Conceptos básicos sobre Clasificación

Minería de Datos e Inteligencia de Negocios

Juan Luis Restituyo

Introducción

La clasificación, que es la tarea de asignar objetos a una de varias categorías predefinidas, es un problema generalizado que abarca muchas aplicaciones diversas. Este capítulo presenta los conceptos básicos de clasificación, describe algunos de los problemas clave, como el sobreajuste de modelos, y presenta métodos para evaluar y comparar el rendimiento de una técnica de clasificación.

Clasificación: La clasificación es la tarea de aprender una función de destino f que asigna cada conjunto de atributos x a una de las etiquetas de clases predefinidas y. La función de destino también se conoce informalmente como un modelo de clasificación. Un modelo de clasificación es útil para los siguientes propósitos:

Modelado descriptivo: un modelo de clasificación puede servir como una herramienta explicativa para distinguir entre objetos de diferentes clases. Por ejemplo, sería útil, tanto para los biólogos como para otros, tener un modelo descriptivo que resuma los datos mostrados en la siguiente tabla y explique qué características definen a un vertebrado como mamífero, reptil, ave, pez o anfibio.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

Modelado predictivo: un modelo de clasificación también se puede usar para predecir el nivel de clase de registros desconocidos. Como se muestra en la figura siguiente, un modelo de clasificación puede tratarse como una caja negra que asigna automáticamente una etiqueta de clase cuando se presenta con el conjunto de atributos de un registro desconocido. Supongamos que nos dan las siguientes características de una criatura conocida como monstruo de Gila:

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
gila monster	cold-blooded	scales	no	no	no	yes	yes	?

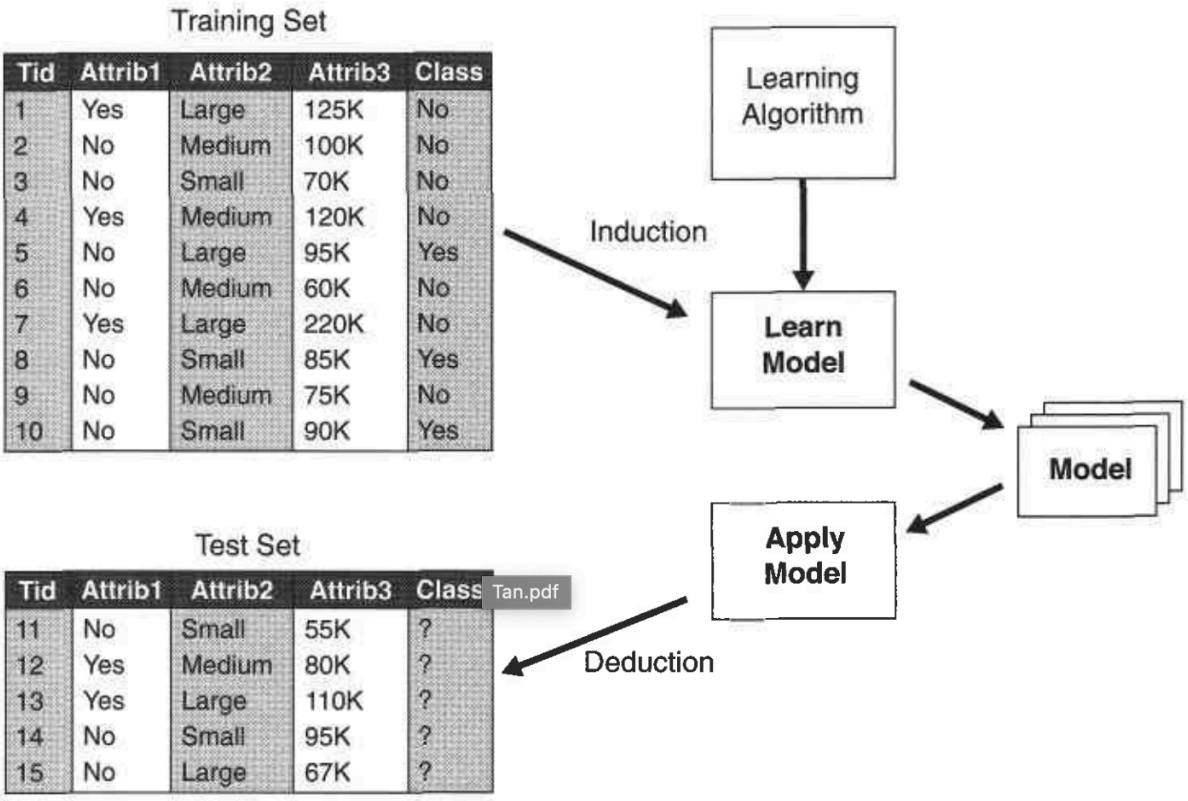
Podemos usar un modelo de clasificación construido a partir del conjunto de datos para determinar la clase a la que pertenece la criatura tal como se hizo en la tabla anterior que clasifica a los vertebrados.

Las técnicas de clasificación son las más adecuadas para predecir o describir conjuntos de datos con categorías binarias o nominales. Son menos efectivos para las categorías ordinales (por ejemplo, para clasificar a una persona como miembro de un grupo de ingresos altos, medios o bajos) porque no consideran el orden implícito entre las categorías. También se ignoran otras formas de relaciones, como las relaciones de superclase de subclase entre categorías (por ejemplo, los humanos y los simios son primates, que a su vez, es una subclase de mamíferos).

