está claro que una banda no es la mejor forma de aproximar un cuadrado, aunque como ya he comentado antes, la diferencia con el resto de distancias es relativamente pequeña.

Sorprendentemente, en las bases de casos de las bandas, la distancia basada en bandas no logra el mejor resultado en ninguna de las variantes, ya que el primer puesto lo logran las distancias de la Suma con Correlación en las variantes "Uniforme" y "Progresivo" la distancia del Máximo con Correlación en las variantes "Mitad". En general suelen obtener mejores resultados las distancias que emplean información sobre la correlación de los atributos con la clase, y los peores resultados los suelen obtener las distancias que no emplean información sobre la correlación. A pesar de todo la distancia basada en bandas logra unos resultados cercanos a los mejores.

Para explicar estos resultados con las bandas debemos tener en cuenta que en la práctica las bandas suelen ser casi paralelas al eje X, pero con una ligera inclinación, ya que se calculan a partir de una serie de puntos concretos que hay en la zona, y no de la banda original, y por lo tanto suele haber una ligera desviación al buscar los puntos de la misma clase. A esto hay que unir que a veces se encuentran mucho más cerca puntos de otras clases que pertenecen a bandas vecinas.

Pero sobre todo, hay que tener en cuenta que las distancias que emplean la Correlación del atributo con la clase obtienen mejores resultados porque con la correlación se detecta mejor que con las bandas que el atributo x es irrelevante (ya hemos comentado las pequeñas desviaciones de las bandas). Por tanto, al ser las bandas paralelas a uno de los ejes nos encontramos en la situación más favorable para las distancias que emplean información sobre la correlación del atributo con la clase.

Si las bandas no fueran paralelas a uno de los ejes, la distancia basada en bandas seguiría identificando bien la dirección de las bandas y obtendría unos resultados

Tabla 5.9: Comparación pareada de diferencias estadísticamente significativas entre clasificadores. Cada celda contiene respectivamente el número de victorias, empates y derrotas estadisticamente significativas entre el método de esa fila y el método de esa columna. *k*–NN con la distancia de las bandas mejora significativamente a los métodos C, E, F y F1 ligeramente, y al resto de los métodos frecuentemente.

	F1	F	E	D	C	B	A
bandas	16–39–13	15–40–13	17–38–13	23–39–6	14–43–11	25–37–6	27–36–5
A	0–53–15	0–53–15	0–53–15	0–63–5	1–53–14	1–62–5	
B	2–55–11	2–55–11	2–55–11	0–67–1	6–48–14		
C	3–58–7	3–58–7	3–58–7	14–47–7			
D	2–55–11	2–55–11	2–55–11				
E	0–68–0	0–68–0					
F	0–68–0						

similares a los que obtiene en estas pruebas, pero las distancias que emplean la Correlación no considerarían un atributo más importante que otro y sufrirían una degradación en sus resultados hasta situarlos en niveles similares o inferiores a los que obtienen las distancias que no emplean correlación. Nos encontraríamos en la situación extrema si las bandas tuvierann una inclinación de 45^o con respecto a cualquier eje, y la correlación del atributo con la clase no proporcionaría ventaja adicional a las distancias. En esa situación la distancia de las bandas obtendría los mejores resultados en todas las variantes, y por ejemplo en las variantes Bandas 20 llegaría a mejoras superiores al 15%.

5.3.2 Resultados de *k*–NN con la distancia de las bandas

En las tablas 5.9, 5.10 y 5.11 se muestra la comparación pareada usando el test *t*–Student con el método de clasificación *k*–NN con la distancia basada en bandas, y con los clasificadores usados en el capítulo anterior (tabla 4.1). Puede observarse cómo el comportamiento de *k*–NN con la distancia de las bandas es clara-

Tabla 5.10: Comparación pareada de diferencias estadísticamente significativas entre clasificadores con las bases de casos del UCI–Repository. Cada celda contiene respectivamente el número de victorias, empates y derrotas estadisticamente significativas entre el método de esa fila y el método de esa columna. *k*–NN con la distancia de las bandas se encuentra al mismo nivel que los métodos C, y F1 desde un punto de vista de significación estadística, y un poco por debajo del método F.

	F1	F	E	D	C	B	A
bandas	4–12–3	3–13–3	5–11–3	3–13–3	3–14–2	4–12–3	5–12–2
A	0–16–3	0–16–3	0–16–3	0–16–3	1–17–1	1–15–3	
B	0–19–0	0–19–0	0–19–0	0–18–1	3–15–1		
C	1–14–4	1–14–4	1–14–4	1–14–4			
D	0–19–0	0–19–0	0–19–0				
E	0–19–0	0–19–0					
F	0–19–0						

Tabla 5.11: Comparación pareada de diferencias estadísticamente significativas entre clasificadores con las bases de casos sintéticas. Cada celda contiene respectivamente el número de victorias, empates y derrotas estadisticamente significativas entre el método de esa fila y el método de esa columna. *k*–NN con la distancia de las bandas mejora significativamente a los métodos C, E, F y F1 ligeramente, y al resto de los métodos frecuentemente.

	F1	F	E	D	C	B	A
bandas	12–27–10	12–27–10	12–27–10	20–26–3	11–29–9	21–25–3	22–24–3
A	0–37–12	0–37–12	0–37–12	0–47–2	0–36–13	0–47–2	
B	2–36–11	2–36–11	2–36–11	0–49–0	3–33–13		
C	2–44–3	2–44–3	2–44–3	13–33–3			
D	2–36–11	2–36–11	2–36–11				
E	0–49–0	0–49–0					
F	0–49–0						

Tabla 5.12: Acierto de los clasificadores en las pruebas con todas las bases de casos: acierto medio y mejora frente a *k*–NN. "Pos." indica la posición relativa de cada clasificador. *k*–NN con la distacia de las bandas logra la mejor posición.

	Clasificador	Acierto Medio	Δ vs *k*–NN	Pos.
A	*k*–NN	85.65%	0.00%	7
B	ε–ball	85.75%	+0.10%	6
C	*k*–NN Heur	87.30%	+1.65%	3
D	ε–ball^{k-NN}	85.88%	+0.23%	5
E	ε–ball Heur	87.29%	+1.64%	4
F	ε–ball^{k-NN} Heur	87.39%	+1.74%	2
	k–NN Distacia Bandas	87.96%	+2.31%	1

mente superior a los métodos A (*k*–NN), B (ε–entorno) y D (ε–entorno^{k-NN}), y sólo mejora con más frecuencia a los métodos C (*k*–NN Heur), E (ε–entorno Heur), F (ε–entorno^{k-NN} Heur), y F1 (ε–entorno^{1-NN} Heur).

El alto número de bases de casos en que el método *k*–NN con la distancia de las bandas es significativamente mejor y peor con respecto a los métodos C, E y F, parece sugerir que las bases de casos en que es mejor aplicar uno u otro tipo de métodos son diferentes. Por lo que se podría construir un clasificador que realice una evaluación sobre el conjunto de casos de entrenamiento de cual de los métodos (por ejemplo *k*–NN con bandas y F1) obtiene mejores resultados, y emplear entonces ese método para realizar la clasificación.

En la tabla 5.12 se muestra para cada clasificador su acierto medio y la mejora del acierto respecto del método *k*–NN básico a lo largo de todas las bases de casos. Para facilitar la comparación entre clasificadores, también se ha incluido una columna con la posición relativa de cada clasificador. Puede observase cómo *k*–NN con la distacia de las bandas obtiene el mejor acierto medio.

En la tabla 5.13 se muestra para cada clasificador su acierto medio desglosado en cada una de las tres categorías o grupos de bases de casos que estamos distinguiendo en este trabajo (UCI–Repository, Sintéticas y Todas). Para facilitar la comparación entre clasificadores, también se ha incluido una columna con la posición relativa de

Tabla 5.13: Acierto de los clasificadores en las pruebas con las bases de casos del UCI–Repository, Sintéticas y todas las bases de casos. "Pos." indica la posición relativa de cada clasificador en cada apartado. *k*–NN con la distacia de las bandas logra la mejor posición en todos los apartados.

		UCI		Sintéticas		Todas	
	Clasificador	Media	Pos.	Media	Pos.	Media	Pos.
A	*k*–NN	85.03%	6	85.89%	7	85.65%	7
B	ε–ball	84.95%	7	86.06%	6	85.75%	6
C	*k*–NN Heur	85.12%	5	88.14%	2	87.30%	2
D	ε–ball^{k-NN}	85.33%	3	86.09%	5	85.88%	5
E	ε–ball Heur	85.14%	4	88.11%	4	87.29%	4
F	ε–ball^{k-NN} Heur	85.48%	2	88.13%	3	87.39%	2
	k–NN Distacia Bandas	85.70%	1	88.83%	1	87.96%	1

cada clasificador en cada una de estas categorías. Puede observarse que *k*–NN con la distacia de las bandas logra la mejor posición en todos los apartados, aunque la mejora que obtiene es superior en las bases de casos sintéticas y más ajustada en las del UCI–Repository. Por lo que, sin información a priori sobre la base de casos, *k*–NN con la distancia de las bandas es un método de clasificación recomendable, aunque se debe tener precaución con el tipo de base de casos donde se va a aplicar.

En la tabla 5.14 se muestra el número de bases en que cada uno de los métodos de clasificación obtiene los mejores resultados con las bases de casos del UCI–Repository, Sintéticas y todas las bases de casos. Es decir, se muestra el número de bases en que cada uno de los métodos de clasificación logra los mejores resultados, de entre los alcanzados por todos los métodos considerados. En este punto me permito recordar que en total se han considerado 68 bases de casos: 19 del UCI–Repository y 49 sintéticas. El método de los *k*–NN con la distancia basada en bandas exhibe globalmente el mejor comportamiento: logra los mejores resultados de este conjunto de clasificadores en aproximadamente la mitad de las bases de casos, tanto globalmente como en los apartados UCI–Repository y sintéticas. Por lo tanto, esta tabla vuelve a indicar que, sin información a priori sobre la base de

Tabla 5.14: Número de bases en que se obtienen los mejores resultados con cada uno de los métodos de clasificación en las bases de casos del UCI–Repository, Sintéticas y todas las bases de casos.

	Clasificador	UCI	Sint.	Todas
A	*k*–NN	4	4	8
B	ε–ball	8	9	17
C	*k*–NN Heur	6	12	18
D	ε–ball^{k-NN}	10	9	19
E	ε–ball Heur	7	11	18
F	ε–ball^{k-NN} Heur	9	11	20
	k–NN Distacia Bandas	9	24	33

casos, es recomendable utilizar el método *k*–NN con la distancia basada en bandas.

Las tablas 5.15[2] y 5.16 muestran de manera detallada los resultados logrados con cada base de casos por el método de clasificación *k*–NN con la distancia basada en bandas, y por los clasificadores usados en el capítulo anterior: A (*k*–NN), B (ε–ball), C (*k*–NN Heur), D (ε–ball^{k-NN}), E (ε–ball Heur) y F (ε–ball^{k-NN} Heur). Los resultados aparecen seguidos por un "$+$" (o un "$-$") si muestran un mejora (o degradación) estadísticamente significativa sobre el método A (*k*–NN básico), de acuerdo con un test pareado *t*–Student con dos colas con un nivel de confianza del 95%.

Analizando estas tablas, cabe destacar que *k*–NN con la distancia basada en bandas logra unos resultados bastante superiores al resto de métodos en las bases de casos Tic–Tac–Toe (+14.51%) y Liver Disorder (6.37%) , y bastante inferiores en Letter Recognition (-8.12%) y Glass (-6.07%). Globalmente *k*–NN con la distancia basada en bandas obtiene en las bases de casos del UCI–Repository el mejor porcentaje medio de acierto (tabla 5.13), y en la tabla 5.14 se muestra que logra los

[2]A pesar del hecho de que los resultados con la base de casos Cleveland pudieran parecer anormalmente bajos, se debe tener en cuenta que estos resultados se han obtenido considerando 5 clases, mientras que los experimentos con esta base normalmente se han concentrado en distinguir simplemente la presencia (valores 1,2,3,4) de la ausencia (valor 0) de enfermedad de corazón.

