Capítulo 5

Clasificación
Aprendizaje supervisado

1. Introducción

El aprendizaje supervisado o clasificación es un problema de gran interés para los expertos en Inteligencia Artificial. Los datos de entrada (denominados conjunto de entrenamiento) son instancias de las clases que se desean modelar e incluyen una serie de atributos o características. El objetivo de la clasificación es obtener una descripción precisa para cada clase utilizando los atributos de los datos de entrada. El modelo así obtenido puede servir para clasificar casos cuyas clases se desconozcan o, simplemente, para comprender mejor la información de la que disponemos.

El modelo de clasificación puede construirse entrevistando a expertos. La mayor parte de los sistemas basados en conocimiento [*SBC* en castellano o *KBS* en inglés] se han construido así a pesar de la dificultad que la extracción manual del conocimiento entraña. No obstante, si se dispone de suficiente información registrada (vg: en una base de datos), el modelo de clasificación se puede construir generalizando a partir de ejemplos específicos mediante algún proceso inductivo.

Los casos de entrenamiento utilizados en la construcción del modelo de clasificación suelen expresarse en términos de un conjunto finito de propiedades o atributos con valores discretos o numéricos.

Las categorías a las que han de asignarse los distintos casos deben establecerse de antemano (aprendizaje supervisado). En general, estas clases serán disjuntas (aunque pueden establecerse jerarquías) y deberán ser discretas (para predecir atributos con valores continuos se suelen definir categorías discretas utilizando términos imprecisos propios del lenguaje natural).

Las técnicas inductivas de clasificación se basan en el descubrimiento de patrones en los datos de entrada, por lo que hemos de disponer de suficientes casos de entrenamiento (bastantes más que clases diferentes) para obtener un modelo de clasificación fiable. Se necesitan bastantes datos para poder diferenciar patrones válidos de patrones debidos a irregularidades o errores. Esta diferenciación se suele realizar utilizando alguna técnica estadística.

Si suponemos que todos los patrones a reconocer son elementos potenciales de J clases distintas denotadas ω_j, llamaremos $\Omega = \{ \omega_j \mid 1 \leq j \leq J \}$ al conjunto de las clases informacionales. En ocasiones extenderemos Ω con una clase de rechazo ω_0 a la que asignaremos todos los patrones para los que no se tiene una certeza aceptable de ser clasificados correctamente en alguna de las clases de Ω. De este modo, $\Omega^* = \{ \omega_0 \} \cup \Omega$ es el conjunto extendido de clases informacionales. Un **clasificador** o regla de clasificación es una función $d: P \rightarrow \Omega$ definida sobre el conjunto de patrones tal que para todo patrón X, $d(X) \in \Omega^*$.

2. Modelos de clasificación

2.1 Árboles de decisión

La construcción de árboles de decisión, también denominados árboles de clasificación o de identificación, es sin duda el método de aprendizaje automático más utilizado.

El dominio de aplicación de los árboles de decisión no está restringido a un ámbito concreto sino que pueden ser utilizados en diversas áreas (desde aplicaciones de diagnóstico médico hasta juegos como el ajedrez o sistemas de predicción meteorológica).

El conocimiento obtenido en el proceso de aprendizaje se representa mediante un árbol en el cual cada nodo interior contiene una pregunta sobre un atributo concreto (con un hijo por cada posible respuesta) y cada hoja del árbol se refiere a una decisión (una clasificación). Un árbol de decisión puede usarse para clasificar un caso comenzando desde su raíz y siguiendo el camino determinado por las respuestas a las preguntas de los nodos internos hasta que encontremos una hoja del árbol.

La construcción de los árboles de decisión se hace recursivamente de forma descendente (se parte de conceptos generales que se van especificando conforme se desciende en el árbol), por lo que se emplea el acrónimo TDIDT [*Top-Down Induction on Decision Trees*] para referirse a la familia completa de algoritmos de este tipo.

La familia de algoritmos TDIDT abarca desde algoritmos ya clásicos de IA como CLS [*Concept Learning System*], ID3, C4.5 o CART [*Classification And Regression Trees*] hasta algoritmos optimizados como SLIQ o SPRINT, dos algoritmos desarrollados en el IBM Almaden Research Center que se usan en *Data Mining*.