Enfoque general para resolver un problema de clasificación

Una técnica de clasificación (o clasificador) es un enfoque sistemático para construir modelos de clasificación a partir de un conjunto de datos de entrada. Los ejemplos incluyen clasificadores de árbol de decisión, clasificadores basados en reglas, redes neuronales, máquinas de vectores de soporte y clasificadores ingenuos de Bayes. Cada técnica emplea un algoritmo de aprendizaje para identificar el modelo que mejor se adapta a la relación entre el conjunto de atributos y la etiqueta de clase de los datos de entrada. El modelo generado por un algoritmo de aprendizaje debe ajustarse bien a los datos de entrada y predecir correctamente las etiquetas de clase de los registros que nunca ha visto. Por lo tanto, un objetivo clave del algoritmo de aprendizaje es construir modelos con buena capacidad de generalización; es decir, modelos que predicen con precisión las etiquetas de clase de registros previamente desconocidos.

La siguiente figura muestra un enfoque general para resolver problemas de clasificación. Primero, se debe proporcionar un conjunto de entrenamiento que consiste en registros cuyas etiquetas de clase son conocidas. El conjunto de entrenamiento se utiliza para construir un modelo de clasificación, que se aplica posteriormente al conjunto de prueba, que consta de registros con etiquetas de clase desconocidas.



La evaluación del rendimiento de un modelo de clasificación se basa en los recuentos de registros de prueba correctamente e incorrectamente pronosticados por el modelo. Estos conteos se tabulan en una tabla conocida como matriz de confusión.

		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

Aunque una matriz de confusión proporciona la información necesaria para determinar qué tan bien funciona un modelo de clasificación, resumir esta información con un solo número haría más conveniente comparar el rendimiento de diferentes modelos. Esto se puede hacer utilizando una métrica de rendimiento como la precisión, que se define de la siguiente manera:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

De manera equivalente, el rendimiento de un modelo puede expresarse como tasa de error, que se obtiene mediante la siguiente ecuación:

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

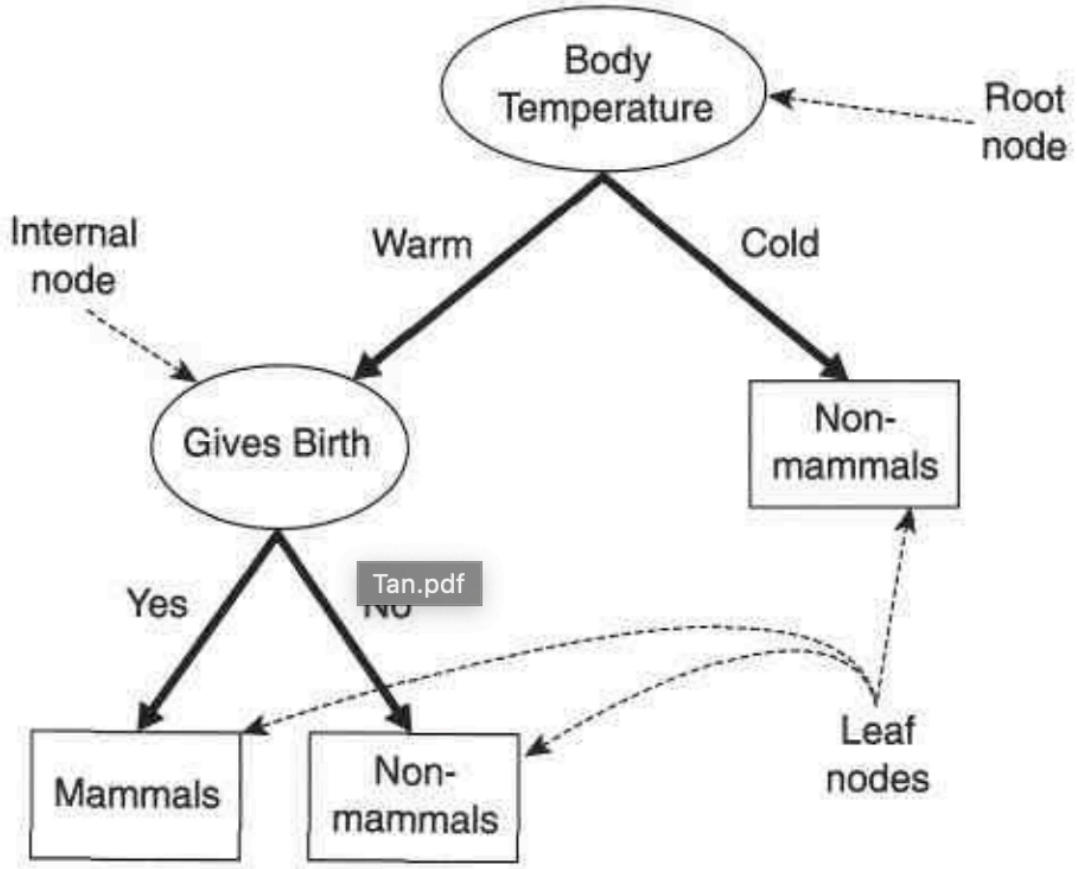
La mayoría de los algoritmos de clasificación buscan modelos que alcancen la mayor precisión, o equivalentemente, la tasa de error más baja cuando se aplican al conjunto de pruebas.

