Evaluación y selección de modelos (8.5)

[Data Mining Concepts and Techniques (Han-3ed-2012)]

Data Scientist
Jhoan Esteban Ruiz Borja Msc

Ahora que usted puede haber construido un modelo de clasificación, puede haber muchas preguntas pasando por su mente. Por ejemplo, suponga que utilizó datos de ventas anteriores para crear un clasificador para predecir el comportamiento de compra del cliente. Desea obtener una estimación de la **precisión** con que el clasificador puede predecir el comportamiento de compra de futuros clientes, es decir, los datos futuros del cliente en los que el clasificador no ha sido entrenado. Incluso puede haber intentado diferentes métodos para construir más de un clasificador y ahora desea comparar su **precisión**. Pero, ¿**qué es exactitud** o **precisión**? ¿**Cómo podemos estimarlo**? ¿**Son algunas medidas de la exactitud o precisión para medir un clasificador más apropiado que otras**? ¿**Cómo podemos obtener un cálculo de exactitud o precisión fiable**? Estas preguntas se tratan en esta sección.

La Sección 8.5.1 describe varias métricas de evaluación para la **precisión** predictiva de un clasificador. El método de **retención** o **Holdout** y **submuestreo aleatorio** (Sección 8.5.2), la **validación cruzada** (Sección 8.5.3) y los métodos de **bootstrap** (Sección 8.5.4) son **técnicas** comunes para evaluar la **precisión**, basadas en particiones aleatoriamente muestreadas de los datos dados. ¿**Qué pasa si tenemos más de un clasificador y queremos elegir el "mejor"**? Esto se denomina **selección de modelo** (es decir, elegir un clasificador sobre otro). Las dos últimas secciones tratan esta cuestión. La sección 8.5.5 discute cómo usar pruebas de significancia estadística para evaluar si la diferencia de precisión entre dos clasificadores se debe al azar. En la sección 8.5.6 se presenta cómo comparar los clasificadores basándose en las curvas características **costo-beneficio** y características operativas del receptor **(ROC)**.

Métricas para evaluar el rendimiento del clasificador [8.5.1]

Esta sección presenta medidas para evaluar cuán bueno o cuán "**exacto**" es su clasificador en la predicción de la etiqueta de una clase a la que pertenece unas tuplas. Consideraremos el caso en el que las clases de tuplas están distribuidas más o menos uniformemente, así como el caso en que las clases son desequilibradas (por ejemplo, donde una clase importante de interés es rara, como en pruebas médicas). Las medidas de evaluación del clasificador presentadas en esta sección se resumen en la **Figura 8.13**. Incluyen **precisión** (también conocida como tasa de reconocimiento, que puede ser definida como el porcentaje de tuplas clasificadas correctamente), **sensibilidad** (o recall), **especificidad**, **precisión**, F_1 y F_β. Tenga en cuenta que, aunque la precisión es una medida específica, la palabra "exactitud o precisión" también se utiliza como un término general para referirse a las capacidades predictivas de un clasificador.

El uso de datos de **entrenamiento** para derivar un clasificador y luego estimar la precisión del modelo aprendido resultante puede resultar en estimaciones demasiado optimistas engañosas debido a exceso de especialización del algoritmo de aprendizaje a los datos. En su lugar, es mejor medir la precisión del clasificador en un conjunto de **pruebas** que consiste en tuplas marcadas en la clase que no fueron usadas para entrenar el modelo.

Antes de discutir las diversas medidas, necesitamos sentirnos cómodos con alguna terminología. Recordemos que podemos hablar en términos de **tuplas positivas** (tuplas de la clase principal de interés) y **tuplas negativas** (todas las demás tuplas).

- Se denomina sensibilidad (sensitivity o recall) al cociente entre los verdaderos positivos y el total de positivos.
- También se denomina recognition.
- Se denomina especificidad (specificity) al cociente entre los verdaderos negativos y el total de negativos.

Medida	**Formula**
Precisión, tasa de reconocimiento	$\frac{TP + TN}{P + N}$
Tasa de error, tasa de malas clasificaciones	$\frac{FP + FN}{P + N}$
Sensibilidad, tasa de verdaderos positivos, recall	$\frac{TP}{P}$
Especificidad, tasa de falsos negativos	$\frac{TN}{N}$
Precisión	$\frac{TP}{TP + FP}$
$F, F_1, F - score$ Media armónica de precisión y recall	$\frac{2 \times Precisión \times recall}{Precisión + recall}$
F_β, donde β es un número real no negativo	

	$\frac{(1+\beta^2) \times Precisión \times recall}{\beta^2 \times Precisión + recall}$

Medidas de evaluación

Figura 8.13: Medidas de evaluación. Tenga en cuenta que algunas medidas son conocidas por más de un nombre. TP, TN, FP, P, N se refieren a la cantidad de muestras **verdadero positivas**, **verdaderas negativas**, **falsos positivas**, **positivas** y **negativas**, respectivamente (véase el texto).

Supongamos que usamos nuestro clasificador en un conjunto de tuplas de prueba etiquetadas. $\boldsymbol{P}$ es el número de tuplas positivas y $\boldsymbol{N}$ es el número de tuplas negativas. Para cada tupla, comparamos la predicción de la etiqueta de clase del clasificador con la etiqueta de clase conocida de la tupla.

Hay cuatro términos adicionales que necesitamos saber que son los "elementos básicos" utilizados en el cálculo de muchas medidas de evaluación. Entenderlas facilitará la comprensión del significado de las diversas medidas.

- **Verdadero positivos (TP):** Estos se refieren a las tuplas positivas que fueron correctamente etiquetadas o clasificadas por el clasificador. Sea **TP** el número de verdaderos positivos.
- **Verdaderos negativos (TN):** Estas son las tuplas negativas que fueron correctamente etiquetadas o clasificadas por el clasificador. Sea **TN** el número de negativos verdaderos.
- **Falso positivos (FP):** Estas son las tuplas negativas que fueron etiquetadas o clasificadas incorrectamente como positivas. Sea **FP** el número de falsos positivos.
- **Falso negativos (FN):** Estas son las tuplas positivas que fueron mal etiquetadas o clasificadas como negativas. Sea **FN** el número de falsos negativos.