Los algoritmos TDIDT suelen presuponer que no existe ruido en los datos de entrada e intentan alcanzar una descripción perfecta de los mismos. Esto suele ser contraproducente en problemas reales, donde se necesitan métodos capaces de manejar información con ruido y mecanismos que eviten el sobreaprendizaje [*overfitting*]. Sin embargo, las técnicas de poda (como las empleadas en ASSISTANT o C4.5) han demostrado ser muy útiles en este sentido. Una vez construido el árbol de decisión completo que se adapta perfectamente a los datos del conjunto de entrenamiento, se podan aquellas ramas del árbol con menor capacidad predictiva.

En resumen, la representación del conocimiento mediante árboles de decisión es bastante simple y, a pesar de carecer de la expresividad de las redes semánticas o de la lógica de primer orden, se utiliza muy a menudo para resolver problemas de clasificación de todo tipo.

2.2 Reglas de producción

Conforme el tamaño los árboles de decisión aumenta, su inteligibilidad disminuye. Cuando el problema de clasificación es complejo, el árbol de decisión generado es tan grande que ni siquiera los expertos pueden comprender el modelo de clasificación construido (ni siquiera simplificándolo al podar el árbol).

Shapiro propuso descomponer un árbol de decisión complejo en una jerarquía de pequeños árboles de decisión para obtener un modelo más comprensible [*structured induction*]. Sin embargo, es mucho más sencillo expresar el árbol de decisión construido como un conjunto de reglas de producción, una forma de representación del conocimiento más inteligible que los árboles.

Las reglas de producción se pueden derivar de un árbol de decisión con facilidad. El algoritmo que nos permite realizar este cambio de modelo de representación es muy sencillo: de cada camino desde la raíz del árbol hasta un nodo hoja se deriva una regla cuyo antecedente es una conjunción de literales relativos a los valores de los atributos situados en los nodos internos del árbol y cuyo consecuente es la decisión a la que hace referencia la hoja del árbol (la clasificación realizada).

También existen otros métodos de clasificación que obtienen reglas de producción directamente, sin necesidad de construir previamente un árbol de decisión. Estas técnicas, más ineficientes, suelen emplear estrategias de búsqueda heurística como la búsqueda dirigida, una variante de la búsqueda primero el mejor.

Michalski y sus colaboradores desarrollaron, bajo el nombre de metodología STAR, un conjunto de técnicas de aprendizaje inductivo incremental basadas en la utilización de expresiones lógicas en forma normal disyuntiva (modelo de representación más expresivo que el empleado por los algoritmos de construcción de árboles de decisión). Estas expresiones lógicas describen conceptos y se pueden utilizar directamente como reglas de clasificación. Entre los algoritmos desarrollados en los años 80 en el entorno de Michalski destacan INDUCE y AQ, del que existen múltiples variantes.

Con posterioridad a los trabajos de Michalski, Peter Clark y su equipo propusieron el algoritmo CN2 en 1989. Este algoritmo intenta combinar adecuadamente las mejores cualidades de los algoritmos TDIDT y AQ: CN2 trata de combinar la eficiencia y el manejo de información con ruido que permite la familia de algoritmos TDIDT con la flexibilidad de la familia AQ en su estrategia de búsqueda de reglas.

2.3 Reglas de asociación

Muchas veces en la vida real no se pueden construir modelos completos que nos permitan una clasificación perfecta de todos los casos con los que uno se pueda encontrar. A veces hay que conformarse con descubrir modelos aproximados, los cuales contemplan algunas características de las distintas clases sin que el modelo abarque todas las clases posibles ni todos los casos particulares de una clase determinada.

La construcción de un modelo de clasificación completo puede no ser factible cuando hemos de tratar con una gran cantidad de atributos, cuando muchos valores son desconocidos, cuando unos atributos deben modelarse en función de otros o cuando el número de casos de entrenamiento es excesivamente elevado.

Los árboles de decisión no son muy adecuados para tratar con información incompleta (valores desconocidos en atributos de los casos de entrenamiento) y resultan problemáticos cuando unos atributos son función de otros. Las redes neuronales tampoco son apropiadas cuando tenemos información incompleta y, además, su entrenamiento puede llegar a consumir demasiado tiempo. Finalmente, las técnicas empleadas en ILP [*Inductive Logic Programming*] suelen ser muy poco eficientes.