Arboles de decisión

Para ilustrar cómo funciona la clasificación con un árbol de decisión, considere una versión más simple del problema de clasificación de vertebrados descrito en la sección anterior. En lugar de clasificar a los vertebrados en cinco grupos distintos de especies, los asignamos a dos categorías: mamíferos y no mamíferos.

Supongamos que los científicos descubren una nueva especie. ¿Cómo podemos saber si es un mamífero o un no mamífero? Un enfoque es plantear una serie de preguntas sobre las características de la especie. La primera pregunta que podemos hacernos es si la especie es de sangre fría o caliente. Si es de sangre fría, entonces definitivamente no es un mamífero. De lo contrario, es un ave o un mamífero. En este último caso, debemos hacer una pregunta de seguimiento: ¿Las hembras de la especie dan a luz a sus crías? Aquellos que sí dan a luz son definitivamente mamíferos, mientras que aquellos que no, no son mamíferos (con la excepción de los mamíferos que ponen huevos, como el ornitorrinco y el oso hormiguero espinoso).

El ejemplo anterior ilustra cómo podemos resolver un problema de clasificación haciendo una serie de preguntas cuidadosamente elaboradas sobre los atributos del registro de prueba. Cada vez que recibimos una respuesta, se hace una pregunta de seguimiento hasta que llegamos a una conclusión sobre la etiqueta de clase del registro. La serie de preguntas y sus posibles respuestas se pueden organizar en forma de un árbol de decisión, que es una estructura jerárquica que consta de nodos y bordes dirigidos. La siguiente figura muestra el árbol de decisión para el problema de clasificación de mamíferos.

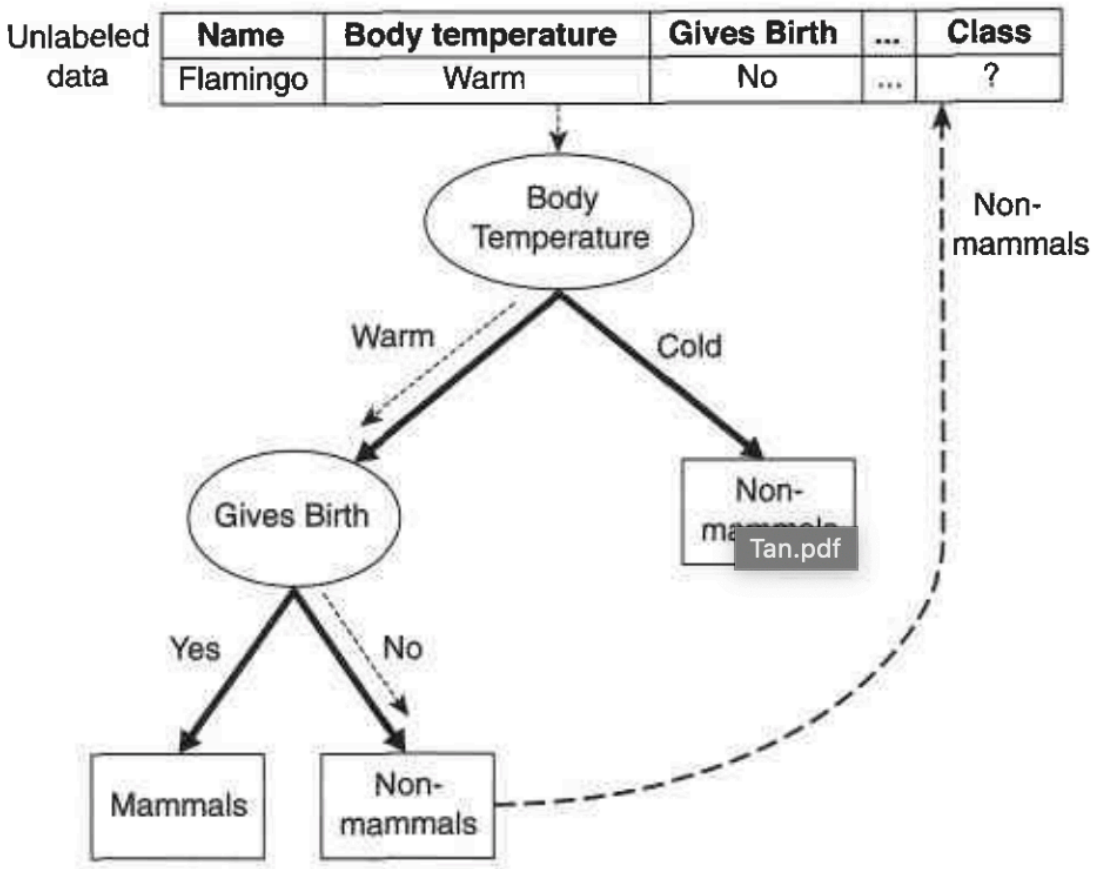


El árbol tiene tres tipos de nodos:

- Un nodo raíz que no tiene bordes entrantes y cero o más bordes salientes.
- Nodos internos, cada uno de los cuales tiene exactamente un borde entrante y dos o más bordes salientes.
- Nodos hoja o terminales, cada uno de los cuales tiene exactamente un borde entrante y no tiene bordes salientes

En un árbol de decisión, a cada nodo hoja se le asigna una etiqueta de clase. Los nodos no terminales, que incluyen la raíz y otros nodos internos, contienen condiciones de prueba de atributos para separar registros que tienen características diferentes. Por ejemplo, el nodo de la raíz que se muestra en la Figura anterior utiliza el atributo Temperatura del cuerpo para separar los vertebrados de sangre caliente de los de sangre fría. Dado que todos los vertebrados de sangre fría no son mamíferos, se crea un nodo de hoja etiquetado como No mamíferos como el hijo derecho del nodo raíz. Si el vertebrado es de sangre caliente, un atributo posterior, Da Nacimiento, se usa para distinguir a los mamíferos de otras criaturas de sangre caliente, que en su mayoría son aves.

La clasificación de un registro de prueba es sencilla una vez que se ha construido un árbol de decisión. A partir del nodo raíz, aplicamos la condición de prueba al registro y seguimos la rama apropiada en función del resultado de la prueba. Esto nos llevará a otro nodo interno, para el cual se aplica una nueva condición de prueba, o a un nodo hoja. La etiqueta de clase asociada con el nodo hoja se asigna al registro. Como ilustración, la siguiente traza la ruta en el árbol de decisión que se usa para predecir la etiqueta de clase de un flamenco. La ruta termina en un nodo hoja etiquetado como No mamífero.



Cómo construir un árbol de decisión

En principio, hay exponencialmente muchos árboles de decisión que pueden construirse a partir de un conjunto dado de atributos. Si bien algunos de los árboles son más precisos que otros, encontrar el árbol óptimo no es factible computacionalmente debido al tamaño exponencial del espacio de búsqueda. Sin embargo, se han desarrollado algoritmos eficientes para inducir un árbol de decisiones razonablemente preciso, aunque no óptimo, en un tiempo razonable. Estos algoritmos generalmente emplean una estrategia codiciosa que hace crecer un árbol de decisiones al tomar una serie de decisiones óptimas a nivel local sobre qué atributo usar para la partición de los datos. Uno de estos algoritmos es el algoritmo Huntts, que es la base de muchos algoritmos de inducción de árbol de decisión existentes

El algoritmo de Hunt

En el algoritmo de Hunt, un árbol de decisión crece de forma recursiva al dividir los registros de entrenamiento en subconjuntos sucesivamente más puros. Sea Dl el conjunto de registros de entrenamiento asociados con los nodos t y g: {At, U2,. . . , A "} sean las etiquetas de clase. La siguiente es una definición recursiva del algoritmo de Hunt.

Paso 1: Si todos los registros en Dt pertenecen a la misma clase yt, entonces t es un nodo hoja etiquetado como yt.

Paso 2: Si Dt contiene registros que pertenecen a más de una clase, se selecciona una condición de prueba de atributo para particionar los registros en subconjuntos más pequeños. Se crea un nodo secundario para cada resultado de la condición de prueba y los registros en Dt se distribuyen a los hijos según los resultados. El algoritmo se aplica recursivamente a cada nodo secundario.

	binary	categorical	continuous	class
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training set for predicting borrowers who will default on loan payments.

Para ilustrar cómo funciona el algoritmo, considere el problema de predecir si un solicitante de préstamo pagará sus obligaciones de préstamo o se convertirá en delincuente, y posteriormente incumplirá con su préstamo. Se puede construir un conjunto de entrenamiento para este problema al examinar los registros de prestatarios anteriores. En el ejemplo que se muestra en la figura, cada registro contiene la información personal de un prestatario junto con una etiqueta de clase que indica si el prestatario ha incumplido con los pagos de los préstamos.
El árbol inicial para el problema de clasificación contiene un solo nodo con la etiqueta de clase Predeterminado. Para este ejemplo vemos que la mayoría de los prestatarios reembolsaron sus préstamos con éxito. Sin embargo, el árbol necesita ser refinado ya que el nodo raíz contiene registros de ambas clases. Los registros se dividen posteriormente en subconjuntos más pequeños según los resultados de la condición de prueba del propietario de la vivienda. La justificación para elegir esta condición de prueba de atributo se discutirá más adelante. Por ahora, asumiremos que este es el mejor criterio para dividir los datos en este momento. El algoritmo de Hunt se aplica recursivamente a cada hijo del nodo raíz. Para el niño correcto, debemos continuar aplicando el paso recursivo del algoritmo de Hunt hasta que todos los registros pertenezcan a la misma clase.