Estos términos se resumen en la matriz de confusión de la figura 8.14.

	Predicción de la clase nueva			**Total**
Clase actual		**Si**	**No**	
	Si	TP	FN	P
	No	FP	TN	N
Total		P’	N’	P+N

Matriz de confusión

La matriz de confusión es una herramienta útil para analizar qué tan bien su clasificador puede reconocer las tuplas de diferentes clases. **TP** y **TN** nos dicen cuando el clasificador está haciendo las cosas bien, mientras **FP** y **FN** nos dicen cuando el clasificador está haciendo las cosas mal (es decir, el etiquetado erróneo). Dado $\boldsymbol{m}$ clases (donde $\boldsymbol{m \geq 2}$), una matriz de confusión es una tabla de tamaño $\boldsymbol{m}$ por $\boldsymbol{m}$. Una entrada, $\boldsymbol{CM_{i,j}}$ en las primeras $\boldsymbol{m}$ filas y $\boldsymbol{m}$ columnas indicando el número de tuplas de la clase $\boldsymbol{i}$ que fueron etiquetadas o clasificadas por el clasificador como de la clase $\boldsymbol{j}$. Para que un clasificador tenga una buena precisión, idealmente la mayoría de las tuplas estarían representadas a lo largo de la diagonal de la

matriz de confusión, desde la entrada $\boldsymbol{CM_{1,1}}$ hasta la entrada $\boldsymbol{CM_{m,m}}$, siendo el resto de las entradas cero o cercano a cero. Es decir, idealmente, **FP** y **FN** son alrededor de cero.

La tabla puede tener filas o columnas adicionales para proporcionar totales. Por ejemplo, en la matriz de confusión de la figura 8.14, se muestran **P** y **N**. Además, **P'** es el número de tuplas que fueron etiquetadas o clasificadas como positivas (**TP + FP**) y **N'** es el número de tuplas que fueron etiquetadas o clasificadas como negativas (**TN + FN**). El número total de tuplas es **TP + TN + FP + TN**, o **P + N**, o **P' + N'**. Tenga en cuenta que, aunque la matriz de confusión mostrada es para un problema de clasificación binaria, las matrices de confusión se pueden dibujar fácilmente para varias clases en una manera similar.

Ahora veamos las medidas de evaluación, empezando por la precisión. La precisión de un clasificador en un conjunto de pruebas dado es el porcentaje de tuplas de conjuntos de prueba que son clasificadas correctamente por el clasificador. Es decir,

$$\boldsymbol{Precisión} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{P + N}} \qquad (\mathbf{8.21})$$

En la literatura de reconocimiento de patrones, esto también se conoce como la **tasa de reconocimiento** global del clasificador, es decir, refleja qué tan bien el clasificador reconoce las tuplas de las diversas clases.

También podemos hablar de la **tasa de error** o **clasificación errónea** de un clasificador $\boldsymbol{M}$, que es simplemente $\mathbf{1} - \boldsymbol{Precisión}(\boldsymbol{M})$, donde la $\boldsymbol{Precisión}(\boldsymbol{M})$ es la precisión de $\boldsymbol{M}$. Esto también se puede calcular como

$$\boldsymbol{tasa\ de\ error} = \frac{\boldsymbol{FP + FN}}{\boldsymbol{P + N}} \qquad (\mathbf{8.22})$$

Si se usara el conjunto de entrenamiento (en lugar de un conjunto de pruebas) para estimar la tasa de error de un modelo, esta cantidad se conoce como el **error de resubstitución**. Esta estimación de error es optimista de la tasa de error real (y de manera similar, la estimación de precisión correspondiente es optimista) porque el modelo no se prueba en ninguna muestra que aún no haya visto.

Ahora consideramos el problema de desequilibrio de clase, donde la clase principal de interés es rara. Es decir, la distribución del conjunto de datos refleja una mayoría significativa de la clase negativa y una clase minoritaria positiva. Por ejemplo, en las aplicaciones de detección de fraude, la clase de interés (o clase positiva) es "fraude", que ocurre con mucha menos frecuencia que la clase negativa "no fraude". En los datos médicos, puede haber una clase rara, como "cáncer". Suponga que ha entrenado un clasificador para clasificar las tuplas de datos médicos, donde el atributo de etiqueta de clase es "cáncer" y los posibles valores de clase son "sí" y "no". Una tasa de exactitud de, por ejemplo, el 97% Bastante exacta, pero lo que si sólo, por ejemplo, el 3% de las tuplas de formación son en realidad el cáncer? Claramente, una tasa de exactitud del 97% puede no ser aceptable, el clasificador podría etiquetar correctamente solamente las tuplas no cancerosas, por ejemplo, y clasificar erróneamente todas las tuplas de cáncer. En lugar de eso, necesitamos otras medidas que

tengan acceso a qué tan bien el clasificador puede reconocer las tuplas positivas (cáncer = sí) y qué tan bien puede reconocer las tuplas negativas (cáncer = no).