Por su parte, un modelo de clasificación parcial intenta descubrir características comunes a los distintos casos de cada clase sin la necesidad de formar un modelo predictivo completo. La extracción de reglas de asociación puede ser útil para resolver problemas de clasificación parcial donde las técnicas de clasificación clásicas no son efectivas.

Como se vio en el capítulo dedicado al uso de las reglas de asociación, el problema de la clasificación parcial se puede resolver usando reglas de asociación de dos formas diferentes: dividiendo el conjunto de casos de entrenamiento (un subconjunto por clase) o considerando la clase como un atributo más. En cualquiera de las dos situaciones anteriores, para cada regla de asociación A⇒C obtenida ha de calcularse su riesgo relativo utilizando la siguiente expresión:

$$r(A \Rightarrow C) = \frac{P(C|A)}{P(C|\neg A)}$$

Cuando el cociente anterior es elevado se puede considerar interesante la regla $A \Rightarrow C$. Intuitivamente se puede comprender con facilidad el sentido de la ecuación que define el riesgo relativo de una regla. Por ejemplo, carecería de interés clasificar una enfermedad atendiendo a síntomas que no siempre se manifiestan asociados a esa enfermedad y sería trascendental identificar síntomas específicos de una enfermedad (aunque éstos sean muy poco frecuentes).

También se pueden construir modelos de clasificación híbridos basados, en mayor o menor medida, en reglas de asociación. Por ejemplo, ART [*Association Rule Tree*] es una propuesta de Juan Carlos Cubero que intenta aprovechar las mejores cualidades de las reglas de asociación como modelo de clasificación parcial con la construcción descendente de árboles de decisión.

2.4 Clasificadores basados en medidas de similaridad [*Instance-based classifiers*]

Una forma básica de clasificar un caso es asignarle la misma clase que a otro caso similar cuya clasificación es conocida. Entre ellos destacan los métodos de clasificación por el vecino más cercano *k-NN*, donde *k* es impar (no tiene sentido probar con valores pares de *k* porque el error asociado a la regla *k-NN* es el mismo para *2x* y *2x-1*).

A la hora de construir clasificadores de este tipo han de resolverse algunas cuestiones previas, entre las que destacan:

¿Cómo se realiza la clasificación?

Podemos asignarle a un caso la clase del caso almacenado más similar (*1-NN*) o utilizar los grados de similaridad con distintos casos almacenados a la hora de realizar la predicción (como en el método *k-NN*).

¿Qué casos deben almacenarse?

Lo ideal sería almacenar aquellos casos típicos que recojan toda la información relevante necesaria para poder realizar una buena clasificación. Almacenar todos los casos conocidos haría muy ineficiente el funcionamiento del clasificador.

Los métodos de edición y condensado se utilizan para mejorar el rendimiento de este tipo de clasificadores. Los métodos de edición (como la edición de Wilson o el Multiedit) intentan eliminar los patrones mal etiquetados que puedan aparecer cerca de las fronteras de decisión. Por su parte, los métodos de condensado (como el algoritmo de Hart) procuran reducir el número de muestras del conjunto de entrenamiento sin que esto afecte a la calidad del clasificador construido.

Para reducir la complejidad computacional del problema se pueden emplear métodos de condensado o, simplemente, utilizar algoritmos optimizados como el de Fukunaga y Narendra para la obtención del vecino más cercano.

¿Cómo se mide la similaridad entre distintos casos?

Cuando los atributos son numéricos, se suele calcular la similaridad entre casos utilizando alguna métrica de distancia (que es una medida de disimilaridad), como la distancia euclídea o la distancia de Mahalanobis. Por ejemplo, se puede utilizar la raíz cuadrada de las suma de los cuadrados de las diferencias de los valores de los atributos (usando factores de escala para que la influencia de todos atributos sea similar). Cuando los atributos son discretos, establecer una medida de similaridad es bastante más problemático.

Además, si hay atributos irrelevantes, se corre el riesgo de considerar muy diferentes casos que sólo difieren en los valores que toman atributos irrelevantes para la clasificación.

2.5 Técnicas estadísticas

Existen muchas técnicas estadísticas aplicables a problemas de clasificación. Estas técnicas suelen ser paramétricas. Se asume la forma del modelo y, a partir de los datos de entrenamiento, se hallan los valores adecuados para los parámetros del modelo.