Tabla 5.15: Resultados de los clasificadores con las bases de casos del UCI–Repository. "+"/"−" representa mejora/degradación estadísticamente significativa sobre *k*–NN (método A).

	A	B	C	D	E	F	*k*–NN bandas
IR	96.00%	96.00%	96.00%	96.00%	96.00%	96.00%	96.00%
WI	97.19%	97.19%	96.63%	97.19%	95.51%	96.63%	98.88%
PI	75.39%	73.31%	76.30%	73.31%	75.52%	75.52%	76.30%
GL	71.50%	71.03%	72.90%	71.50%	73.36%	75.23%	69.16%
CL	57.10%	58.42%	58.09%	58.42%	57.43%	57.43%	58.09%
GD	96.70%	95.70%	96.70%	96.70%	95.70%	96.70%	93.50%
SN	87.02%	87.02%	88.94%	87.98%	90.87%	90.38%	87.50%
LD	65.22%	63.48%	65.22%	65.51%	63.48%	65.51%	71.88%+
ZO	96.04%	96.04%	97.03%	97.03%	96.04%	97.03%	96.04%
TT	84.24%	82.57%−	78.18%−	84.24%	80.90%	80.90%	98.75%+
L7	74.48%	74.36%	74.48%	74.36%	74.36%	74.36%	74.48%
L24	72.34%	73.80%+	73.74%+	73.80%+	73.52%+	73.52%+	70.14%−
W21	85.42%	85.04%	85.42%	85.04%	85.04%	85.04%	86.64%+
W40	84.54%	84.62%	85.40%	84.62%	84.46%	84.46%	85.86%+
SO	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
F1	80.68%	82.93%+	80.58%	82.93%+	82.93%+	82.93%+	81.05%+
F2	96.34%	96.62%	96.25%	96.62%	96.62%	96.62%	96.44%
F3	99.44%	99.53%	99.44%	99.53%	99.53%	99.53%	99.44%
LR	96.02%	96.42%+	96.02%	96.42%+	96.42%+	96.42%+	88.30%−

mejores resultados en 9 de las 19 bases del UCI–Repository.

La medida en que la presencia de atributos irrelevantes afecta al porcentaje de acierto del método *k*–NN con la distancia basada en bandas depende de la base de casos concreta. Así por ejemplo en las bases de casos Led 7 y led 24 sí afecta claramente la presencia de 17 atributos irrelevantes. En cambio en las bases de casos WaveForm–21 y WaveForm–40 practicamente no afecta la presencia de 19 atributos irrelevantes.

Como conclusión podemos indicar que si la base de casos tiene atributos irrelevantes es recomendable usar previamente algún método para eliminarlos, y así prevenir que afecten negativamente al rendimiento del clasificador. Esto puede realizarse con cualquier método de eliminación de atributos irrelevantes, o bien, como comentaremos en el apartado 5.4, pueden usarse las propiedades de la distancia basada en bandas para proponer una reducción de dimensiones (y eliminación de atributos irrelevantes), y después realizar la clasificación propiamente dicha.

Tabla 5.16: Resultados de los clasificadores con las bases de casos sintéticas. "+"/"−" representa mejora/degradación estadísticamente significativa sobre *k*–NN (método A).

	A	B	C	D	E	F	*k*–NN bandas
bandas 5u	96.20%	96.00%	99.40%+	96.00%	99.20%+	99.20%+	98.40%+
bandas 5h	95.40%	95.80%	98.80%+	95.80%	98.20%+	98.20%+	97.20%
bandas 5p	95.40%	94.80%	98.20%+	94.80%	98.60%+	98.60%+	97.20%+
bandas 10u	88.20%	89.60%	97.00%+	89.60%	96.60%+	96.60%+	94.00%+
bandas 10h	89.80%	88.80%	97.20%+	88.80%	96.80%+	96.80%+	94.60%+
bandas 10p	87.80%	88.40%	95.80%+	88.40%	96.80%+	96.80%+	93.80%+
bandas 20u	72.60%	72.40%	93.80%+	72.60%	94.00%+	94.00%+	89.00%+
bandas 20h	71.60%	72.00%	93.20%+	72.00%	92.20%+	92.20%+	88.40%+
bandas 20p	71.60%	71.40%	91.80%+	71.60%	89.80%+	89.80%+	86.80%+
Gauss	87.60%	88.20%	86.60%	88.20%	86.40%	86.40%	87.80%
anillo a 3u	90.60%	91.40%	93.00%+	91.40%	92.20%	92.20%	94.60%+
anillo a 3h	91.40%	93.00%+	91.40%	93.00%+	93.00%+	93.00%+	95.40%+

(continúa en la página siguiente)

Tabla 5.16: Resultados de los clasificadores con las bases de casos sintéticas. "+"/"−" representa mejora/degradación estadísticamente significativa sobre *k*–NN (método A).

	A	B	C	D	E	F	*k*–NN bandas
anillo a 3p	93.80%	93.60%	93.60%	93.60%	93.80%	93.80%	95.00%
anillo a 6u	80.80%	81.20%	80.80%	81.20%	81.20%	81.20%	89.60%+
anillo a 6h	80.00%	81.20%	80.00%	81.20%	81.20%	81.20%	87.00%+
anillo a 6p	82.80%	82.20%	82.40%	82.20%	81.80%	81.80%	87.60%+
anillo a 9u	70.00%	71.20%	70.00%	71.20%	69.60%	69.60%	77.40%+
anillo a 9h	70.60%	69.80%	70.20%	69.80%	69.80%	70.20%	75.88%+
anillo a 9p	72.60%	72.60%	72.60%	72.60%	72.60%	72.60%	79.80%+
anillo r 3u	95.40%	95.00%	96.20%	95.00%	95.60%	95.60%	97.80%+
anillo r 3h	95.80%	95.00%	95.20%	95.00%	95.60%	95.60%	96.00%
anillo r 3p	94.20%	94.40%	94.60%	94.40%	94.60%	94.60%	95.40%
anillo r 6u	88.00%	88.20%	88.20%	88.20%	88.60%	88.60%	91.20%+
anillo r 6h	89.80%	91.20%+	89.80%	91.20%+	91.20%+	91.20%+	92.60%
anillo r 6p	85.60%	85.60%	84.80%	85.60%	85.80%	85.80%	87.20%
anillo r 9u	82.60%	81.60%	84.40%	81.60%	82.80%	82.80%	86.80%+
anillo r 9h	78.20%	78.60%	79.20%	78.60%	79.60%	79.60%	83.40%+
anillo r 9p	77.00%	77.80%	78.20%	77.80%	79.20%	79.20%	82.80%+
senos 3u	93.40%	92.60%	93.40%	92.60%	92.60%	92.60%	93.20%
senos 3h	91.40%	92.00%	90.60%	92.00%	90.80%	90.80%	92.60%
senos 3p	93.00%	91.80%	94.40%	91.80%	93.80%	93.80%	91.20%
senos 6u	85.80%	86.20%	86.00%	86.20%	86.60%	86.60%	86.00%
senos 6h	83.60%	84.00%	84.20%	84.00%	84.80%	84.80%	83.20%
senos 6p	83.80%	84.40%	84.60%	84.40%	85.00%	85.00%	86.60%
senos 9u	75.80%	75.80%	75.80%	76.00%	75.80%	76.00%	76.80%
senos 9h	78.40%	78.80%	78.60%	78.80%	78.60%	78.60%	78.60%
senos 9p	76.60%	76.40%	79.80%+	77.20%	77.40%	77.40%	80.00%+
cuad. 2u	97.60%	97.20%	97.60%	97.20%	97.20%	97.20%	97.80%
cuad. 2h	96.80%	97.00%	96.80%	97.00%	97.00%	97.00%	96.80%
cuad. 2p	98.20%	98.60%	98.20%	98.60%	98.60%	98.60%	98.60%

(continúa en la página siguiente)

Tabla 5.16: Resultados de los clasificadores con las bases de casos sintéticas. "+"/"−" representa mejora/degradación estadísticamente significativa sobre *k*–NN (método A).

	A	B	C	D	E	F	*k*–NN bandas
cuad. 4u	93.80%	94.40%	94.40%	94.40%	94.20%	94.20%	92.40%−
cuad. 4h	93.80%	93.80%	93.60%	93.80%	93.40%	93.60%	93.00%−
cuad. 4p	93.20%	92.60%	93.60%	92.60%	92.00%	92.00%	91.00%
cuad. 6u	87.20%	87.20%	86.40%	87.20%	87.00%	87.00%	85.00%
cuad. 6h	85.80%	86.80%	87.80%	86.80%	87.80%	87.80%	85.80%
cuad. 6p	84.60%	85.20%	84.60%	85.20%	85.20%	85.20%	84.20%
cuad. 8u	75.80%	76.40%	75.80%	76.40%	76.40%	76.40%	75.80%
cuad. 8h	80.80%	81.00%	82.40%+	81.00%	82.80%+	82.80%+	79.00%−
cuad. 8p	84.00%	83.80%	84.00%	83.80%	83.80%	83.80%	82.60%

Respecto de las bases de casos sintéticas, *k*–NN con la distancia basada en bandas logra globalmente los mejores resultados con diferencia (tablas 5.13 y 5.14). Cabe destacar que logra los mejores resultados en la base de casos de Gauss, en todas las bases de casos de los anillos con radio constante y anillos con área constante, y en la mitad de las bases de casos de los senos. Como era previsible, en las bases de casos de los cuadrados obtiene peores resultados, aunque logra el mejor resultado en la variante más sencilla de los cuadrados (cuadrados 2×2 uniforme). De nuevo en las bases de las bandas no logra el mejor resultado en ninguna de las variantes, ya que los primeros puestos se reparten entre los métodos C (*k*–NN Heur), E (ε–entorno Heur) y F (ε–entorno^{k-NN} Heur), que con la Heurística seleccionan distancias que ponderan la importancia de cada atributo por su correlación con la clase. A pesar de todo la distancia basada en bandas logra situarse por delante del resto de métodos: A (*k*–NN), B (ε–entorno) y D (ε–entorno^{k-NN}), que no emplean la correlación del atributo con su clase.

Para explicar estos resultados con las bandas debemos tener en cuenta que en la práctica a menudo las bandas son casi paralelas al eje X, pero con una ligera inclinación, ya que se calculan a partir de una serie de puntos concretos, y no de la

banda original, y por lo tanto suele haber una ligera desviación al buscar los puntos de la misma clase. A esto hay que unir que a veces se encuentran mucho más cerca puntos de otras clases que de la propia clase.

Pero sobre todo, hay que tener en cuenta que tal y como se comentó al final del apartado 5.3.1, al ser las bandas paralelas a uno de los ejes nos encontramos en la situación más favorable para las distancias que emplean información sobre la correlación del atributo con la clase.

En el otro extremo se encuentra la situación en que las bandas tienen una inclinación de 45^o con respecto a cualquier eje, y la correlación del atributo con la clase no proporciona ventaja adicional a las distancias (no hay ningún atributo más importante que otro). En esa situación sería muy parecido el porcentaje de acierto de las medidas de distancia que emplean información sobre la correlación y el de las que no emplean esa información. En cambio, las distancias de las bandas mantienen su comportamiento independientemente de la dirección concreta de las bandas.

5.4 Transformaciones del problema original mediante bandas

A continuación vamos a comentar cómo se puede utilizar la información que aporta la posición de los puntos y las bandas que éstos aprenden para proponer transformaciones sobre el problema original, principalmente con el objetivo de simplificarlo o incluso de mejorar la legibilidad e interpretación de los resultados que pueda proporcionar.

Básicamente vamos a comentar en qué situaciones y cómo puede realizarse la reducción en una dimensión del problema original mediante una proyección sobre un hiperplano. Así, y empleando sucesivamente este método se puede reducir el número de dimensiones del problema original eliminando atributos irrelevantes o proponiendo una combinación lineal de atributos que expresa el problema original con un número inferior de dimensiones.