Las medidas de **sensibilidad** y **especificidad** pueden utilizarse, respectivamente, para este propósito. La sensibilidad también se conoce como la tasa de verdadero positiva (reconocimiento) (es decir, la proporción de tuplas positivas que se identifican correctamente), mientras que la especificidad es la tasa de verdadero negativo (es decir, la proporción de tuplas negativas que se identifican correctamente). Estas medidas se definen como

$$\boldsymbol{Sensibilidad} = \frac{\boldsymbol{TP}}{\boldsymbol{P}} \qquad (\mathbf{8.23})$$

$$\boldsymbol{Especificidad} = \frac{\boldsymbol{TN}}{\boldsymbol{N}} \qquad (\mathbf{8.24})$$

Se puede demostrar que la precisión es una función de la sensibilidad y la especificidad:

$$\boldsymbol{Precisión} = \boldsymbol{Sensibilidad}\frac{\boldsymbol{P}}{(\boldsymbol{P}+\boldsymbol{N})} + \boldsymbol{Especificidad}\frac{\boldsymbol{N}}{(\boldsymbol{P}+\boldsymbol{N})} \qquad (\mathbf{8.25})$$

Las medidas de precisión y recall también son ampliamente utilizadas en la clasificación. La precisión puede considerarse como una medida de la exactitud (es decir, qué porcentaje de tuplas etiquetadas como positivas son realmente tales), mientras que **recall** es una medida de completitud (qué porcentaje de tuplas positivas se etiquetan como tales). Si recall parece familiar, es porque es lo mismo que la sensibilidad (o la verdadera tasa positiva). Estas medidas se pueden calcular como

$$\boldsymbol{Precisión} = \frac{\boldsymbol{TP}}{\boldsymbol{TP}+\boldsymbol{FP}} \qquad (\mathbf{8.26})$$

$$\boldsymbol{recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP}+\boldsymbol{FN}} = \frac{\boldsymbol{TP}}{\boldsymbol{P}} \qquad (\mathbf{8.27})$$

Una perfecta puntuación de la precisión es 1.0 para una clase C significa que cada tupla que el clasificador haya etiquetado como perteneciente a la clase C pertenece de hecho a la clase C. Sin embargo, no nos dice nada sobre el número de tuplas de clase C que clasificó erróneamente. Una perfecta puntuación de **recall** es 1.0 para C significa que cada elemento de la clase C fue etiquetado como tal, pero no nos dice cuántas otras tuplas fueron etiquetadas incorrectamente como pertenecientes a la clase C. Tiende a existir una relación inversa entre la precisión y **recall**, donde es posible aumentar uno a costa de reducir el otro. Por ejemplo, nuestro clasificador médico puede lograr una alta precisión al etiquetar todas las tuplas de cáncer que presentan cierta manera como cáncer, pero puede tener poca memoria si se hace mal en muchos otros casos de tuplas de cáncer. Las puntuaciones de precisión y recall se usan típicamente juntas, donde los valores de precisión se comparan para un valor fijo de **recall**, o viceversa. Por ejemplo, podemos comparar valores de precisión con un valor de **recall** de, por ejemplo, 0,75.

Una forma alternativa de usar precisión y recordar es combinarlas en una sola medida. Este es el enfoque de la medida $\boldsymbol{F}$ (también conocida como la puntuación $\boldsymbol{F_1}$ o F-score) y la medida $\boldsymbol{F_\beta}$. Se definen como

$$\boldsymbol{F = \frac{2 \times Precisión \times recall}{Precisión + recall}} \qquad (\mathbf{8.28})$$

$$\boldsymbol{F_\beta = \frac{\left(1 + \beta^2\right) \times Precisión \times recall}{\beta^2 \times Precisión + recall}} \qquad (\mathbf{8.29})$$

Donde $\boldsymbol{\beta}$ es un número real no negativo. La medida $\boldsymbol{F}$ es la media armónica de precisión y recall (cuya prueba se deja como un ejercicio). Da igual peso a la precisión y recall. La medida $\boldsymbol{F}$ es una medida ponderada de precisión y recall. Asigna $\boldsymbol{\beta}$ tiempos de tanto recall como de precisión. Las medidas $\boldsymbol{F_\beta}$ F de uso común son $\boldsymbol{F_2}$ (pesos que recuerdan el doble de precisión) y $\boldsymbol{F_{0.5}}$ (que pesa la precisión dos veces más que recordar).

"¿Hay otros casos donde la exactitud puede no ser apropiada?" En los problemas de clasificación, se supone comúnmente que todas las tuplas son clasificables de forma única, es decir, que cada tupla de entrenamiento puede pertenecer a una sola clase. Sin embargo, debido a la gran diversidad de datos en grandes bases de datos, no siempre es razonable suponer que todas las tuplas son exclusivamente clasificables. Más bien, es más probable asumir que cada tupla puede pertenecer a más de una clase. ¿Cómo se puede medir la precisión de los clasificadores en las grandes bases de datos? La medida de exactitud no es apropiada, porque no tiene en cuenta la posibilidad de tuplas pertenecientes a más de una clase.

En lugar de devolver una etiqueta de clase, es útil devolver una distribución de clases de probabilidad. Las medidas de precisión pueden entonces utilizar una segunda heurística de conjetura, por lo que una predicción de clase se juzga como correcta si coincide con la primera o segunda clase más probable. Aunque esto tiene en cuenta, hasta cierto punto, la clasificación no singular de las tuplas, no es una solución completa.

Además de las medidas basadas en la exactitud, los clasificadores también pueden compararse con respecto a los siguientes aspectos adicionales:

- **Velocidad:** Se refiere a los costos computacionales involucrados en la generación y uso del clasificador dado.
- **Robustez:** Esta es la capacidad del clasificador de hacer predicciones correctas dados datos ruidosos o datos con valores faltantes. La robustez se evalúa típicamente con una serie de conjuntos de datos sintéticos que representan grados cada vez mayores de ruido y valores faltantes.
- **Escalabilidad:** Se refiere a la capacidad de construir el clasificador eficientemente dado grandes cantidades de datos. La escalabilidad se evalúa típicamente con una serie de conjuntos de datos de tamaño creciente.

- **Interpretabilidad:** Se refiere al nivel de comprensión y perspicacia que proporciona el clasificador o predictor. La interpretabilidad es subjetiva y por lo tanto más difícil de evaluar. Los árboles de decisión y las reglas de clasificación pueden ser fáciles de interpretar, pero su interpretabilidad puede disminuir cuanto más complejas lleguen a ser complejas. Discutimos algunos trabajos en esta área, como la extracción de reglas de clasificación de un clasificador de red neural de "caja negra" llamado backpropagation. Capítulo 9.