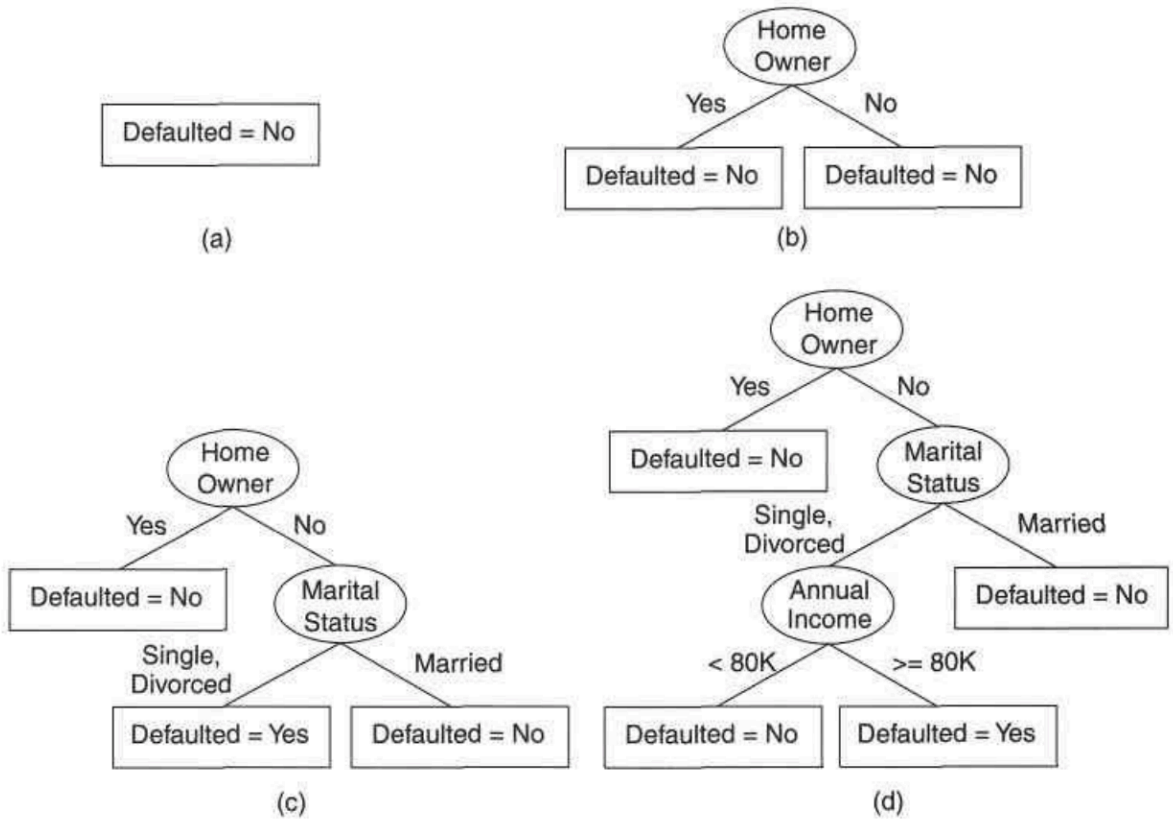


Figure 4.7. Hunt's algorithm for inducing decision trees.

El algoritmo de Hunt funcionará si cada combinación de valores de atributo está presente en los datos de entrenamiento y cada combinación tiene una etiqueta de clase única. Estas

suposiciones son demasiado estrictas para su uso en la mayoría de las situaciones prácticas. Se necesitan condiciones adicionales para manejar los siguientes casos.

Es posible que algunos de los nodos secundarios creados en el Paso 2 estén vacíos; es decir, no hay registros asociados con estos nodos. Esto puede suceder si ninguno de los registros de entrenamiento tiene la combinación de valores de atributos asociados con dichos nodos. En este caso, el nodo se declara nodo hoja con la misma etiqueta de clase que la mayoría de los registros de entrenamiento asociados con su nodo principal.

En el Paso 2, si todos los registros asociados con D; tienen valores de atributos idénticos (excepto la etiqueta de clase), entonces no es posible dividir estos registros más. En este caso, el nodo se declara un nodo hoja con la misma etiqueta de clase que la clase mayoritaria de registros de entrenamiento asociados con este nodo.

Problemas de diseño de la inducción del árbol de decisión

Un algoritmo de aprendizaje para inducir árboles de decisión debe abordar los dos problemas siguientes:

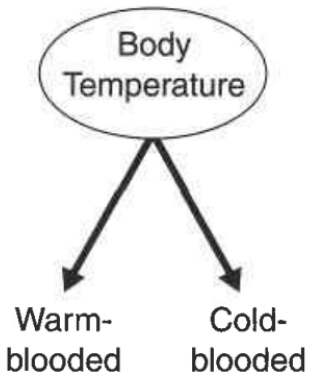
¿Cómo se deben dividir los registros de entrenamiento? Cada paso recursivo del proceso de crecimiento de árboles debe seleccionar una condición de prueba de atributo para dividir los registros en subconjuntos más pequeños. Para implementar este paso, el algoritmo debe proporcionar un método para especificar la condición de prueba para diferentes tipos de atributos, así como una medida objetiva para evaluar la bondad de cada condición de prueba.

¿Cómo debe detenerse el procedimiento de división? Se necesita una condición de parada para terminar el proceso de crecimiento de árboles. Una posible estrategia es continuar expandiendo un nodo hasta que todos los registros pertenezcan a la misma clase o todos los registros tengan valores de atributos idénticos. Aunque ambas condiciones son suficientes para detener cualquier algoritmo de inducción del árbol de decisión, se pueden imponer otros criterios para permitir que el procedimiento de crecimiento de árboles termine antes.