Por otra parte vamos a analizar las situaciones en que los puntos se encuentran

distribuídos formando una figura curva alrededor de un centro. Vamos a analizar cómo se puede determinar ese centro de curvatura, y si es conveniente proponer un cambio a coordenadas polares o no. Con este método incluso vamos a poder identificar centros de curvatura o giro de figuras complejas.

También hay que dejar claro que el análisis que se realiza en este apartado debe considerarse como introductorio a las posibilidades que ofrecen las bandas para proponer transformaciones del problema o identificar formas y figuras que forman los puntos. Este estudio debe desarrollarse con mucho más detalle en trabajos posteriores, y debe incluir más posibilidades que la simple proyección o cambio a coordenadas polares. Además estas transformaciones se han explicado sobre problemas de dos dimensiones, pero pueden generalizarse facilmente a problemas N–Dimensionales. Aunque en este caso se debe tener cuidado porque existen muchas más posibilidades que solo en 2–D. Por ejemplo, en 2–D se puede proponer un cambio a coordenadas polares si nos encontramos con que los puntos forman una circunferencia, pero en 3–D se puede proponer un cambio a coordenadas polares si los puntos forman una esfera, o puede ser más interesante un cambio a coordenadas cilíndricas o una proyección si los puntos forman un cilindro. En general, cuando nos enfrentamos a un problema N–dimensional, podemos encontrar subespacios interesantes de diferentes dimensiones sobre los que proponer alguna transformación. Por ejemplo, en un espacio de dimensión N podemos distinguir subespacios de dimensión 1 (p.ej. una recta o una circunferencia), de dimensión 2 (p.ej. un plano, una esfera, o un cilindro de dimensión infinita), de dimensión 3, ..., y así sucesivamente hasta subespacios de dimensión $N-1$ (p.ej. un hiperplano o una hiperesfera).

Después de aplicar cualquiera de estos métodos, se puede volver a intentar aplicar cualquier de ellos. Por ejemplo, primero se puede pasar de un espacio original de dimensión 4 a otro de dimensión 3 proyectando sobre un hiperplano. A continuación se puede realizar otra reducción de dimensiones a un plano de dimensión 2. Después puede ser interesante realizar un cambio a coordenadas polares, y por último tal vez se pueda volver a realizar una reducción de dimensiones para acabar con un problema donde se tenga solamente un atributo. Este ejemplo creo que clarifica bastante la manera en que puede secuenciarse la aplicación de este tipo

de métodos, y aunque pueda parecer rebuscado el problema original podría corresponder con una serie de anillos o cilindros concéntricos que se encuentren en un espacio de dimensión 4 y que no estén alineados con los ejes.

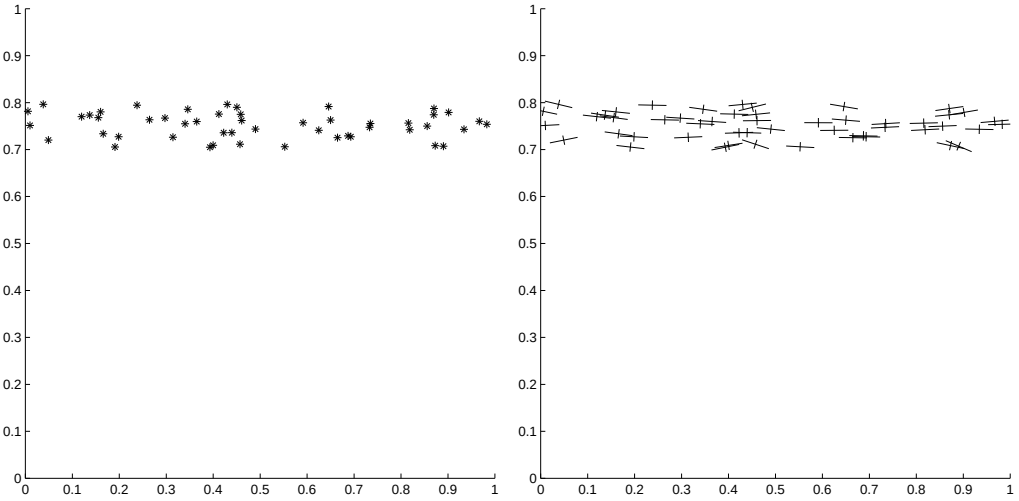
5.4.1 Reducción de dimensiones mediante proyecciones

Si la base de casos tiene atributos irrelevantes es especialmente importante usar de manera previa algún método para eliminarlos, o bien usar la distancia basada en bandas primero para proponer la reducción de dimensiones (y eliminación de atributos irrelevantes), y después para realizar la clasificación propiamente dicha.

Si nos encontramos en una base de casos como las bandas, donde el atributo x es irrelevante, las bandas que aprenden los puntos tenderán a ser paralelas al eje x y perpendiculares al eje y. Localmente cada punto nos indica así que en su zona el valor del atributo x es poco relevante y que ese punto mide la distancia a la que se encuentran los demás puntos teniendo en cuenta sobre todo el valor del atributo y. Si observamos que un gran número de puntos muestra este comportamiento, la tendencia no es solo local, y nos puede estar sugiriendo que realicemos globalmente una proyección de acuerdo a la dirección de las bandas, y que por tanto nos quedemos sólo con el atributo y.

Para comprobar este hecho hemos generado 50 puntos aleatorios en una banda horizontal (figura 5.12.a), y en la figura 5.12.b se muestra la dirección de la banda que aprende cada punto. Puede observarse claramente como los puntos tienden a aprender la dirección de una banda horizontal. Numéricamente podemos calcular la dirección media y desviación típica de la dirección de las bandas que aprenden los puntos. La forma más sencilla de calcular esa dirección media es a partir de su vector v de cosenos directores. Podemos observar también si su desviación típica es baja y por tanto los puntos tienden a elegir bandas con direcciones parecidas.

En el ejemplo de la figura 5.12, el valor medio de v es $\overline{v} = (0.0230, 0.9916)$ y la desviación típica es $(0.1279, 0.0141)$. Podemos observar que la desviación típica es bastante baja, sobre todo la segunda componente. Estos valores indican que los puntos tienden a elegir bandas con cosenos directores donde la segunda coordenada es muy cercana a 1 (y con muy poca variación), y la primera componente es bastante



a. Puntos de una banda horizontal b. Dirección de las bandas

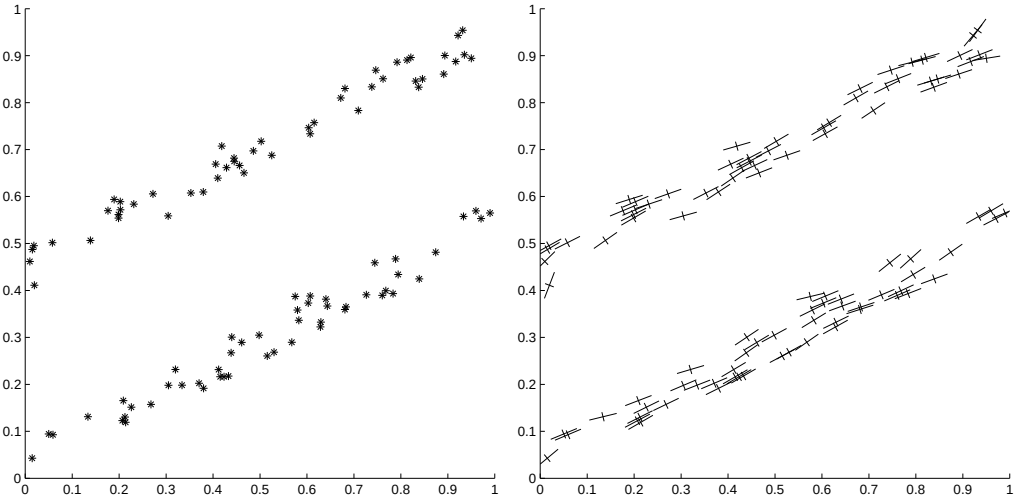
Figura 5.12: Puntos aleatorios en una banda horizontal y dirección de las bandas que aprenden esos puntos.

cercana a 0 (con un poco más de variación).

Por tanto en el ejemplo podemos concluir que globalmente los puntos eligen bandas paralelas al eje X, por lo que podemos considerar realizar una proyección sobre la dirección perpendicular, es decir, sobre el eje X.

Originalmente los puntos fueron generados tomando valores aleatorios en el intervalo [0,1] para la coordenada X, y valores aleatorios en el intervalo [0.7, 0.8] para la coordenada Y. Por tanto la característica de esta banda es que los puntos tienen un valor de Y comprendido entre 0.7 y 0.8. Si realizamos directamente la proyección sobre el eje Y obtendremos valores en el intervalo [0.7049, 0.7962].

Si en lugar de emplear sólo el valor de un atributo, la clase se determina mediante una combinación lineal de los atributos, también se pueden emplear la información de las bandas. En general una combinación lineal de atributos nos llevará geométricamente a bandas que no son paralelas a los ejes, pero localmente los puntos tenderán a tomar la dirección de esas bandas y si observamos que un gran número de puntos toma la misma dirección nos volverá a sugerir que realicemos una proyección de acuerdo a la dirección de las bandas. Al realizar la proyección



a. Puntos de dos bandas inclinadas b. Dirección de las bandas

Figura 5.13: Puntos aleatorios en dos bandas inclinadas y dirección de las bandas que aprenden esos puntos.

estamos reduciendo en uno el número de atributos o dimensiones.

Para comprobar este hecho hemos generado 100 puntos aleatorios repartidos en las dos bandas que se muestran en la figura 5.13.a. En la figura 5.13.b se muestra la dirección de la banda que aprende cada punto. Puede observarse claramente como los puntos tienden a aprender la dirección de dos bandas inclinadas paralelas.

Las dos bandas son paralelas con una inclinación de 30^o respecto del eje X y una anchura de 0.1. Por lo tanto, las coordenadas de los puntos deben de cumplir la relación $x = 2y$, salvo por la anchura de la banda y el desplazamiento de la banda respecto del eje X (una banda está ubicada más hacia arriba).

Si calculamos el valor medio de v obtenemos $\overline{v} = (-0.4397, 0.8858)$ y la desviación típica es $(0.1237, 0.0835)$. Podemos volver a observar en este ejemplo que la desviación típica es bastante baja, sobre todo en la segunda componente. Estos valores indican que los puntos tienden a elegir bandas con cosenos directores $(-0.4397, 0.8858)$, es decir, con una inclinación de 26.3992^o respecto del eje X (muy cercano a los 30^o originales).

También podemos interpretar $\overline{v} = (-0.4397, 0.8858)$ con esa desviación típi-

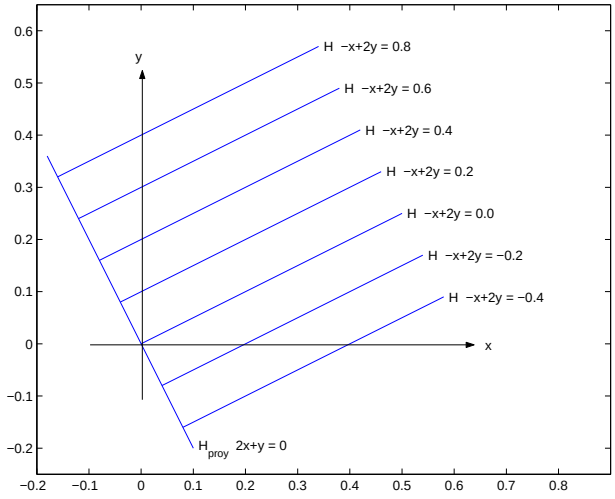


Figura 5.14: Planos paralelos de la forma $H \equiv -x+2y = cte$ y plano de proyección $H_{proy} \equiv 2x+y = 0$.

ca tan baja, como que los puntos tienden a aprender una banda con ese vector de cosenos directores, es decir que las bandas puede expresarse como $H \equiv -0.4397x + 0.8858y = cte$. Si elegimos como representante la banda con valor $cte = 0$, tendríamos la relación $x = 2.0146y$, que es muy parecido a la relación $x = 2y$ original, siempre salvo por desplazamiento de la banda.