En resumen, hemos presentado varias medidas de evaluación. La medida de exactitud funciona mejor cuando las clases de datos están bastante distribuidas. Otras medidas, como sensibilidad (o recuerdo), especificidad, precisión, $\boldsymbol{F}$ y $\boldsymbol{F_\beta}$, se adaptan mejor al problema de desequilibrio de clase, donde la clase principal de interés es rara. Las subsecciones restantes se centran en la obtención de estimaciones confiables de exactitud de clasificador.

Método de Holdout y submuestreo aleatorio [8.5.2]

Método Holdout o de retención: El método de Holdout es lo que hemos aludido hasta ahora en nuestras discusiones sobre la precisión. En este método, los datos se reparten al azar en dos grupos independientes, un conjunto de **entrenamiento (training)** y un conjunto de **prueba (test)**. Por lo general, dos terceras partes de los datos se asignan al conjunto de entrenamiento (training), y el tercio restante se destina al conjunto de prueba (test). El conjunto de **entrenamiento** se utiliza para **derivar el modelo**. La precisión del modelo se calcula entonces con el conjunto de prueba (Figura 8.17). La estimación es pesimista, porque solamente una porción de los datos iniciales se utiliza para derivar el modelo.

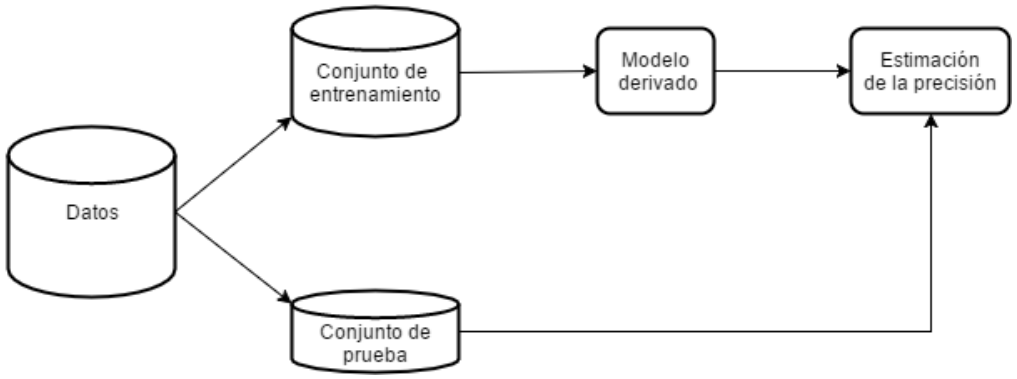

Figura 8.17: Estimación de la precisión con el método Holdout.

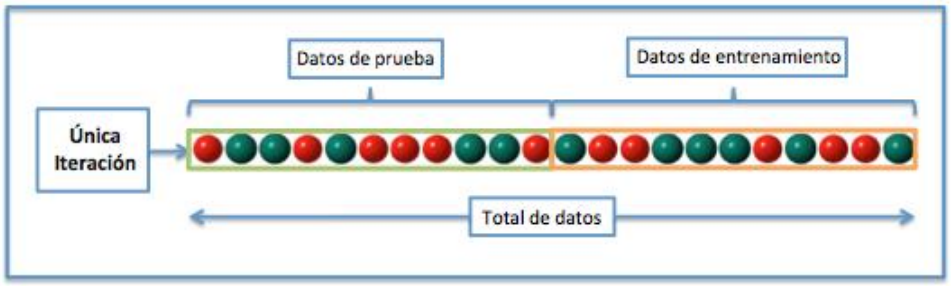

Figura: Estimación con el método Holdout.

Sub-muestreo aleatorio: Sub-muestreo aleatorio es una variación del método de Holdout en la cual el método de Holdout es repetido $\boldsymbol{k}$ veces. La estimación de la precisión general se toma como el promedio de las precisiones de cada iteración.

Validación Cruzada (8.5.3)

Objetivo de la validación cruzada

Suponemos que tenemos un modelo con uno o más parámetros de ajuste desconocido y unos datos de entrenamiento que queremos analizar. El proceso de ajuste optimiza los parámetros del modelo para que éste se ajuste a los datos de entrenamiento tan bien como prueba. Si tomamos una muestra independiente como datos de prueba (validación), del mismo grupo que los datos de entrenamiento, normalmente el modelo no se ajustará a los datos de prueba igual de bien que a los datos de entrenamiento. Esto se denomina sobreajuste y acostumbra a pasar cuando el tamaño de los datos de entrenamiento es pequeño o cuando el número de parámetros del modelo es grande. La validación cruzada es una manera de predecir el ajuste de un modelo a un hipotético conjunto de datos de prueba cuando no disponemos del conjunto explícito de datos de prueba.

Validación cruzada k-iteraciones o k-fold

En la **validación cruzada de $\boldsymbol{k}$-iteraciones o $\boldsymbol{k}$-fold**, los datos iniciales se dividen aleatoriamente en $\boldsymbol{k}$ subconjuntos mutuamente excluyentes o "pliegues," $\boldsymbol{D_1, D_2, \dots, D_k}$, cada uno de aproximadamente el mismo tamaño. El entrenamiento y la prueba se realizan $\boldsymbol{k}$ veces. En la iteración $\boldsymbol{i}$, la partición $\boldsymbol{D_i}$ se reserva como el conjunto de prueba, y las particiones restantes se usan colectivamente para entrenar el modelo. Es decir, en la primera iteración, los subconjuntos $\boldsymbol{D_2, \dots, D_k}$ sirven colectivamente como el conjunto de entrenamiento para obtener un primer modelo, que se prueba en $\boldsymbol{D_1}$; La segunda iteración se entrena en subconjuntos $\boldsymbol{D_1, D_3, \dots, D_k}$ y se prueba en $\boldsymbol{D_2}$; y así. A diferencia de los métodos holdout y submuestreo aleatorio, aquí cada muestra se utiliza el mismo número de veces para el entrenamiento y una vez para la prueba. Para la clasificación, la estimación de exactitud o precisión es el número total de clasificaciones correctas de las $\boldsymbol{k}$ iteraciones, dividido por el número total de tuplas en los datos iniciales.