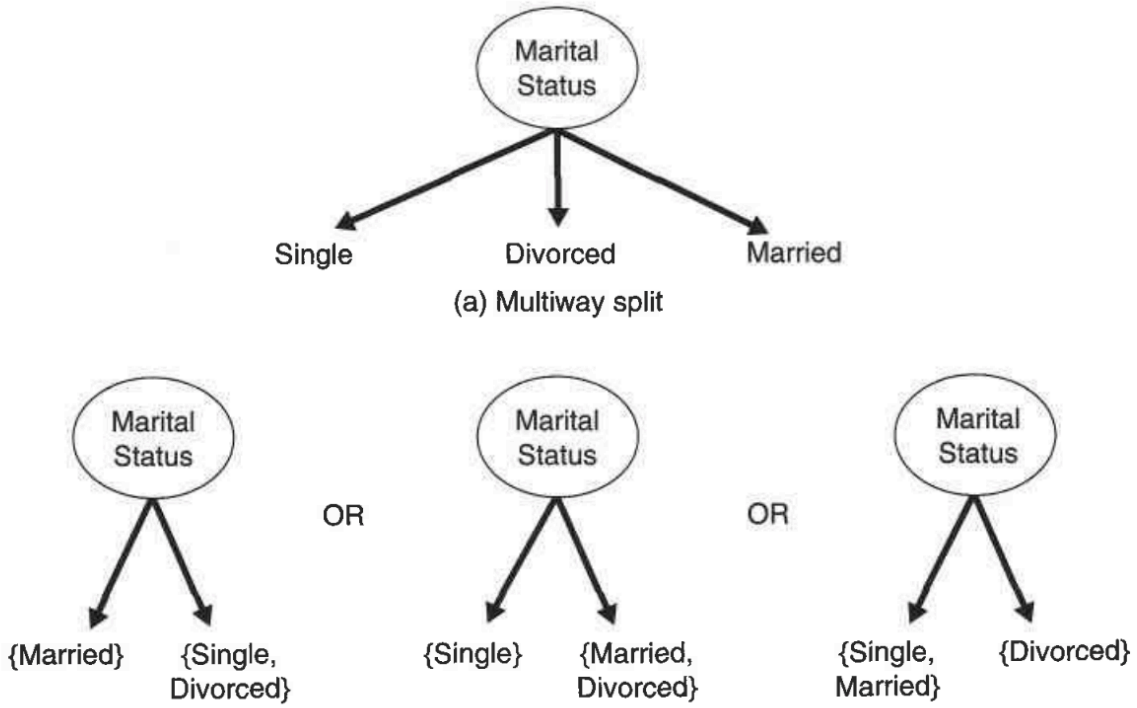
Métodos para expresar condiciones de prueba de atributo

Los algoritmos de inducción del árbol de decisión deben proporcionar un método para expresar una condición de prueba de atributo y sus resultados correspondientes para diferentes tipos de atributo

Atributos binarios: la condición de prueba para un atributo binario genera dos resultados potenciales.



Atributos nominales: dado que un atributo nominal puede tener muchos valores, su condición de prueba se puede expresar de dos maneras, el número de resultados depende del número de valores distintos para el atributo correspondiente. Por otro lado, algunos algoritmos de árbol de decisión, como CART, producen solo divisiones binarias.



Atributos ordinales Los atributos ordinales también pueden producir divisiones binarias o de múltiples vías. Los valores de atributo ordinales se pueden agrupar siempre que la agrupación no infrinja la propiedad de orden de los valores de atributo.

Medidas para seleccionar la mejor división

Hay muchas medidas que se pueden usar para determinar la mejor manera de dividir los registros. Estas medidas se definen en términos de la distribución de clase de los registros antes y después de la división.

Las medidas desarrolladas para seleccionar la mejor división a menudo se basan en el grado de impureza de los nodos secundarios. Cuanto menor sea el grado de impureza, más sesgada será la distribución de clase.

Características de la inducción del árbol de decisión

El siguiente es un resumen de las características importantes de los algoritmos de inducción del árbol de decisión.

La inducción del árbol de decisión es un enfoque no paramétrico para construir modelos de clasificación. En otras palabras, no requiere suposiciones previas con respecto al tipo de distribuciones de probabilidad satisfechas por la clase y otros atributos.

Encontrar un árbol de decisión óptimo es un problema NP-completo. Muchos algoritmos de árbol de decisión emplean un enfoque basado en heurística para guiar su búsqueda en el vasto espacio de hipótesis.

Las técnicas desarrolladas para construir árboles de decisión son computacionalmente baratas, lo que hace posible construir modelos rápidamente incluso cuando el tamaño del conjunto de entrenamiento es muy grande. Además, una vez que se ha construido un árbol de decisión, la clasificación de un registro de prueba es extremadamente rápida, con la peor complejidad posible de O (w), donde, w es la profundidad máxima del árbol.

Los árboles de decisión, especialmente los árboles de menor tamaño, son relativamente fáciles de interpretar. Las precisiones de los árboles también son comparables a otras técnicas de clasificación para muchos conjuntos de datos simples.