Por tanto en el ejemplo podemos concluir que globalmente los puntos eligen bandas paralelas con una inclinación de 30^o respecto al eje X, por lo que podemos considerar realizar una proyección sobre la dirección perpendicular, es decir, sobre 120^o respecto del eje X. Esto es equivalente a realizar una reducción de la dimensión proyectando sobre un plano del tipo $H_{proy} \equiv 2x+y = cte$, si por ejemplo seleccionamos el plano que pasa por el origen tendríamos $H_{proy} \equiv 2x+y = 0$. Al realizar la proyección sobre H_{proy} desde direcciones perpendiculares (las de las bandas paralelas), vamos a tener distintos valores, de acuerdo a las distintas bandas de la forma $H \equiv -x+2y = cte$ que se obtienen con distintos valores de la constante cte (fig. 5.14).

Por lo tanto podemos reducir el problema original a una sola dimensión empleando los valores de $\bar{v}$ y calculando $x\prime = -0.4397x + 0.8858y$. De esta forma los

puntos de la banda superior tienen un valor de $x\prime$ en torno a 0.4, y los puntos de la banda inferior tienen un valor de $x\prime$ en torno a 0. En concreto, en todos los puntos de la banda superior $x\prime$ pertenece al intervalo [0.3554,0.4425], con una media de 0.4045 y desviación típica de 0.0238; en todos los puntos de la banda inferior $x\prime$ pertenece al intervalo [0.0025,0.0899], con una media de 0.0377 y desviación típica de 0.0260.

Originalmente los puntos fueron generados tomando puntos aleatorios en dos bandas paralelas con una inclinación de 30^o y una anchura de 0.1. Por tanto la característica de estas bandas es la inclinación de 30^o y una relación entre coordenadas de $x = 2y$ salvo por la anchura y el desplazamiento de las bandas. A partir de las bandas que han aprendido los puntos se extrae que existen dos bandas con una inclinación de 26.40^o y una relación $x = 2.01y$.

Como puede apreciarse, la información extraida a partir de los puntos es muy parecido a la información original con la que se diseñó el experimento. Y todo ello a pesar de la anchura de las bandas, de la elección aleatoria de los puntos, y de las posibles desviaciones respecto de la dirección de la banda original al elegir la dirección de la banda en cada punto.

En resumen, las bandas pueden sugerir automáticamente la reducción del número de dimensiones o atributos en una unidad. Al hacer esto mediante combinaciones lineales de los atributos originales, podemos dotar todavía de significado a los nuevos atributos y hacer que resulten comprensibles para un humano que tuviera que interactuar con el sistema. Este sistema permite reducir sólo en una unidad el número de dimensiones, pero si es necesario se puede repetir varias veces para reducir un número superior de atributos.

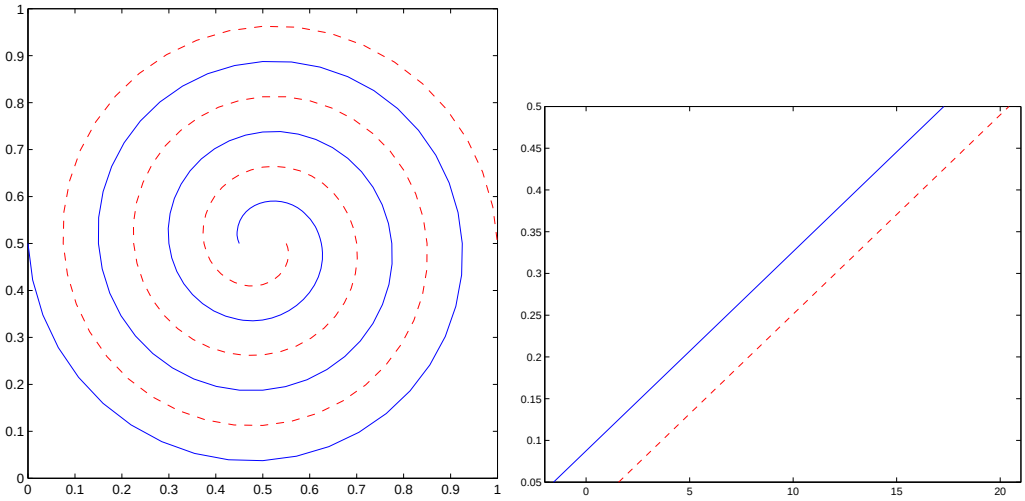
5.4.2 Cambio a coordenadas polares

La reducción de características del apartado anterior puede verse también como un cambio de base de coordenadas, pero podemos generalizar esta idea aún más. Si recordamos lo que sucedía con las bandas que aprendían los puntos aleatorios de una circunferencia (fig. 5.4.a), según el valor de ε las bandas "dibujaban" localmente la circunferencia o apuntaban al centro de la misma.

Se puede identificar este tipo de situaciones para proponer por ejemplo un cambio a coordenadas polares, tomando como centro el centro de curvatura de la figura que forman esos puntos. Así, por ejemplo, en las bases sintéticas de los anillos con área constante y con radio constante, se puede identificar que muchos puntos tienden a "dibujar" circunferencias concéntricas, y se puede proponer un cambio a coordenadas polares con centro en el punto (0.5, 0.5). El problema original se ha transformado en otro equivalente pero mucho más sencillo, donde el ángulo se ha convertido en un atributo irrelevante y la clasificación se puede realizar fijándonos solo en el radio. Además este problema equivalente es mucho más fácil de interpretar, ya que se puede decir que un punto tiene tal clase porque se encuentra a determinada distancia del punto (0.5, 0.5). También vamos a poder determinar con más facilidad los límites de las clases, necesitamos solamente dos valores reales que serán los radios inicial y final de la clase.

En problemas donde resulta complejo realizar una separación entre clases a priori, se puede aplicar este principio y realizar una transformación en la representación del problema. Por ejemplo, si nos encontramos con un problema con dos clases, donde los puntos de ambas clases se encuentran distribuidos a lo largo de dos espirales concéntricas como las que se muestran en la figura 5.15.a, en principio no es sencillo encontrar un método que permita distinguir entre los puntos de ambas espirales. Sin embargo, si realizamos una transformación y representamos los puntos en coordenadas polares, tenemos un problema equivalente al original, pero ahora las dos espirales se representan como dos rectas (figura 5.15.b) y es mucho más fácil distinguir entre los puntos de ambas clases porque son linealmente separables.

Podríamos aprender las bandas para cada uno de los puntos y detectar que sugieren un cambio a coordenadas polares con centro en el punto (0.5,0.5) para obtener clases linealmente separables. Además, si volvemos a emplear por segunda vez las bandas que aprende cada punto representado en coordenadas polares, podríamos detectar que los puntos de las clases están alineados. Entonces podemos usar el método de reducción de características del apartado anterior para proponer una reducción de dimensiones, reemplazando las dos coordenadas por una combinación de ambas. Así se podría transformar el problema original en otro equivalente con



a. Dos espirales concéntricas b. Transf. coordenadas polares

Figura 5.15: Problema donde los puntos estan distribuidos en dos clases que se corresponden con dos espirales concéntricas en R^2.

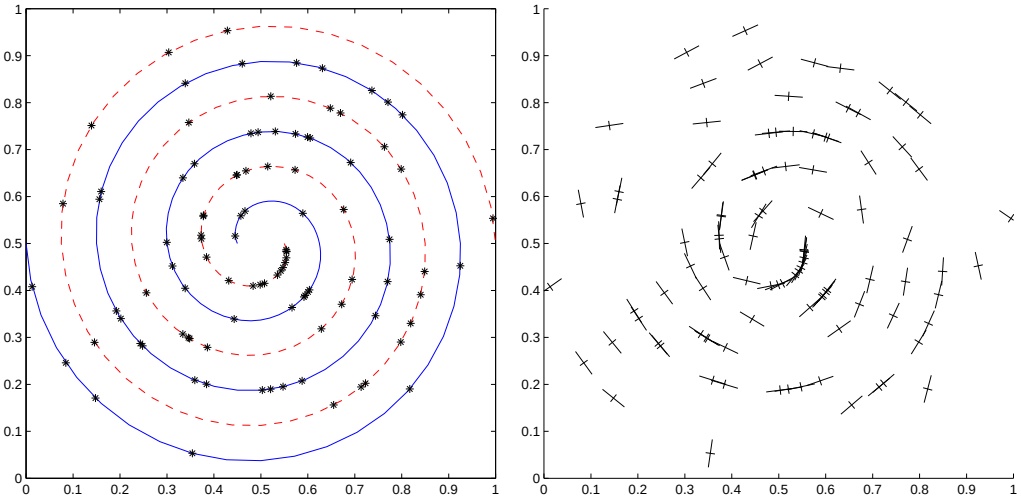
un solo atributo.

Para comprobar este hecho hemos generado 100 puntos aleatorios repartidos en dos espirales (figura 5.16.a), y en la figura 5.16.b se muestra la dirección de la banda que aprende cada punto. Puede observarse claramente como los puntos tienden a aprender la dirección de la espiral.

En primer lugar podemos detectar que no se están aprendiendo bandas paralelas y no se propone la reducción de dimensiones mediante la proyección sobre un hiperplano porque al calcular la media de los vectores de cosenos directores obtenemos $\overline{v} = (-0.0508, 0.5743)$ con una desviación típica de $(0.7575, 0.3172)$, que resulta demasiado elevada como para sugerir que los puntos tienden a aprender bandas paralelas.

Para detectar si debe realizarse un cambio a coordenadas polares y qué punto debe elegirse como centro de ese cambio, proponemos el método que se detalla a continuación.

Dado una propuesta de punto *c* que actúe como centro del cambio de coordenadas, proponemos calcular la suma de las distancias de *c* a cada una de las direc-



a. Puntos de las espirales b. Dirección de las bandas

Figura 5.16: Puntos aleatorios en las dos espirales y dirección de las bandas que aprenden esos puntos.

ciones perpendiculares a las bandas al cuadrado. Se puede calcular la distancia de un punto a la perpendicular de una banda o Hiperplano de acuerdo a la ecuación 5.9. Por tanto, para calcular la suma de estas distancias al cuadrado proponemos:

$$d_{centro}(c,P,V) = \sum_{j=1}^{n} \left[\sum_{i=1}^{N} (c_i - p_i)^2 - \left(\sum_{i=1}^{N} v_i (c_i - p_i) \right)^2 \right] \tag{5.11}$$

donde n es el número de puntos que se consideran, N es el número de dimensiones, $c = (c_1, c_2, \ldots, c_N)$ es el punto que se propone como centro, P es el conjunto de los n puntos conocidos, cada uno de la forma $p = (p_1, p_2, \ldots, p_N)$, y V es el conjunto de vectores de cosenos directores de esos puntos, cada uno de la forma $v = (v_1, v_2, \ldots, v_N)$. Para determinar qué punto debemos utilizar como centro, debemos encontrar el punto c que minimiza esta expresión. En una situación ideal de puntos con bandas perfectamente alineadas alrededor de un centro, la ecuación 5.11 tomaría el valor 0 para ese centro.

Aplicando esta forma de elegir el centro de cambio a coordenadas polares en el ejemplo anterior de los puntos elegidos aleatoriamente en las dos espirales concéntricas obtenemos que se propone como centro el punto (0.4859,0.4807), que como

puede apreciarse es muy parecido al (0.5,0.5), el centro original de las dos espirales. Para reafirmar que este método funciona, cabe mencionar que en el punto (0.4859,0.4807) la expresión 5.11 toma el valor 0.8914, esto implica que dado que hay 100 puntos, la distancia media del centro propuesto a la perpendicular de la banda de cada punto es de sólo 0.0089. En cambio, en los extremos del cuadrado unidad, el valor de la expresión 5.11 está en torno a 25.