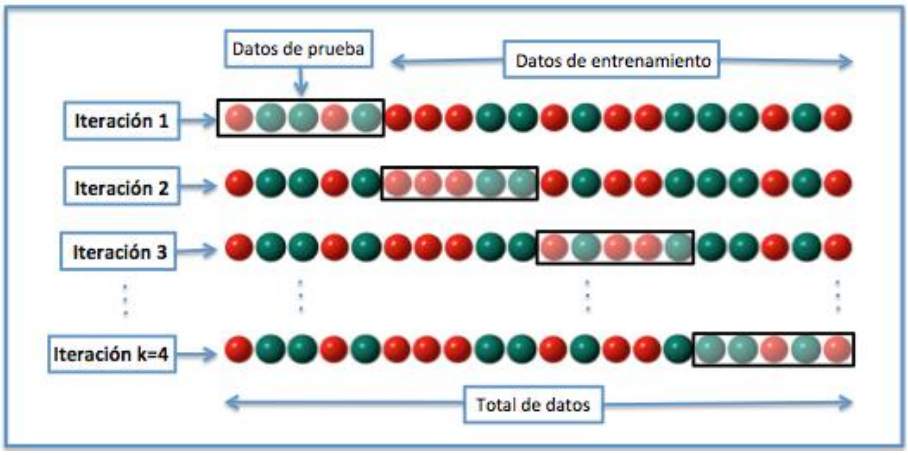

Figura: Método k-iteraciones o k-fold.

Validación cruzada aleatoria

Este método consiste en dividir aleatoriamente el conjunto de datos de entrenamiento y el conjunto de datos de prueba. Para cada división la función de aproximación se ajusta a partir de los datos de entrenamiento y calcula los valores de salida para el conjunto de datos de prueba. El resultado final corresponde a la media aritmética de los valores obtenidos para las diferentes divisiones. La ventaja de este método es que la división de datos de entrenamiento-prueba no depende del número de iteraciones. Pero, en cambio, con este método hay algunas muestras que quedan sin evaluar y otras que se evalúan más de una vez, es decir, los subconjuntos de prueba y entrenamiento se pueden solapar.

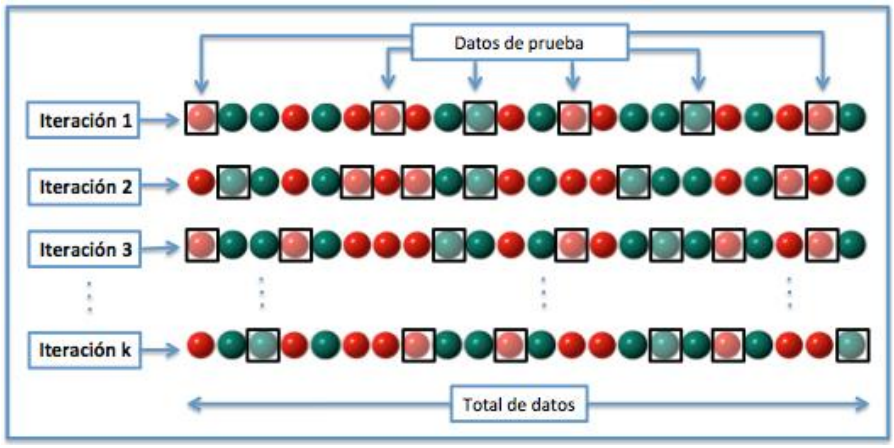

Figura: Método de validación cruzada aleatoriamente.

Validación cruzada dejando uno fuera (Leave one out)

La validación cruzada dejando una fuera o Leave-une-out-validation implica separar los datos de forma que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento. La evaluación viene dada por el error, y en este tipo de validación cruzada el error es muy bajo, pero en cambio, a nivel computacional es muy costoso, puesto que se tienen que realizar un elevado número de iteraciones, tantas como $\boldsymbol{N}$ muestras tengamos y para cada una analizar los datos tanto de entrenamiento como de prueba.

Leave-one-out es un caso especial de validación cruzada de $\boldsymbol{k}$ veces, donde $\boldsymbol{k}$ se ajusta al número de tuplas iniciales. Es decir, sólo una muestra es "dejada fuera" a la vez para el conjunto de prueba. En la **validación cruzada estratificada**, los pliegues son estratificados de manera que la distribución de clase de las tuplas en cada pliegue es aproximadamente la misma que en los datos iniciales.

En general, se recomienda la validación cruzada estratificada de 10 veces para estimar la exactitud (incluso si la potencia de cálculo permite usar más pliegues) debido a su sesgo y varianza relativamente bajos.

Bootstrap [8.5.4]

A diferencia de los métodos de estimación de precisión que acabamos de mencionar, el método bootstrap muestrea las tuplas de entrenamiento dadas uniformemente con reemplazo. Es decir, cada vez que se selecciona una tupla, es igualmente probable que se vuelva a seleccionar y se vuelva a agregar al conjunto de entrenamiento. Por ejemplo, imagine una máquina que seleccione aleatoriamente tuplas para nuestro conjunto de entrenamiento. En el muestreo con reemplazo, la máquina puede seleccionar la misma tupla más de una vez.

Hay varios métodos de bootstrap. Un comúnmente utilizado es el **.632 bootstrap**, que funciona de la siguiente manera. Supongamos que se nos da un conjunto de datos de $\boldsymbol{d}$ tuplas. El conjunto de datos se muestrea $\boldsymbol{d}$ veces, con reemplazo, resultando en una muestra de bootstrap o conjunto de entrenamiento de $\boldsymbol{d}$ muestras. Es muy probable que algunas de las tuplas de datos originales ocurrirán más de una vez en esta muestra. Las tuplas de datos que no entraron en el conjunto de entrenamiento terminan formando el conjunto de pruebas. Supongamos que deberíamos probar esto varias veces. Como resultado, en promedio, el 63,2% de las tuplas de datos originales terminará en la muestra de bootstrap, y el 36,8% restante formará el conjunto de pruebas (de ahí el nombre, .632 bootstrap).