Los árboles de decisión proporcionan una representación expresiva para el aprendizaje de funciones de evaluación discreta. Sin embargo, no generalizan bien a ciertos tipos de problemas booleanos. Un ejemplo notable es la función de paridad, cuyo valor es 0 (1) cuando hay un número impar (par) de atributos booleanos con valueTrue. El modelado preciso de tal función requiere un árbol de decisión completo con segundos nodos, donde d es el número de atributos booleanos.

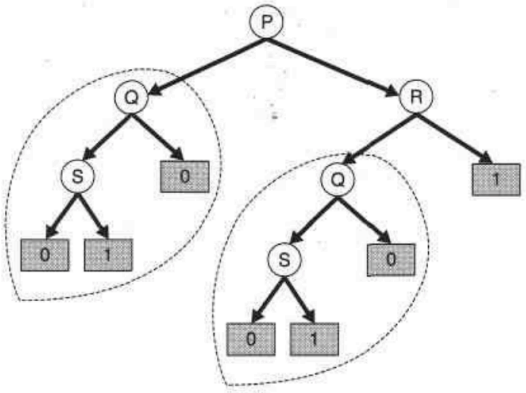
Los algoritmos del árbol de decisión son bastante robustos ante la presencia de ruido, especialmente cuando se utilizan métodos para evitar el sobreajuste.

La presencia de atributos redundantes no afecta adversamente la precisión de los árboles de decisión. Un atributo es redundante si está fuertemente correlacionado con otro atributo en los datos. Uno de los dos atributos redundantes no se utilizará para dividir una vez que se haya elegido el otro atributo. Sin embargo, si el conjunto de datos contiene muchos atributos irrelevantes, es decir, atributos que no son útiles para la tarea de clasificación, algunos de los atributos irrelevantes pueden elegirse accidentalmente durante el proceso de crecimiento de árboles, lo que da como resultado un árbol de decisiones que es más grande que necesario. Las técnicas de selección de características pueden ayudar a mejorar la precisión de los árboles de decisión al eliminar los atributos irrelevantes durante el preprocesamiento.

Como la mayoría de los algoritmos del árbol de decisión emplean un enfoque de partición recursiva descendente, el número de registros se reduce a medida que avanzamos por el árbol. En los nodos hoja, el número de registros puede ser demasiado pequeño para tomar una decisión estadísticamente significativa acerca de la representación de clase de los nodos. Esto se conoce como el problema de fragmentación de datos. Una posible solución es no permitir una mayor división cuando el número de registros cae por debajo de un cierto umbral.

Un subárbol se puede replicar varias veces en un árbol de decisión. Esto hace que el árbol de decisiones sea más complejo de lo necesario y quizás más difícil de interpretar. Tal situación puede surgir de las implementaciones del árbol de decisión que dependen de una condición

de prueba de atributo único en cada nodo interno. Dado que la mayoría de los algoritmos del árbol de decisión utilizan una estrategia de partición de dividir y conquistar, la misma condición de prueba se puede aplicar a diferentes partes del espacio de atributos, por lo que se dirige al problema de replicación de subárbol.



Tree replication problem. The same subtree can appear at different branches.

Las condiciones de prueba descritas hasta ahora en este capítulo involucran el uso de un solo atributo a la vez. Como consecuencia, el procedimiento de crecimiento de árboles puede verse como el proceso de partición del espacio de atributos en regiones desunidas hasta que cada región contenga registros de la misma clase.

El límite entre dos regiones vecinas de diferentes clases se conoce como límite de decisión. Dado que la condición de prueba involucra solo un único atributo, los límites de decisión son rectilíneos; es decir, paralelo a los "ejes de coordenadas". Esto limita la expresividad de la representación del árbol de decisión para modelar relaciones complejas entre atributos continuos.

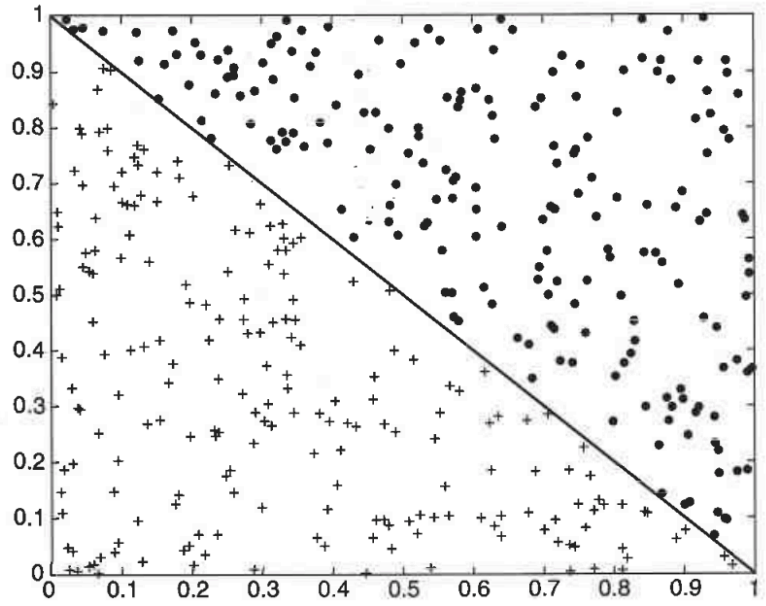


Figure 4.21. Example of data set that cannot be partitioned optimally using test conditions involving single attributes.