Con este sistema se puede detectar también los centros de curvatura de alguna figura que formasen los puntos, aunque esa figura no llegue a ser algo parecido a una circunferencia completa, sino por ejemplo, un arco de 120^o, o varios de esos arcos enlazados.

Estos cambios a coordenadas polares también pueden ser de tipo local en una zona, de tal forma que a la hora de clasificar un punto nuevo, se puedan realizar diferentes transformaciones desde diferentes puntos para ver cómo de lejos ve cada zona al punto nuevo. Y podemos realizar la clasificación combinando la información que nos proporcionan las distintas zonas, o símplemente de acuerdo a la zona que ve más cercano al punto nuevo.

También hay que dejar claro que estas transformaciones (proyecciones sobre un hiperplano y cambios a coordenadas polares), se han explicado en problemas de dos dimensiones, pero pueden generalizarse facilmente a problemas N–Dimensionales. En este último caso se debe tener cuidado porque existen muchas más posibilidades que en 2–D, por ejemplo, si realizamos un cambio a coordenadas polares en 3–D nos podemos encontrar con que el mínimo de la expresión 5.11 se alcanza en un punto o a lo largo de una recta. Si el mínimo se alcanza en un punto, probablemente sea el centro de una esfera, y debamos realizar un cambio a coordenadas polares en 3–D. En cambio si el mínimo se alcanza a lo largo de una recta, probablemente sea el eje de un cilindro, y sea mejor realizar un cambio a coordenadas cilíndricas o realizar una proyección según la dirección de ese eje. Después de realizar esta proyección tendríamos un problema 2–D donde podríamos volver a utilizar estos métodos para reducir dimensiones o proponer cambios de coordenadas.

5.5 Conclusiones

En este capítulo se ha propuesto una nueva medida de distancia que puede emplearse, en conjunción con algún método de clasificación, en problemas de clasificación: la distancia basada en bandas. En concreto se han realizado pruebas con los métodos 1–NN y *k*–NN utilizando como distancia básica esta distancia basada en bandas. Se ha estudiado la utilidad de esta medida de distancia y se ha realizado una comparación exhaustiva en 68 bases de casos con los métodos 1–NN y *k*–NN cuando emplean las funciones de distancia básicas de la Geometría (con las seis funciones de distancia empleadas en el capítulo anterior). Se deja para trabajos futuros el estudio de los métodos de los ε–entornos con esta medida de distancia.

La nueva medida de distancia se ha revelado bastante útil. Cuando empleamos el método de clasificación 1–NN, la distancia basada en bandas obtiene un porcentaje de acierto medio superior al resto de las funciones de distancia, y es la medida de distancia que obtiene los mejores resultados en mayor número de bases de casos. Cuando empleamos el método de clasificación *k*–NN con la distancia basada en bandas, también obtiene unos buenos resultados, aunque bastante desiguales. En algunos dominios la mejora es bastante importante, llegando a lograr en la base de casos Tic–Tac–Toe un incremento del acierto de casi un 15% con el método de los *k*–NN y del 22% con los 1–NN. En cambio, en otros dominios del UCI–Repository sufre una degradación bastante importante, como por ejemplo en la base Granada Digits donde obtiene un 38.2% menos que el mejor método.

En cuanto a las bases de casos sintéticas, la distancia basada en bandas obtiene los mejores resultados con todas las variantes de las bases de casos de los anillos con área constante y con radio constante, y tiende a obtener buenos resultados con las bases de los senos. En cambio, con los cuadrados y las bandas horizontales los resultados son discretos. Este hecho era previsible con las bases de casos de los cuadrados, pero resulta especialmente sorprendente con las bases de casos de las bandas horizontales, donde en principio parecía que debía comportarse mejor. En el capítulo se han analizado los posibles motivos de este comportamiento.

Parece que si la base de casos tiene atributos irrelevantes es especialmente importante usar de manera previa algún método para eliminarlos, o bien usar la distan-

cia basada en bandas primero para proponer la reducción de dimensiones (y eliminación de atributos irrelevantes), y después para realizar la clasificación propiamente dicha.

Es interesante usar las bandas que localmente aprende cada punto para extraer conclusiones o características que afectan a un subconjunto o a todos los puntos de una base de casos. Cuando muchos puntos responden a un esquema o patrón común, se puede interpretar ese comportamiento de manera global y por ejemplo proponer la proyección sobre un hiperplano perpendicular a esas bandas y reducir las dimensiones del problema original. Esta reducción de características puede verse también como un cambio de base de coordenadas.

También se puede identificar las situaciones en que los puntos "dibujan" localmente una circunferencia o un arco, seleccionar como centro el centro de curvatura de la figura que forman esos puntos, y proponer un cambio a coordenadas polares.

De cualquiera de esta formas, el problema original se transforma en otro equivalente pero más sencillo de resolver. Además este problema equivalente suele ser más fácil de interpretar, ya que por ejemplo se puede justificar que un punto tiene tal clase porque se encuentra a determinada distancia de cierto punto.

5.6 Trabajos Futuros

Ahora tenemos un parámetro ε que controla el grado de localidad que el punto debe usar cuando aprenda la dirección de su hiperplano. Básicamente tenemos dos alternativas para elegir su valor. En primer lugar ε puede ser un valor real fijo, constante para toda la base de casos. Se puede proporcionar directamente ese valor o puede ser calculado, por ejemplo como resultado de alguna expresión que permita tener en cuenta las características de cada base de casos. O bien, ε puede ser un valor real diferente para cada punto, que se calcula basándose en los alrededores del punto, es decir, cada punto tienen su propio ε que recoge las características especiales de esa región.

En principio el valor de ε debe ser suficientemente grande para incluir la información de los alrededores del punto. Lo ideal sería incluir al *cluster* o grupo de

puntos donde se encuentra ubicado el punto. Pero no debe ser demasiado grande para que no se vea afectado por puntos demasiado alejados, que son puntos aislados o pertenecen a otros grupos de puntos.

Hemos realizado algunos experimentos para encontrar un método que permita determinar de manera automática el valor de ε para cada punto, pero nos hemos encontrado con más dificultades de las esperadas y los resultados no han sido concluyentes. Esta parte de la investigación está directamente relacionada con el *clustering*, es decir dado un conjunto de ejemplos determinar en primer lugar el número de conjuntos o clases en que están agrupados esos ejemplos, y posteriormente, dado un punto, determinar a cuál de esos conjuntos pertenece. Dado que este problema cae fuera del ámbito de nuestra línea principal de trabajo, y dadas las dificultades que hemos encontrado, hemos decidido dejar la determinación automática del valor de ε como una línea de trabajo que debe abordarse en un futuro. Aquí podrían usarse técnicas ya conocidas de *clustering* para determinar el tamaño de la nube donde se encuentra el punto, y elegir un valor de ε que recoja la información de los puntos de la nube pero sin llegar a incluir información de otras nubes de puntos. Otra posibilidad sería explorar alguna medida que recoja cómo influyen los puntos de la misma clase/nube y los de otras en el ajuste del hiperplano. Para así determinar un valor de corte ε que permita tener una influencia grande antes de acabar la nube donde se encuentra el punto y que reduzca notablemente la influencia mucho antes de alcanzar otras nubes de puntos.

Resulta interesante emplear las bandas que aprenden los puntos para extraer información sobre la base de casos. En una primera aproximación nos podemos plantear intentar recordar sólo algunos puntos junto con las bandas que han aprendido, y realizar la clasificación de los puntos nuevos usando sólo esa información. Se puede intentar usar varios criterios para seleccionar los puntos que se deben conservar. Por ejemplo se puede intentar conservar los puntos que se consideren más representativos, o algunos puntos de cada nube de puntos (especialmente de las nubes alargadas en forma de banda), o intentar conservar los puntos frontera (aquellos que sirven para delimitar las fronteras entre las clases o las nubes de puntos). Así logramos reducir la cantidad de información que es necesario almacenar para realizar la clasificación.

Otra aproximación diferente consistiría en intentar aprender grandes bandas, de tal forma que almacenaríamos información sobre una banda por cada nube de puntos "importante". Estaríamos aprendiendo información de un nivel superior, olvidándonos de los puntos concretos. De hecho estas "grandes bandas" o metabandas no tienen porqué estar ubicadas en ninguno de los puntos concretos de entrenamiento.

Una tercera aproximación podría consistir en realizar una clasificación jerárquica. Se pueden aprender esas metabandas o "grandes bandas" y conservar la información de los puntos concretos. Cuando el punto que se debe clasificar se encuentre alejado de las metabandas, o más de una diga que ve el punto a una distancia similar, entonces recurrir a los puntos concretos con sus bandas para realizar la clasificación.

Si la base de casos tiene atributos irrelevantes es especialmente importante usar de manera previa algún método para eliminarlos, o bien usar la distancia basada en bandas primero para proponer la reducción de dimensiones (y eliminación de atributos irrelevantes), y después para realizar la clasificación propiamente dicha.

Es interesante usar las bandas que localmente aprende cada punto para extraer conclusiones o características que afectan a un subconjunto o a todos los puntos de una base de casos. En el capítulo se ha introducido cómo se puede interpretar que muchos puntos aprendan bandas con una dirección parecida, con un comportamiento global que tiende a ubicar los puntos de acuerdo a grandes bandas. Así podemos proponer la proyección sobre un hiperplano perpendicular a esas bandas y reducir las dimensiones del problema original.

Esta reducción de características puede verse también como un cambio de base de coordenadas. Pero esta idea puede generalizarse aún más y podemos proponer también cambios a coordenadas polares.

Se puede identificar las situaciones en que los puntos "dibujan" localmente una circunferencia o un arco, seleccionar como centro el centro de curvatura de la figura que forman esos puntos, y proponer un cambio a coordenadas polares.

De cualquiera de esta formas, el problema original se transforma en otro equivalente pero más sencillo de resolver. Además este problema equivalente suele ser

mucho más fácil de interpretar, ya que por ejemplo se puede justificar que un punto tiene tal clase porque se encuentra a determinada distancia de cierto punto.

Estos cambios a coordenadas polares también pueden ser de tipo local en una zona, de tal forma que a la hora de clasificar un punto nuevo, se puedan realizar diferentes transformaciones desde diferentes puntos para ver cómo de lejos ve cada zona al punto nuevo. Y podemos realizar la clasificación combinando la información que nos proporcionan las distintas zonas, o símplemente de acuerdo a la zona que ve más cercano al punto nuevo.

Será interesante profundizar en trabajos futuros en este tipo de cambios de base y coordenadas y la extracción de información a partir de los puntos y las bandas que ha aprendido cada punto.

Capítulo 6

Resumen, Conclusiones y Trabajos Futuros

En este capítulo final, vamos a realizar un resumen de los aspectos más destacables de este trabajo y del camino que se ha realizado para llevar a cabo esta investigación. A continuación recordaremos las conclusiones a las que se ha llegado tras su finalización, y por último se discutirán algunas líneas de trabajo que no se han concluido durante este trabajo o que se abren tras esta investigación.

6.1 Resumen

En el primer capítulo se realizó una presentación del trabajo, se definieron los objetivos a conseguir y se esbozó la organización de este trabajo. Se explicó la metodología de pruebas que se ha empleado para obtener los resultados de los distintos clasificadores, y se han presentado las 68 bases de casos utilizadas en las pruebas.

En el segundo capítulo se ha analizado el Razonamiento Basado en Casos (RBC) en general: tipos de razonadores, fundamentos del RBC, funcionamiento, ventajas

e inconvenientes de su uso.