"¿De dónde proviene la figura, 63.2%?" Cada tupla tiene una probabilidad de $\boldsymbol{1/d}$ de ser seleccionado, por lo que la probabilidad de no ser elegido es $\boldsymbol{(1 - 1/d)}$. Tenemos que seleccionar $\boldsymbol{d}$ veces, por lo que la probabilidad de que una tupla no se elija durante todo este tiempo es $\boldsymbol{(1 - 1/d)^d}$. Si $\boldsymbol{d}$ es grande, la probabilidad se aproxima a $\boldsymbol{e^{-1} = 0.368}$. Por lo tanto, 36.8% de tuplas no serán seleccionadas para entrenamiento y por lo tanto terminarán en el conjunto de prueba, y el 63.2% restante formará el conjunto de entrenamiento.

Podemos repetir el procedimiento de muestreo $\boldsymbol{k}$ veces, donde en cada iteración, usamos el conjunto de pruebas actuales para obtener una estimación de exactitud del modelo obtenido

a partir de la muestra de bootstrap actual. La precisión global del modelo $\boldsymbol{M}$, se estima entonces como

$$\boldsymbol{Acc(M) = \frac{1}{k}\sum_{i=1}^{k}\left(0.632 * Acc(M_i)_{test_{set}} + 0.368 * Acc(M_i)_{train_{set}}\right)}$$

Donde $\boldsymbol{Acc(M_i)_{test_{set}}}$ es la precisión del modelo obtenido con la muestra bootstrap $\boldsymbol{i}$ cuando es aplicado al conjunto de prueba $\boldsymbol{i}$. $\boldsymbol{Acc(M_i)_{train_{set}}}$ es la precisión del modelo obtenido con la muestra bootstrap $\boldsymbol{i}$ cuando se aplica al conjunto original de tuplas de datos. Bootstrapping tiende a ser demasiado optimista. Funciona mejor con pequeños conjuntos de datos.

Selección de modelos mediante pruebas estadísticas de significación [8.5.5]

Supongamos que hemos generado dos modelos de clasificación, $\boldsymbol{M_1}$ y $\boldsymbol{M_2}$, a partir de nuestros datos. Hemos realizado una CV de 10-fold para obtener una tasa de error media para cada uno. ¿Cómo podemos determinar cuál es el mejor modelo? Puede parecer intuitivo seleccionar el modelo con la tasa de error más baja; sin embargo, las tasas de error promedio son solo estimaciones de error en la población real de casos de datos futuros. Puede haber una variación considerable entre las tasas de error dentro de cualquier experimento de validación cruzada de 10-fold. Aunque las tasas de error promedio obtenidas para, $\boldsymbol{M_1}$ y $\boldsymbol{M_2}$ pueden parecer diferentes, esa diferencia puede no ser estadísticamente significativa. ¿Qué pasa si alguna diferencia entre los dos puede ser atribuida al azar? Esta sección aborda estas preguntas.

Para determinar si hay alguna diferencia "real" en las tasas de error promedio de dos modelos, debemos emplear una prueba de significación estadística. Además, queremos obtener algunos límites de confianza para nuestras tasas de error promedio para que podamos hacer afirmaciones como, "cualquier media observada no variará $\pm$ en dos errores estándar el 95% del tiempo para las muestras futuras" o "un modelo es Mejor que el otro por un margen de error $\pm$ del 4%. "

¿Qué necesitamos para realizar la prueba estadística? Supongamos que, para cada modelo, realizamos una validación cruzada 10-fold, es decir, 10 veces, cada vez utilizando una partición de datos diferente de 10-fold. Cada partición se dibuja de forma independiente. Podemos promediar las 10 tasas de error obtenidas cada una para $\boldsymbol{M_1}$ y $\boldsymbol{M_2}$, respectivamente, para obtener la tasa de error media para cada modelo. Para un modelo dado, las tasas de error individuales calculadas en las validaciones cruzadas pueden considerarse como muestras diferentes e independientes de una distribución de probabilidad. En general, siguen una distribución $\boldsymbol{t}$ con $\boldsymbol{k-1}$ grados de libertad donde, aquí, $\boldsymbol{k=10}$. (Esta distribución parece muy similar a una distribución normal, o gaussiana, aunque las funciones que definen a las dos son bastante diferentes. Son unimodales, simétricas y con forma de campana.) Esto nos permite hacer pruebas de hipótesis en las que la prueba de significación utilizada es la prueba $\boldsymbol{t}$, o la prueba $\boldsymbol{t}$ de Student. Nuestra hipótesis es que los dos modelos son iguales, o, en otras

palabras, que la diferencia en la tasa de error media entre los dos es cero. Si podemos rechazar esta hipótesis (conocida como hipótesis nula), podemos concluir que la diferencia entre los dos modelos es estadísticamente significativa, en cuyo caso podemos seleccionar el modelo con la tasa de error más baja.

En la práctica de minería de datos, a menudo podemos emplear un solo conjunto de pruebas, es decir, el mismo conjunto de pruebas se puede usar tanto para $\boldsymbol{M_1}$ y $\boldsymbol{M_2}$. En tales casos, hacemos una comparación por pares de los dos modelos para cada ronda de validación cruzada de 10-fold. Es decir, para la ronda $\boldsymbol{i}$-esima de 10-fold la validación cruzada, se utiliza la misma partición de validación cruzada para obtener una tasa de error para $\boldsymbol{M_1}$ y para $\boldsymbol{M_2}$. Sea $\boldsymbol{err(M_1)_i}$ (o $\boldsymbol{err(M_2)_i}$) la tasa de error del modelo $\boldsymbol{M_1}$ (o $\boldsymbol{M_2}$) en la ronda $\boldsymbol{i}$. Las tasas de error para $\boldsymbol{M_1}$ se promedian para obtener una tasa de error media para $\boldsymbol{M_1}$, denotado $\boldsymbol{\overline{err}(M_1)}$. Del mismo modo, podemos obtener $\boldsymbol{\overline{err}(M_2)}$. La varianza de la diferencia entre los dos modelos se denota $\boldsymbol{var(M_1 - M_2)}$. La prueba $\boldsymbol{t}$ calcula el estadístico $\boldsymbol{t}$ con $\boldsymbol{k-1}$ grados de libertad para $\boldsymbol{k}$ muestras. En nuestro ejemplo tenemos $\boldsymbol{k = 10}$ ya que, aquí, las $\boldsymbol{k}$ muestras son nuestras tasas de error obtenidas de diez validaciones cruzadas de 10-fold para cada modelo. El estadístico $\boldsymbol{t}$ para la comparación por pares se calcula de la siguiente manera:

$$\boldsymbol{t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{\frac{var(M_1 - M_2)}{k}}}} \qquad (\boldsymbol{8.31})$$