Sobreajuste del modelo

Los errores cometidos por un modelo de clasificación generalmente se dividen en dos tipos: errores de entrenamiento y errores de generalización. El error de entrenamiento, también conocido como error de sustitución o error aparente, es el número de errores de clasificación errónea cometidos en los registros de entrenamiento, mientras que el error de generalización es el error esperado del modelo en registros nunca vistos.

Recuerde que un buen modelo de clasificación no solo debe ajustarse bien a los datos de entrenamiento, sino que también debe clasificar con precisión los registros que nunca ha visto. En otras palabras, un buen modelo debe tener un error de entrenamiento bajo así como un error de generalización bajo. Esto es importante porque un modelo que se ajusta demasiado bien a los datos de entrenamiento puede tener un error de generalización más pobre que un modelo con un error de entrenamiento más alto. Tal situación se conoce como sobreajuste de modelo.

Si las tasas de error de entrenamiento y de prueba del modelo son grandes cuando el tamaño del árbol es muy pequeño, esta situación se conoce como subajuste del modelo. El faltante ocurre porque el modelo aún tiene que aprender la verdadera estructura de los datos. Como resultado, se desempeña mal tanto en el entrenamiento como en los conjuntos de pruebas. A medida que aumenta el número de nodos en el árbol de decisión, el árbol tendrá menos entrenamientos y pruebas. Sin embargo, una vez que el árbol se vuelve demasiado grande, su tasa de error de prueba comienza a aumentar a pesar de que su tasa de error de entrenamiento continúa.

Estimación de errores de generalización

Aunque la razón principal para el sobreajuste aún es un tema de debate, en general se acepta que la complejidad de un modelo tiene un impacto en el ajuste excesivo del modelo. La pregunta es, ¿cómo determinamos la complejidad del modelo correcto? La complejidad ideal es la de un modelo que produce el error de generalización más bajo. El problema es que el algoritmo de aprendizaje solo tiene acceso al conjunto de entrenamiento durante la construcción del modelo.

No tiene conocimiento del conjunto de pruebas y, por lo tanto, no sabe qué tan bien se desempeñará el árbol en los registros que nunca ha visto antes. Lo mejor que puede hacer es estimar el error de generalización del árbol inducido. Esta sección presenta varios métodos para hacer la estimación.

Usando Estimación de Sustitución

El enfoque de estimación de sustitución supone que el conjunto de capacitación es una buena representación de los datos generales. En consecuencia, el error de entrenamiento, también conocido como error de sustitución, se puede usar para proporcionar una estimación optimista del error de generalización. Bajo este supuesto, un algoritmo de inducción de árbol de decisión simplemente selecciona el modelo que produce la tasa de error de entrenamiento más baja como su modelo final. Sin embargo, el error de entrenamiento suele ser una mala estimación del error de generalización.

Evaluando el rendimiento de un clasificador

A menudo es útil medir el rendimiento del modelo en el conjunto de prueba porque dicha medida proporciona una estimación imparcial de su error de generalización. La precisión o la tasa de error calculada a partir del conjunto de pruebas también se puede utilizar para comparar el rendimiento relativo de diferentes clasificadores en el mismo dominio. Sin embargo, para hacer esto, se debe conocer la etiqueta de clase de los registros de prueba. Esta sección revisa algunos de los métodos comúnmente utilizados para evaluar el desempeño de un clasificador.

Aprendizaje supervisado

En aprendizaje automático y minería de datos, el aprendizaje supervisado es una técnica para deducir una función a partir de datos de entrenamiento. Los datos de entrenamiento consisten de pares de objetos (normalmente vectores): una componente del par son los datos de entrada y el otro, los resultados deseados. La salida de la función puede ser un valor numérico (como en los problemas de regresión) o una etiqueta de clase (como en los de clasificación). El objetivo del aprendizaje supervisado es el de crear una función capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos, los datos de entrenamiento. Para ello, tiene que generalizar a partir de los datos presentados a las situaciones no vistas previamente.

Aprendizaje no supervisado

Aprendizaje no supervisado es un método de Aprendizaje Automático donde un modelo es ajustado a las observaciones. Se distingue del Aprendizaje supervisado por el hecho de que no hay un conocimiento a priori. En el aprendizaje no supervisado, un conjunto de datos de objetos de entrada es tratado. Así, el aprendizaje no supervisado típicamente trata los objetos de entrada como un conjunto de variables aleatorias, siendo construido un modelo de densidad para el conjunto de datos.

El aprendizaje no supervisado puede ser usado en conjunto con la Inferencia bayesiana para producir probabilidades condicionales (es decir, aprendizaje supervisado) para cualquiera de las variables aleatorias dadas. El Santo Grial del aprendizaje no supervisado es la creación de un código factorial de los datos, esto es, un código con componentes estadísticamente independientes. El aprendizaje supervisado normalmente funciona mucho mejor cuando los datos iniciales son primero traducidos en un código factorial.

El aprendizaje no supervisado también es útil para la compresión de datos: fundamentalmente, todos los algoritmos de compresión dependen tanto explícita como implícitamente de una distribución de probabilidad sobre un conjunto de entrada.

Otra forma de aprendizaje no supervisado es la agrupación (en inglés, *clustering*), el cual a veces no es probabilístico.