Por ejemplo, un clasificador lineal asume que la clasificación puede realizarse mediante una combinación lineal de los valores de los atributos y emplea la combinación lineal que mejor se adapte al conjunto de casos de entrenamiento a la hora de clasificar nuevos casos.

En determinadas circunstancias, un clasificador cuadrático puede obtener mejores resultados que un clasificador lineal simple. Sin embargo, el ADC [Análisis Discriminante Cuadrático] requiere muchas más muestras de entrenamiento que el ADL [Análisis Discriminante Lineal] para obtener resultados similares ya que es más sensible al número de muestras requeridas.

Pero no siempre es mejor un clasificador cuadrático. Para determinados conjuntos de datos el ADC ni siquiera se puede aplicar, como sucede con un conjunto estándar de datos de la ionosfera formado por 351 patrones de 34 atributos cada uno (John Hopkins University Ionosphere Database). En estos casos, no disponemos de suficientes muestras para estimar la matriz de covarianza de los datos adecuadamente (de hecho, para los datos de la ionosfera ni siquiera podemos calcularle su inversa).

Aunque en teoría el error de Bayes decrece conforme la dimensionalidad de los datos se incrementa, en la práctica disponemos de un conjunto fijo y finito de muestras para construir el clasificador (los estimadores están sesgados por las muestras disponibles). La bondad conseguida con un clasificador aumenta con la dimensionalidad de los datos hasta cierto punto, a partir del cual decrece conforme se incorporan nuevas variables (fenómeno de Hughes).

El problema anterior podría solucionarse consiguiendo más muestras de entrenamiento (lo cual no suele ser posible) o eligiendo un subespacio del espacio de patrones (usando técnicas de selección de características).

2.6 Redes neuronales

Las redes neuronales representan una de las aportaciones más importantes que las ciencias biológicas han realizado al campo de la Inteligencia Artificial. Su característica más importante es su capacidad de aprender a partir de ejemplos, que les permite generalizar sin tener que formalizar el conocimiento adquirido.

Con las redes neuronales se intenta expresar la solución de problemas complejos no como una secuencia de pasos, sino como la evolución de unos sistemas de computación inspirados en el funcionamiento del cerebro humano, los cuales no son sino la combinación de una gran cantidad de elementos simples de proceso (*neuronas*) interconectados que operan en paralelo.

Es importante destacar que, aunque se pueden desarrollar aplicaciones mediante programas de simulación, codificando algoritmos de funcionamiento y aprendizaje, la verdadera potencia de las redes neuronales se pone de manifiesto mediante su implementación física en hardware.

Una ventaja de las redes neuronales respecto a otros modelos de clasificación es que también pueden utilizarse para predecir valores reales y no sólo clases discretas.

Sin embargo, el modelo de clasificación construido al entrenar la red suele ser totalmente incomprensible para un experto humano (lo que dificulta que el clasificador goce de la confianza de los expertos). El conjunto de métodos LVQ (de aprendizaje por cuantificación vectorial) constituye una honrosa excepción al destacar por la sencillez de las heurísticas que utiliza.

2.6.1 LVQ

Los métodos LVQ [*Linear-Vector Quantization*] son métodos de aprendizaje adaptativo basados en los mapas autoorganizativos [*SOM: Self-Organizing Maps*] de Kohonen. Se caracterizan por utilizar un número fijo y relativamente bajo de prototipos para aproximar las funciones de densidad de probabilidad de las distintas clases.

Dada una secuencia de observaciones vectoriales (patrones), se selecciona un conjunto inicial de vectores de referencia (codebooks o prototipos). Iterativamente, se selecciona una observación X y se actualiza el conjunto de prototipos de forma que case mejor con X.

Una vez finalizado el proceso de construcción del conjunto de prototipos (es decir, el entrenamiento de la red), las observaciones se clasificarán utilizando la regla *1-NN* (buscando el vecino más cercano en el conjunto de vectores de referencia).

2.7 Algoritmos genéticos

Los algoritmos genéticos están inspirados en la Naturaleza, en la teoría de la evolución descrita por Charles Darwin en su libro "Sobre el Origen de las Especies por medio de la Selección Natural", escrito 20 años después del viaje de su autor por las islas Galápagos en el Beagle. La hipótesis de Darwin (y de Wallace, que llegó a las mismas conclusiones independientemente) es que pequeños cambios heredables en los seres vivos y la selección son los dos hechos que provocan el cambio en la Naturaleza y la generación de nuevas especies. Fue Lendel quien descubrió que los caracteres se heredaban de forma discreta, y que se tomaban del padre o de la madre, dependiendo de su carácter dominante o recesivo. A estos caracteres que podían tomar diferentes valores se les llamaron genes, y a los valores que podían tomar, alelos. En los seres vivos, los genes están en los cromosomas.