Donde

$$\boldsymbol{var(M_1 - M_2) = \frac{1}{k}\sum_{i=1}^{k}\left[err(M_1)_i - err(M_2)_i - \left(\overline{err}(M_1) - \overline{err}(M_2)\right)\right]^2} \quad (\boldsymbol{8.32})$$

Para determinar si $\boldsymbol{M_1}$ y $\boldsymbol{M_2}$ son significativamente diferentes, calculamos $\boldsymbol{t}$ y seleccionamos un nivel de significación, $\boldsymbol{sig}$. En la práctica, normalmente se usa un nivel de significación del 5% o 1%. Luego consultamos una tabla para la distribución $\boldsymbol{t}$, disponible en libros de texto estándar sobre estadísticas. Esta tabla generalmente se muestra organizada por grados de libertad como filas y niveles de significación como columnas. Supongamos que queremos determinar si la diferencia entre $\boldsymbol{M_1}$ y $\boldsymbol{M_2}$ es significativamente diferente para el 95% de la población, es decir, $\boldsymbol{sig} = 5\%$ o 0.05. Necesitamos encontrar el valor de distribución $\boldsymbol{t}$ correspondiente a $\boldsymbol{k-1}$ grados de libertad (o 9 grados de libertad para nuestro ejemplo) de la tabla. Sin embargo, debido a que la distribución $\boldsymbol{t}$ es simétrica, generalmente solo se muestran los puntos porcentuales superiores de la distribución . Por lo tanto, buscamos el valor de la tabla para $\boldsymbol{z = sig/2}$, que en este caso es 0.025, donde z también se conoce como límite de confianza. Si $\boldsymbol{t > z}$ o $\boldsymbol{t < -z}$, entonces nuestro valor de $\boldsymbol{t}$ se encuentra en la región de rechazo, dentro de las colas de la distribución. Esto significa que podemos rechazar la hipótesis nula de que las medias de $\boldsymbol{M_1}$ y $\boldsymbol{M_2}$ son las mismas y concluir que existe una diferencia estadísticamente significativa entre los dos modelos. De lo contrario, si no podemos rechazar la hipótesis nula, llegamos a la conclusión de que cualquier diferencia entre $\boldsymbol{M_1}$ y $\boldsymbol{M_2}$ puede atribuirse al azar.

Si hay dos conjuntos de prueba disponibles en lugar de un único conjunto de prueba, entonces se usa una versión no pareada de la prueba $\boldsymbol{t}$, donde la variación entre las medias de los dos modelos se estima como

$$var(M_1 - M_2) = \sqrt{\frac{var(M_1)}{k_1} + \frac{var(M_2)}{k_2}} \qquad (8.33)$$

y k_1 y k_2 son el número de muestras de validación cruzada (en nuestro caso, rondas de validación cruzada de 10-fold) utilizadas para M_1 y M_2, respectivamente. Esto también se conoce como la prueba $\boldsymbol{t}$ de dos muestras. Al consultar la tabla de distribución $\boldsymbol{t}$, el número de grados de libertad utilizado se toma como el número mínimo de grados de los dos modelos.

Comparación de clasificadores basados en costo-beneficio y curvas ROC [8.5.6]

Los verdaderos positivos, los verdaderos negativos, los falsos positivos y los falsos negativos también son útiles para evaluar los costos y beneficios (o riesgos y ganancias) asociados con un modelo de clasificación. El costo asociado con un falso negativo (como predecir incorrectamente que un paciente canceroso no es canceroso) es mucho mayor que el de un falso positivo (etiquetar incorrectamente a un paciente no canceroso como canceroso). En tales casos, podemos superar un tipo de error sobre otro asignando un costo diferente a cada uno. Estos costos pueden considerar el peligro para el paciente, los costos financieros de las terapias resultantes y otros costos hospitalarios. De manera similar, los beneficios asociados con una verdadera decisión positiva pueden ser diferentes a los de un verdadero negativo. Hasta ahora, para calcular la precisión del clasificador, hemos asumido costos iguales y esencialmente dividimos la suma de los positivos verdaderos y los negativos verdaderos por el número total de tuplas de prueba.

Alternativamente, podemos incorporar costos y beneficios al calcular el costo promedio (o beneficio) por decisión. Otras aplicaciones que involucran el análisis de costo-beneficio incluyen decisiones de solicitud de préstamos y envíos de marketing de destino. Por ejemplo, el costo del préstamo a un incumplimiento de pago excede en gran medida el de la pérdida de negocios incurrida al denegar un préstamo a un impago. De manera similar, en una aplicación que trata de identificar hogares que probablemente respondan a los envíos de ciertos materiales promocionales, el costo de los envíos a numerosos hogares que no responden puede compensar el costo de la pérdida de negocios de no enviar correos a hogares que hubieran respondido. Otros costos a considerar en el análisis general incluyen los costos para recopilar los datos y desarrollar la herramienta de clasificación.