En el tercer capítulo se ha enlazado el uso del RBC con los problemas de clasificación. Comenzó con una descripción de los problema de clasificación en general. A continuación se presentaron algunos métodos que se han utilizado en la literatura para abordar los problemas de clasificación. Se analizó la forma de utilizar el Razonamiento Basado en Casos para realizar labores de clasificación, y las peculiaridades que plantea aplicar RBC en este tipo concreto de problemas. Aquí el peso de la clasificación suele recaer en gran medida en el concepto de similitud que consideremos. Las funciones de distancia constituyen la pieza básica con que trabajar, porque toda medida de distancia nos define implicitamente similitud. Por tanto las distancias van a permitir definir similitud entre ejemplos, y dependiendo de la función de distancia que usemos tendremos una noción de similitud u otra. Se aclara qué entendemos por función de distancia a lo largo de este trabajo y se discute que éstas pueden ser menos restrictivas que las funciones de distancia utilizadas tradicionalmente en Geometría. Por último se presenta una nueva medida de distancia: la distancia basada en bandas o hiperplanos.

En el capítulo 4 se comienza realizando un repaso y estudio de los métodos del vecino más cercano (1–NN) y de los *k* vecinos más cercanos (*k*–NN). A continuación se proponen variantes de los *k*–NN que se basan en la inclusión de hasta 3 características: ε–entornos, ε–entornos^{k-NN} y una heurísticas para seleccionar la medida de distancia.

En los ε–entornos se impone un umbral de distancia ε en el conjunto K de los *k* vecinos más cercanos. Así para clasificar un caso nuevo, se establece un entorno alrededor de ese punto y se tiene en cuenta sólo los puntos que están dentro de ese entorno. De este modo siempre se tiene garantizado considerar solo puntos cercanos, independientemente de la densidad de puntos de la base de casos: cuando el punto nuevo se encuentra en un área poco poblada del espacio se considerarán menos puntos y cuando se encuentra en una región densamente poblada se considerarán más.

Si el valor de ε es demasiado pequeño, puede ocurrir que el conjunto K esté vacío. En los ε–entornos^{k-NN} siempre que K esté vacío se elige la clase del vecino

o los vecinos más cercanos, es decir, cuando el ε–entorno "falla" entonces debemos clasificar usando *k*–NN, que siempre proporciona vecinos más cercanos y una clasificación. También se ha estudiado en este trabajo usar 1–NN cuando K está vacío (método ε–entornos^{1-NN}).

Analizando los resultados de los clasificadores con las distintas funciones de distancia se aprecia que existe cierta relación entre el acierto que obtiene 1–NN con diferentes distancias y el acierto de otros métodos de clasificación basados en esas distancias. Por eso, y considerando que los métodos 1–NN son relativamente rápidos, se propuso la siguiente heurística: *entrena y prueba métodos 1–NN con un conjunto de medidas de distancia y utiliza el método basado en distancias deseado (k–NN, ε–entorno, ε–entorno^{k-NN}...) con la distancia que ofrece mayor porcentaje de acierto con 1–NN.*

Se han realizado diferentes combinaciones de estas características y se han estudiado 8 métodos de clasificación basados en *k*–NN. Se han realizado pruebas exhaustivas con las 68 bases de casos y contrate de hipótesis con un test pareado *t*–Student con dos colas y un grado de confianza del 95% para determinar si las diferencias observadas entre los clasificadores eran significativas desde un punto de vista estadístico o explicables símplemente por fluctuaciones debidas al muestreo de la población.

En el capítulo 5 se parte de la distancia de las bandas introducida en el apartado 3.6.1, y se introduce un método nuevo de clasificación basado en esta medida de distancia.

En primer lugar se estudia cómo aprender la dirección de la banda, y después de analizar varias alternativas se ha optado por un método con un parámetro ε que controla la localidad del método a la hora de aprender la dirección de las bandas. Después se realiza la extensión del aprendizaej de la dirección a problemas multiclase, donde se distinguen los puntos de la misma clase de los de otras clases, y se introduce un nuevo parámetro F que permite controlar el grado en que queremos que la banda "huya" de direcciones donde se encuentran puntos de otras clases.

Por último se propone un algoritmo para el aprendizaje de dos radios r y R que permiten controlar la anchura de la banda y si queremos o no bandas de longitud in-

finita. Se realizan experimentos exhaustivos con los métodos 1–NN y *k*–NN usando como distancia base la distancia de las bandas.

6.2 Conclusiones

Básicamente podemos separar las conclusiones en dos grandes bloques.

6.2.1 Conclusiones sobre las características C1, C2 y C3

Las pruebas exhaustivas revelan claramente que las tres características propuestas se han revelado útiles, y las variantes propuestas mejoran el porcentaje de acierto del método *k*–NN básico. Como era de esperar, la elección de la medida de distancia es importante. Afecta de manera decisiva al rendimiento del clasificador, indepedientemente de la familia de clasificadores, y su elección depende de la base de casos. La característica C3 o heurística es usada por los mejores clasificadores y se ha revelado como la característica más útil con diferencia. Las características C1 y C2 logran mejorar la clasificación pero en menor medida.

El método ε–entornos^{1-NN} Heur logra mejorar de manera significativa al método *k*–NN básico en 15 bases de casos y no es significativamente inferior en ninguna. También mejora frecuentemente a los métodos *k*–NN Heur (7–58–3) y ε–entorno (11–55–2) de manera significativa. Desde un punto de vista estadístico no existen diferencias significativas entre los métodos ε–entornos^{1-NN} Heur, ε–entornos^{k-NN} Heur y ε–entornos Heur. Como conclusión, el método que se propone para emplear cuando no se dispone de información a priori sobre la base de casos es el F1 o ε–entornos^{1-NN} Heur, que incluye las características C1, C2$'$ y C3.

Se ha encontrado cierta evidencia sobre la debilidad de *k*–NN cuando la distribución de puntos no es constante a lo largo del espacio (y por lo tanto el valor óptimo de *k* varía). Los métodos *k*–NN tienen más problemas que los ε–entornos si existen grandes zonas del espacio con una distribución de puntos y un valor óptimo de *k* muy diferentes. El método *k*–NN Heur es claramente preferible en las bases de casos de las bandas, y los ε–entornos en las bases de casos de los anillos, tanto de

área como de radio constante.

6.2.2 Conclusiones sobre la distancia basada en bandas

La nueva medida de distancia se ha revelado bastante útil, aunque con un comportamiento muy dispar según la base de casos y el método de clasificación empleado. Cuando empleamos el método de clasificación 1–NN, la distancia basada en bandas obtiene un porcentaje de acierto medio superior al resto de las funciones de distancia, y es la medida de distancia que obtiene los mejores resultados en mayor número de bases de casos, concretamente en 38 de las 68 bases. Cuando empleamos el método de clasificación *k*–NN con la distancia basada en bandas, también obtiene unos buenos resultados, aunque bastante desiguales. En algunos dominios la mejora es bastante importante, llegando a lograr en la base de casos Tic–Tac–Toe un incremento del acierto de casi un 15% con el método de los *k*–NN y del 22% con los 1–NN. En cambio, en otros dominios del UCI–Repository sufre una degradación bastante importante, como por ejemplo en la base Granada Digits donde obtiene un 38.2% menos que el mejor método.

En cuanto a las bases de casos sintéticas, la distancia basada en bandas obtiene los mejores resultados con todas las variantes de las bases de casos de los anillos con área constante y con radio constante, y tiende a obtener buenos resultados con las bases de los senos. En cambio, con los cuadrados y las bandas horizontales los resultados son discretos. Este hecho era previsible con las bases de casos de los cuadrados, pero resulta especialmente sorprendente con las bases de casos de las bandas horizontales, donde en principio parecía que debía comportarse mejor. En el capítulo se han analizado los posibles motivos de este comportamiento.

Parece que si la base de casos tiene atributos irrelevantes es especialmente importante usar de manera previa algún método para eliminarlos, o bien usar la distancia basada en bandas primero para proponer la reducción de dimensiones (y eliminación de atributos irrelevantes), y después para realizar la clasificación propiamente dicha.

6.3 Trabajos Futuros

En un futuro creemos que puede ser interesante el estudio de nuevas medidas de distancia no usuales que permitan ajustarse mejor a las peculiaridades de cada base de casos.

Se pueden estudiar también mecanismos para reducir las necesidades de memoria con técnicas de compactación de información, técnicas de recuerdo y olvido selectivo (recordar los ejemplos útiles y olvidar los que aportan menos información). Entonces tendrá sentido estudiar el acierto con el conjunto de entrenamiento y el tamaño del conjunto conservado.

Se deja para trabajos futuros el estudio de los métodos de los ε–entornos con la medida de distancia de las bandas. También se deja para trabajos futuros una comparación detallada de los métodos de clasificación empleados en esta tesis con otros métodos de clasificación utilizados en la literatura científica. Esta comparativa está más allá del ámbito de este trabajo y además no ha sido posible realizarla por razones de tiempo.

En la clasificación con la distancia de las bandas tenemos un parámetro ε que es un valor real fijo que controla el grado de localidad que el punto debe usar cuando aprende la dirección de su hiperplano. En este trabajo se ha estimado el valor de ε mediante validación cruzada, y es constante para toda la base de casos. En trabajos futuros se puede estudiar la posibilidad de que este valor sea calculado, por ejemplo como resultado de alguna expresión que permita tener en cuenta las características de cada base de casos. Aunque es mucho más interesante que ε sea un valor real diferente para cada punto, que se calcula basándose en los alrededores del punto, es decir, cada punto tendría su propio ε que recoge las características especiales de esa región.

En principio el valor de ε debe ser suficientemente grande para incluir la información de los alrededores del punto. Lo ideal sería incluir al *cluster* o grupo de puntos donde se encuentra ubicado el punto. Pero no debe ser demasiado grande para que no se vea afectado por puntos demasiado alejados, que son puntos aislados o pertenecen a otros grupos de puntos.

Hemos realizado algunos experimentos para encontrar un método que permita determinar de manera automática el valor de ε para cada punto, pero nos hemos encontrado con más dificultades de las esperadas y los resultados no han sido concluyentes. Esta parte de la investigación está directamente relacionada con el *clustering*, es decir dado un conjunto de ejemplos determinar en primer lugar el número de conjuntos o clases en que están agrupados esos ejemplos, y posteriormente, dado un punto, determinar a cuál de esos conjuntos pertenece. Dado que este problema cae fuera del ámbito de nuestra línea principal de trabajo, y dadas las dificultades que hemos encontrado, hemos decidido dejar la determinación automática del valor de ε como una línea de trabajo que debe abordarse en un futuro. Aquí podrían usarse técnicas ya conocidas de *clustering* para determinar el tamaño de la nube donde se encuentra el punto, y elegir un valor de ε que recoja la información de los puntos de la nube pero sin llegar a incluir información de otras nubes de puntos. Otra posibilidad sería explorar alguna medida que recoja cómo influyen los puntos de la misma clase/nube y los de otras en el ajuste del hiperplano. Para así determinar un valor de corte ε que permita tener una influencia grande antes de acabar la nube donde se encuentra el punto y que reduzca notablemente la influencia mucho antes de alcanzar otras nubes de puntos.

Resulta interesante emplear las bandas que aprenden los puntos para extraer información sobre la base de casos. En una primera aproximación nos podemos plantear intentar recordar sólo algunos puntos junto con las bandas que han aprendido, y realizar la clasificación de los puntos nuevos usando sólo esa información. Se puede intentar usar varios criterios para seleccionar los puntos que se deben conservar. Por ejemplo se puede intentar conservar los puntos que se consideren más representativos, o algunos puntos de cada nube de puntos (especialmente de las nubes alargadas en forma de banda), o intentar conservar los puntos frontera (aquellos que sirven para delimitar las fronteras entre las clases o las nubes de puntos). Así logramos reducir la cantidad de información que es necesario almacenar para realizar la clasificación.

Otra aproximación diferente consistiría en intentar aprender grandes bandas, de tal forma que almacenaríamos información sobre una banda por cada nube de puntos "importante". Estaríamos aprendiendo información de un nivel superior,

olvidándonos de los puntos concretos. De hecho estas "grandes bandas" o metabandas no tienen porqué estar ubicadas en ninguno de los puntos concretos de entrenamiento.