En la evolución natural, los mecanismos de cambio alteran la proporción de alelos de un tipo determinado en una población, y se dividen en dos tipos: los que disminuyen la variabilidad (la selección natural y la deriva genética), y los que la aumentan (la mutación, la poliploidía, la recombinación o cruce y el flujo genético).

A principios de los 60, en la Universidad de Michigan en Ann Arbor, las ideas de John Holland comenzaron a desarrollarse y a dar frutos. Leyendo un libro escrito por un biólogo evolucionista, R.A. Fisher, titulado "La teoría genética de la selección natural", aprendió que la evolución era una forma de adaptación más potente que el simple aprendizaje y tomó la decisión de aplicar estas ideas para desarrollar programas bien adaptados para un fin determinado. Los objetivos de su investigación fueron dos: imitar los procesos adaptativos de los sistemas naturales y diseñar sistemas artificiales (programas) que retengan los mecanismos de los sistemas naturales.

Los algoritmos evolutivos tratan de imitar los mecanismos de la evolución para resolver problemas. La aplicación de un algoritmo genético consiste en hallar de qué parámetros depende el problema, codificarlos en un cromosoma y aplicar los métodos de la evolución (selección y reproducción sexual con intercambio de información y alteraciones que generen diversidad). La mayoría de las veces una codificación correcta es la clave de una buena resolución del problema.

Los algoritmos genéticos en sí son métodos de optimización. En el algoritmo genético va implícito el método para resolver el problema. Un algoritmo genético es independiente del problema, lo cual lo hace robusto, por ser útil para cualquier problema, pero a la vez débil, pues no está especializado en ninguno.

Como método general de resolución de problemas que son, los algoritmos genéticos también pueden utilizarse para resolver problemas de clasificación.

Al igual que las redes neuronales y los clasificadores basados en medidas de similaridad, los clasificadores construidos utilizando algoritmos genéticos suelen destacar porque su rendimiento no se ve excesivamente afectado por la aparición errores en los casos de entrenamiento (lo que sí ocurre con determinados modelos simbólicos.

3. Bibliografía

Kamal Ali, Stefanos Manganaris & Ramakrishnan Skirant
"Partial Classification using Association Rules"
KDD'97 [International Conference on Knowledge Discovery in Databases and Data Mining]
California, USA, 1997

En esta ponencia, los investigadores de IBM utilizan reglas de asociación para realizar clasificaciones parciales (clasificaciones en las que se descubren modelos que pueden no cubrir todas las clases ni todos los casos de una clase concreta).

R. Duda & P. Hart
"Pattern Classification and Scene Analysis"
John Wiley and Sons, 1973

Los clasificadores paramétricos de los que se habla en la sección dedicada a técnicas estadísticas aparecen en la sección *2.8* de este famoso libro de Reconocimiento de Formas.

J. Ross Quinlan
"C4.5: Programs for Machine Learning"
Morgan Kaufman, 1993

En este libro se expone el *C4.5*, uno de los muchos algoritmos derivados de *ID3*. Forma parte de la familia de sistemas de aprendizaje *TDIDT* [*Top-Down Induction of Decision Trees*], que se basa en la construcción recursiva de árboles de decisión.

Elaine Rich & Kevin Knight
"Inteligencia Artificial"
McGraw-Hill Interamericana de España, 1994 [2ª edición]

Uno de los libros de IA más populares. Le dedica un capítulo entero a técnicas de aprendizaje. Aunque no habla de la metodología STAR de Michalki, sí trata los espacios de versiones de Mitchell (que siguen ideas similares).

Sabrina Sestito & Tharam S. Dillon
"Automated Knowledge Acquisition"
Prentice-Hall Series in Computer Science and Engineering, Australia, 1994

Este libro de ML [Machine Learning] le dedica uno de sus capítulos a la metodología STAR ideada por Michalski y sus colaboradores, el tercero para ser más concretos.