Una tercera aproximación podría consistir en realizar una clasificación jerárquica. Se pueden aprender esas metabandas o "grandes bandas" y conservar la información de los puntos concretos. Cuando el punto que se debe clasificar se encuentre alejado de las metabandas, o más de una diga que ve el punto a una distancia similar, entonces recurrir a los puntos concretos con sus bandas para realizar la clasificación.

Si la base de casos tiene atributos irrelevantes es especialmente importante usar de manera previa algún método para eliminarlos. Tambien puede ser factible usar la distancia basada en bandas primero para proponer la reducción de dimensiones (y eliminación de atributos irrelevantes), y después para realizar la clasificación propiamente dicha.

Si nos encontramos en una base de casos como las bandas, donde el atributo x es irrelevante, las bandas que aprenden los puntos tenderán a ser paralelas al eje x y perpendiculares al eje y. Localmente un punto puede indicar que en su zona el valor de un atributo es poco relevante y que ese punto mide la distancia a la que se encuentran los demás puntos teniendo en cuenta sobre todo el valor de otros atributos. Si observamos que un gran número de puntos muestra este comportamiento, la tendencia no es solo local, y nos puede estar sugiriendo que realicemos globalmente una proyección de acuerdo a la dirección de las bandas, y que por tanto nos quedemos con un atributo menos, o con una combinación lineal de los atributos que tiene una dimensión menos que la base de casos original.

En resumen, las bandas pueden sugerir automáticamente la reducción del número de dimensiones o atributos en una unidad. Al hacer esto mediante combinaciones lineales de los atributos originales, podemos dotar todavía de significado a los nuevos atributos y resultar comprensibles para un humano que interactuara con el sistema. Este sistema permite reducir sólo en una unidad el número de dimensiones, pero si es necesario se puede repetir varias veces para reducir un número superior de atributos.

Esta reducción de características puede verse también como un cambio de base de coordenadas. Pero esta idea puede generalizarse aún más y se puede identificar situaciones en que se proponga por ejemplo un cambio a coordenadas polares, tomando como centro el centro de curvatura de la figura que forman un conjunto de puntos.

Estos cambios a coordenadas polares también pueden ser de tipo local en una zona, de tal forma que a la hora de clasificar un punto nuevo, se puedan realizar diferentes transformaciones desde diferentes puntos para ver cómo de lejos ve cada zona al punto nuevo. Y podemos realizar la clasificación combinando la información que nos proporcionan las distintas zonas, o símplemente de acuerdo a la zona que ve más cercano al punto nuevo.

En ambos casos, tanto si se propone usar combinaciones lineales de atributos o una cambio a coordenadas polares, el problema original se transforma en otro equivalente pero mucho más sencillo. Además este problema equivalente puede resultar incluso más fácil de interpretar que el original.

Apéndice A

Minimización de $ajuste\_multi_H$ mediante el método de los multiplicadores de Lagrange

Para usar la medida de distancia basada en bandas en problemas de clasificación, se proponía aprender para cada punto conocido la dirección de la banda que mejor se adapta a los puntos que se encuentran a su alrededor.

Al final, en la sección 5.1.2 se proponía encontrar el Hiperplano H o, lo que es equivalente, encontrar el vector $v = H^{\perp}$ que minimiza $ajuste\_multi_H$. Se definía el *ajuste multiclase* de un hiperplano $v = H^{\perp}$ como (ecuación (5.6))

$$\begin{aligned} ajuste\_multi_H &= \sum_{i=1}^{n_=} \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right)^2 e^{-\frac{4}{\varepsilon^2} d(x_i, x_0)} + \\ & F \sum_{i=1}^{n_{\neq}} \left(\sum_{j=1}^{N} (x\prime_{ij} - x_{0j})^2 - \left(\sum_{j=1}^{N} (x\prime_{ij} - x_{0j}) v_j \right)^2 \right) e^{-\frac{4}{\varepsilon^2} d(x\prime_i, x_0)} \end{aligned} \tag{A.1}$$

donde $F \in \mathrm{R}_0^+$, N es el número de atributos o dimensiones, n es el número de puntos, $n_=$ es el número de puntos de la misma clase, $n_{\neq}$ es el número de puntos de otras

clases, x_{0j} es el j–ésimo atributo o coordenada de p, x_{ij} es el j–ésimo atributo del i–ésimo punto de la misma clase, $x\prime_{ij}$ es el j–ésimo atributo del i–ésimo punto que es de otra clase, y $v_j = \cos\alpha_j$ es el j–ésimo coseno director de H. Se debe hacer notar que $\sum_{i=1}^{n_=}$ sólo recopila información de los puntos de la misma clase, y $\sum_{i=1}^{n_\neq}$ hace lo propio con los de otras clases.

Este problema en principio puede parecer bastante complejo y lento de resolver, pero puede abordarse como un problema de minimización con restricciones donde se pueden aplicar el método de los multiplicadores de Lagrange para obtener una solución mucho más directa. Así podemos transformarlo en resolver un sistema de N ecuaciones lineales con N incógnitas, mucho más sencillo de afrontar. Dado que en algunas bases de casos el número de atributos (y por lo tanto el número de ecuaciones e incógnitas) puede llegar a ser bastante elevado, en las pruebas que se han realizado hemos usado métodos numéricos de aproximación a la solución.

En general, dadas dos funciones $f(x_1,x_2,\ldots,x_n)$ y $g(x_1,x_2,\ldots,x_n)$ con derivadas parciales de primer órden, si queremos encontrar los puntos de la superficie dada por $g(x_1,x_2,\ldots,x_n)=0$ en los que la función $f(x_1,x_2,\ldots,x_n)$ alcanza valores máximos o mínimos, podemos aplicar el método de maximización o minimización con restricciones de los multiplicadores de Lagrange. La función $g(x_1,x_2,\ldots,x_n)$ se comporta como una restricción, y el teorema de los multiplicadores de Lagrange nos dice que el punto p donde se encuentra el máximo o mínimo debe cumplir simultaneamente las siguientes ecuaciones

$$\begin{aligned} g(x_1,x_2,\ldots,x_n) &= 0 \\ \frac{\partial f(x_1,x_2,\ldots,x_n)}{\partial x_1} &= \lambda\frac{\partial g(x_1,x_2,\ldots,x_n)}{\partial x_1} \\ \frac{\partial f(x_1,x_2,\ldots,x_n)}{\partial x_2} &= \lambda\frac{\partial g(x_1,x_2,\ldots,x_n)}{\partial x_2} \\ &\vdots \\ \frac{\partial f(x_1,x_2,\ldots,x_n)}{\partial x_n} &= \lambda\frac{\partial g(x_1,x_2,\ldots,x_n)}{\partial x_n} \end{aligned} \tag{A.2}$$

para algún escalar λ. Después de resolver este sistema de $n+1$ ecuaciones con $n+1$ incógnitas ($x_1,x_2,\ldots,x_n$ y λ), se puede evaluar la función f en cada uno de los puntos solución para comprobar si se alcanza un máximo o un mínimo.

Nosotros queremos encontrar el hiperplano H que minimiza $ajuste\_multi_H$, por tanto podemos plantear el problema como buscar el vector $v = H^{\perp}$ que minimiza $ajuste\_multi_H$. Entonces la función que se debe minimizar es $ajuste\_multi_H$ (ecuación (A.1)), y la restricción que se debe cumplir es $\sum_{i=1}^{N} v_i^2 = 1$ porque $v = H^{\perp}$ debe ser un vector unitario perpendicular al hiperplano H, es decir, sus componentes son los cosenos directores de H. Si seguimos usando la notación de la ecuación (A.1) tendríamos que encontrar los valores $v_1, v_2, \ldots, v_N$ y λ que resuelven el siguiente sistema de ecuaciones:

$$\begin{aligned}
\sum_{i=1}^{N} v_i^2 - 1 &= 0 \\
\frac{\partial ajuste_multi_H(v_1, v_2, \ldots, v_N)}{\partial v_1} &= \lambda \frac{\partial \sum_{i=1}^{N} v_i^2 - 1}{\partial v_1} \\
\frac{\partial ajuste_multi_H(v_1, v_2, \ldots, v_N)}{\partial v_2} &= \lambda \frac{\partial \sum_{i=1}^{N} v_i^2 - 1}{\partial v_2} \\
&\vdots \\
\frac{\partial ajuste_multi_H(v_1, v_2, \ldots, v_N)}{\partial v_N} &= \lambda \frac{\partial \sum_{i=1}^{N} v_i^2 - 1}{\partial v_N}
\end{aligned} \tag{A.3}$$

Si calculamos las derivadas parciales respecto de una componente v_k tenemos que

$$\frac{\partial \sum_{i=1}^{N} v_i^2 - 1}{\partial v_k} = \frac{\partial \sum_{i=1}^{N} v_i^2}{\partial v_k} - \frac{\partial 1}{\partial v_k} = \sum_{i=1}^{N} \frac{\partial v_i^2}{\partial v_k} = \frac{\partial v_k^2}{\partial v_k} = 2v_k \tag{A.4}$$

y

$$\begin{aligned}
&\frac{\partial ajuste_multi_H(v_1, v_2, \ldots, v_N)}{\partial v_k} = \\
&\quad \frac{\partial \sum_{i=1}^{n_=} \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right)^2 e^{-\frac{4}{\varepsilon^2} d(x_i, x_0)}}{\partial v_k} + \\
&\qquad \frac{\partial F \sum_{i=1}^{n_{\neq}} \left(\sum_{j=1}^{N} (x\prime_{ij} - x_{0j})^2 - \left(\sum_{j=1}^{N} (x\prime_{ij} - x_{0j}) v_j \right)^2 \right) e^{-\frac{4}{\varepsilon^2} d(x\prime_i, x_0)}}{\partial v_k} =
\end{aligned}$$

$$\frac{\partial \sum_{i=1}^{n_=} \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right)^2 e^{-\frac{4}{\varepsilon^2} d(x_i, x_0)}}{\partial v_k} -$$
$$F \frac{\partial \sum_{i=1}^{n_\neq} \left(\sum_{j=1}^{N} (x\prime_{ij} - x_{0j}) v_j \right)^2 e^{-\frac{4}{\varepsilon^2} d(x\prime_i, x_0)}}{\partial v_k} = \tag{A.5}$$
$$\sum_{i=1}^{n_=} \frac{\partial \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right)^2}{\partial v_k} e^{-\frac{4}{\varepsilon^2} d(x_i, x_0)} -$$
$$F \sum_{i=1}^{n_\neq} \frac{\partial \left(\sum_{j=1}^{N} (x\prime_{ij} - x_{0j}) v_j \right)^2}{\partial v_k} e^{-\frac{4}{\varepsilon^2} d(x\prime_i, x_0)} =$$
$$\sum_{i=1}^{n_=} 2 \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right) \frac{\partial \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right)}{\partial v_k} e^{-\frac{4}{\varepsilon^2} d(x_i, x_0)} -$$
$$F \sum_{i=1}^{n_\neq} 2 \left(\sum_{j=1}^{N} (x\prime_{ij} - x_{0j}) v_j \right) \frac{\partial \left(\sum_{j=1}^{N} (x\prime_{ij} - x_{0j}) v_j \right)}{\partial v_k} e^{-\frac{4}{\varepsilon^2} d(x\prime_i, x_0)} =$$
$$\sum_{i=1}^{n_=} 2 \left(\sum_{j=1}^{N} (x_{ij} - x_{0j}) v_j \right) (x_{ik} - x_{0k}) v_k e^{-\frac{4}{\varepsilon^2} d(x_i, x_0)} -$$
$$F \sum_{i=1}^{n_\neq} 2 \left(\sum_{j=1}^{N} (x\prime_{ij} - x_{0j}) v_j \right) (x\prime_{ik} - x_{0k}) v_k e^{-\frac{4}{\varepsilon^2} d(x\prime_i, x_0)} =$$
$$2 \sum_{j=1}^{N} \left(\sum_{i=1}^{n_=} (x_{ij} - x_{0j})(x_{ik} - x_{0k}) e^{-\frac{4}{\varepsilon^2} d(x_i, x_0)} - \right.$$
$$\left. F \sum_{i=1}^{n_\neq} (x\prime_{ij} - x_{0j})(x\prime_{ik} - x_{0k}) e^{-\frac{4}{\varepsilon^2} d(x\prime_i, x_0)} \right) v_j$$

Podemos reescribir el desarrollo anterior (A.5) como:

$$\frac{\partial ajuste\_multi_H(v_1, v_2, \ldots, v_N)}{\partial v_k} = \tag{A.6}$$
$$2 \sum_{j=1}^{N} \left(\sum_{i=1}^{n} F_{clase}(x_i)(x_{ij} - x_{0j})(x_{ik} - x_{0k}) e^{-\frac{4}{\varepsilon^2} d(x_i, x_0)} \right) v_j$$

donde ahora no se distingue los x_i de los $x\prime_i$, sino que se usa la notación x_i independientemente de la clase a la que pertenezca, n es el número total de ejemplos, y la

función $F_{clase}(x_i)$ queda definida como

$$F_{clase}(x_i) = \begin{cases} 1 & \text{si } clase(x_i) = clase(x_0) \\ -F & \text{en otro caso} \end{cases} \tag{A.7}$$

Por lo tanto sustituyendo en el sistema de ecuaciones (A.3) el valor de las derivadas parciales obtenemos el siguiente sistema de ecuaciones:

$$\begin{aligned}
\sum_{i=1}^{N} v_i^2 - 1 &= 0 \\
2\sum_{j=1}^{N}\left(\sum_{i=1}^{n} F_{clase}(x_i)(x_{ij}-x_{0j})(x_{i1}-x_{01})e^{-\frac{4}{\varepsilon^2}d(x_i,x_0)}\right)v_j &= 2\lambda v_1 \\
2\sum_{j=1}^{N}\left(\sum_{i=1}^{n} F_{clase}(x_i)(x_{ij}-x_{0j})(x_{i2}-x_{02})e^{-\frac{4}{\varepsilon^2}d(x_i,x_0)}\right)v_j &= 2\lambda v_2 \\
&\vdots \\
2\sum_{j=1}^{N}\left(\sum_{i=1}^{n} F_{clase}(x_i)(x_{ij}-x_{0j})(x_{ik}-x_{0k})e^{-\frac{4}{\varepsilon^2}d(x_i,x_0)}\right)v_j &= 2\lambda v_k \\
&\vdots \\
2\sum_{j=1}^{N}\left(\sum_{i=1}^{n} F_{clase}(x_i)(x_{ij}-x_{0j})(x_{iN}-x_{0N})e^{-\frac{4}{\varepsilon^2}d(x_i,x_0)}\right)v_j &= 2\lambda v_N
\end{aligned} \tag{A.8}$$

o simplificando,

$$\begin{aligned}
\sum_{i=1}^{N} v_i^2 - 1 &= 0 \\
\sum_{j=1}^{N}\left(\sum_{i=1}^{n} F_{clase}(x_i)(x_{ij}-x_{0j})(x_{i1}-x_{01})e^{-\frac{4}{\varepsilon^2}d(x_i,x_0)}\right)v_j &= \lambda v_1 \\
\sum_{j=1}^{N}\left(\sum_{i=1}^{n} F_{clase}(x_i)(x_{ij}-x_{0j})(x_{i2}-x_{02})e^{-\frac{4}{\varepsilon^2}d(x_i,x_0)}\right)v_j &= \lambda v_2 \\
&\vdots \\
\sum_{j=1}^{N}\left(\sum_{i=1}^{n} F_{clase}(x_i)(x_{ij}-x_{0j})(x_{ik}-x_{0k})e^{-\frac{4}{\varepsilon^2}d(x_i,x_0)}\right)v_j &= \lambda v_k
\end{aligned} \tag{A.9}$$

$$\begin{array}{c} \vdots \\ \sum_{j=1}^{N}\left(\sum_{i=1}^{n} F_{clase}(x_i)(x_{ij}-x_{0j})(x_{iN}-x_{0N})\mathrm{e}^{-\frac{4}{\varepsilon^2}d(x_i,x_0)}\right)v_j \quad = \quad \lambda v_N \end{array}$$

Con lo que el problema original de minimización en un espacio de N dimensiones de la función $ajuste\_multi_H$ mostrada en la ecuación (A.1) se ha transformado en un problema en el que hay que resolver el sistema de $N+1$ ecuaciones con $N+1$ incógnitas mostrado en la ecuación (A.9).

Bibliografía

[AD91] Hussein Almuallim and Thomas G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI–91)*, volume 2, pages 547–552, Anaheim, California, 1991. AAAI Press.

[Aha92] David W. Aha. Tolerating noisy, irrelevant and novel attributes in instance–based learning algorithms. *International Journal of Man–Machine Studies*, 36:267–287, 1992.

[Aha98] David W. Aha. The omnipresence of case–based reasoning in science and application. *Knowledge–Based Systems*, 11:261–273, 1998.

[AKA91] David W. Aha, Dennis Kibler, and Marc K. Albert. Instance–based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[AMS97a] Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997. Special Issue on "Lazy Learning".

[AMS97b] Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning for control. *Artificial Intelligence Review*, 11:75–113, 1997. Special Issue on "Lazy Learning".

[AP94] A. Aamodt and Enric Plaza. Case–based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications*, 7:39–59, 1994.

[BL97] Arvin L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1–2):245–271, 1997.

[BM97] B. Baets and R. Mesiar. Pseudo-metrics and t-equivalences. *Journal Fuzzy Mathematics*, 5(2):471–481, 1997.

[BM98] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[BV92] Léon Bottou and Vladimir Vapnik. Local learning algorithms. *Neural Computation*, 4:888–900, 1992.

[CCJL03] P. Carmona, J. L. Castro, Castro-Schez J. J., and M. Laguia. Learning default fuzzy rules with general and punctual exceptions. In J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, editors, *Accuracy Improvements in Linguistic Fuzzy Modeling*, number 129 in Studies in Fuzziness and Soft Computing, pages 302–337. Springer–Verlag, 2003.

[CH67] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21–27, 1967.

[CS93] Scott Cost and Steven Salzberg. A weighted nearest algorithm for learning with symbolic features. *Machine Learning*, 10:57–78, 1993.

[Das91] B. V. Dasarathy. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, 1991.

[Dom97] Pedro Domingos. Context–sensitive feature selection for lazy learners. *Artificial Intelligence Review*, 11:227–253, 1997. Special Issue on “Lazy Learning”.

[DvdBW97] Walter Daelemans, Antal van den Bosch, and Ton Weijters. Igtree: Using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11:407–423, April 1997. Special Issue on "Lazy Learning".

[FH51] E. Fix and J. L. Hodges, Jr. Discriminatory analysis, nonparametric discrimination, consistency properties. Technical report, Randolph Field, TX: United States Air Force, School of Aviation Medicine, 1951. Technical Report 4.

[Fri94] Jerome H. Friedman. Flexible metric nearest neighbor classification. Technical report, Dept. of Statistics, Stanford University, 1994. Available by anonymous FTP from playfair.stanford.edu (see /pub/friedman/README).

[GB97] A. D. Griffiths and D. G. Bridge. Towards a theory of optimal similarity measures. In *Third UK Workshop on Case–Based Reasoning (UKCBR–97)*. Springer–Verlag, September 1997.

[HC97] Nicholas Howe and Claire Cardie. Examining locally varying weights for nearest neighbor algorithms. In *Case–Based Reasoning Research and Development: Second International Conference on Case-Based Reasoning (ICCBR–97)*. Springer–Verlag, 1997.

[HKSC95] K. Hanney, M. Keane, B. Smyth, and P. Cunningham. Systems, tasks, and adaptation knowledge: Revealing some dependencies. In Veloso M. and Aamodt A., editors, *Case-Based Reasoning Research and Development*, pages 461–470. Springer–Verlag, 1995.

[HT96] Trevor Hastie and Robert Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:607–616, 1996.

[Jan93] Cezary Z. Janikow. Fuzzy processing in decision trees. In *Proceedings of the Sixth International Symposium on Artificial Intelligence*, pages 360–367, 1993.

[Kol92] Janet L. Kolodner. An introduction to case–based reasoning. *Artificial Intelligence Review*, 6:3–34, 1992.

[Kol93] Janet L. Kolodner. *Case–Based Reasoning*. Morgan Kaufmann, 1993.

[LC01] Manuel Laguía and Juan Luis Castro. Similarity relations based on distances as fuzzy concepts. In *Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT–2001)*, 2001.

[LW97] Charles X. Ling and Handong Wang. Computing optimal attribute weight settings for nearest neighbor algorithms. *Artificial Intelligence Review*, 11:255–272, 1997. Special Issue on "Lazy Learning".

[MM97] Oded Maron and Andrew W. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11:193–225, 1997. Special Issue on "Lazy Learning".

[PLA96] E. Plaza, R. López, and E. Armengol. On the importance of similitude: An entropy-based assessment. In *Third European Workshop on Case-Based Reasoning (EWCBR–96)*. Springer–Verlag, 1996.

[Qui86] J. R. Quinlan. Induction on decision trees. *Machine Learning*, 1:81–106, 1986.

[Qui93] J. R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, 1993.

[RA95] F. Ricci and P. Avesani. Learning a local similarity metric for case–based reasoning. In *First International Conference on Case-Based Reasoning (ICCBR–95)*, pages 301–312. Springer–Verlag, 1995. Sesimbra, Portugal.

[RA99] Francesco Ricci and Paolo Avesani. Data compression and local metrics for nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):380–384, 1999.

[Rit92] M. M. Ritcher. Classification and learning of similarity measures. In *16. Jahrestagung der Gesellschaft für Klassifikation (GFKL–92)*. Springer–Verlag, 1992.

[Rit95] M. M. Ritcher. On the notion of similarity in case-based reasoning. In G. del Viertl, editor, *Mathematical and Statistical Methods in Artificial Intelligence*, pages 171–184. Springer–Verlag, 1995.

[Sal91] Steven Salzberg. A nearest hyperrectangle learning method. *Machine Learning*, 6:251–276, 1991.

[SW86] C. Stanfill and D. Waltz. Towards memory–based reasoning. *Communications of the ACM*, 29:1213–1228, 1986.

[UOH+94] M. Umano, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, and J. Kinoshita. Fuzzy decision trees by fuzzy id3 algorithm and its application to diagnosis systems. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'94)*, volume III, pages 2113–2118, june 1994.

[WAM97] Dietrich Wettschereck, David W. Aha, and Takao Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11:273–314, April 1997. Special Issue on "Lazy Learning".

[Wat96] Ian Watson. Case-based reasoning tools: an overview. In *Proceedings of the Second United Kingdom Workshop on Case-Based Reasoning (UKCBR2)*, pages 71–88, 1996.

[WD94] Dietrich Wettschereck and Thomas G. Dietterich. Locally adaptive nearest neighbor algorithms. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 184–191. Morgan Kaufmann Publishers, Inc., 1994.

[WD95] Dietrich Wettschereck and Thomas G. Dietterich. An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. *Machine Learning*, 19:5–28, 1995.

[Wet94] Dietrich Wettschereck. *A Study of Distance-Based Machine Learning Algorithms*. PhD thesis, Oregon State University, 1994.

[WK91] S. M. Weiss and C. A. Kulikowski. *Computer Systems that Learn*. Morgan Kaufmann, 1991.

[WM97] D. Randall Wilson and Tony R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Reseach (JAIR)*, 6:1–34, 1997.

[ZYY97] Jianping Zhang, Yee-Sat Yim, and Junming Yang. Intelligent selection of instances for prediction functions in lazy learning algorithms. *Artificial Intelligence Review*, 11:175–191, 1997. Special Issue on "Lazy Learning".