Las curvas de características operativas del receptor son una herramienta visual útil para comparar dos modelos de clasificación. Las curvas ROC provienen de la teoría de detección de señales que se desarrolló durante la Segunda Guerra Mundial para el análisis de imágenes de radar. Una curva ROC para un modelo dado muestra la compensación entre la tasa de positivos verdaderos (TPR) y la tasa de falsos positivos (FPR). Dado un conjunto de pruebas y un modelo, TPR es la proporción de positivos (o "sí") tuplas que están correctamente

etiquetadas por el modelo; FPR es la proporción de tuplas negativas (o "no") que están mal etiquetadas como positivas. Dado que TP, FP, P y N son el número de tuplas verdaderas positivas, falsas positivas, positivas y negativas, respectivamente, de la Sección 8.5.1 sabemos que $\boldsymbol{TPR = \frac{TP}{P}}$, que es la sensibilidad. Además, $\boldsymbol{TPR = \frac{FP}{N}}$, que es de $\boldsymbol{1 - especificidad}$.

Para un problema de dos clases, una curva ROC nos permite visualizar la compensación entre la velocidad a la que el modelo puede reconocer con precisión los casos positivos y la velocidad a la que identifica erróneamente los casos negativos como positivos para diferentes partes del conjunto de pruebas. Cualquier aumento en TPR ocurre a costa de un aumento en FPR. El área bajo la curva ROC es una medida de la precisión del modelo.

Para trazar una curva ROC para un modelo de clasificación dado, $\boldsymbol{M}$, el modelo debe poder devolver una probabilidad de la clase predicha para cada tupla de prueba. Con esta información, clasificamos y clasificamos las tuplas para que la tupla que es más probable que pertenezca a la clase positiva o "sí" aparezca en la parte superior de la lista, y la tupla que sea menos probable que pertenezca a la clase positiva. los clasificadores devuelven una distribución de probabilidad de clase para cada predicción y, por lo tanto, son apropiados, aunque otros clasificadores, como los clasificadores del árbol de decisión (Sección 10), los clasificadores Bayesian (Sección 8.3) y Backpropagation (Sección 9.2) devuelven una distribución de probabilidad de clase para cada predicción (Sección 8.2), se puede modificar fácilmente para devolver las predicciones de probabilidad de clase. Deje que el valor que devuelve un clasificador probabilístico para una tupla $\boldsymbol{X}$ dada sea $\boldsymbol{f(X) \rightarrow [0, 1]}$. Para un problema binario, normalmente se selecciona un umbral $\boldsymbol{t}$ para que las tuplas donde $\boldsymbol{f(X) \geq t}$ se consideren positivas y todas las otras tuplas se consideren negativas. Tenga en cuenta que el número de positivos verdaderos y el número de falsos positivos son funciones de $\boldsymbol{t}$, por lo que podríamos escribir $\boldsymbol{TP(t)}$ y $\boldsymbol{FP(t)}$. Ambas son funciones descendentes monótonas.

Primero describimos la idea general detrás de trazar una curva ROC, y luego seguimos con un ejemplo. El eje vertical de una curva ROC representa TPR. El eje horizontal representa FPR. Para trazar una curva ROC para $\boldsymbol{M}$, comenzamos de la siguiente manera. Comenzando en la esquina inferior izquierda (donde TPR = FPR = 0), verificamos la etiqueta de clase real de la tupla en la parte superior de la lista. Si tenemos un verdadero positivo (es decir, una tupla positiva que fue clasificada correctamente), entonces TP y, por tanto, aumentan TPR. En la gráfica, nos movemos hacia arriba y trazamos un punto. Si, en cambio, el modelo clasifica una tupla negativa como positiva, tenemos un falso positivo y, por lo tanto, tanto FP como FPR aumentan. En la gráfica, nos movemos a la derecha y trazamos un punto. Este proceso se repite para cada una de las tuplas de prueba en orden ordenado, cada vez que se mueve hacia arriba en el gráfico para un positivo verdadero o hacia la derecha para un falso positivo.

Ejemplo 8.11: Graficando curva ROC: La Figura 8.18 muestra el valor de probabilidad (columna 3) devuelto por un clasificador probabilístico para cada una de las 10 tuplas en un

conjunto de pruebas, ordenado por orden de probabilidad decreciente. La columna 1 es simplemente un número de identificación de tupla, lo que ayuda en nuestra explicación. La columna 2 es la etiqueta de clase real de la tupla. Hay cinco tuplas positivas y cinco tuplas negativas, por lo tanto, $\boldsymbol{P = 5}$ y $\boldsymbol{N = 5}$. Al examinar la etiqueta de clase conocida de cada tupla, podemos determinar los valores de las columnas restantes, TP, FP, TN, FN, TPR y FPR. Comenzamos con la tupla 1, que tiene la puntuación de probabilidad más alta, y tomamos esa puntuación como nuestro umbral, es decir, $\boldsymbol{t = 0.9}$. Por lo tanto, el clasificador considera que la tupla 1 es positiva, y todas las otras tuplas se consideran negativas. Como la etiqueta de clase real de la tupla 1 es positiva, tenemos un verdadero positivo, por lo tanto, $\boldsymbol{TP = 1}$ y $\boldsymbol{FP = 0}$. Entre las nueve tuplas restantes, todas clasificadas como negativas, cinco son en realidad negativas (por lo tanto, $\boldsymbol{TN = 5}$) . Los cuatro restantes son realmente positivos, por lo tanto, $\boldsymbol{FN = 4}$. Por lo tanto, podemos calcular TPR $= \frac{TP}{P} = \frac{1}{5} = 0.2$, mientras que $\boldsymbol{FPR = 0}$. Así, tenemos el punto (0.2, 0) para la curva ROC.

A continuación, el umbral $\boldsymbol{t}$ se establece en 0,8, el valor de probabilidad para la tupla 2, por lo que esta tupla ahora también se considera positiva, mientras que las tuplas 3 a 10 se consideran negativas. La etiqueta de clase real de la tupla 2 es positiva, por lo tanto, ahora $\boldsymbol{TP = 2}$. El resto de la fila se puede calcular fácilmente, lo que resulta en el punto (0.4, 0). A continuación, examinamos la etiqueta de clase de la tupla 3 y dejamos que $\boldsymbol{t}$ sea 0.7, el valor de probabilidad devuelto por el clasificador para esa tupla. Por lo tanto, la tupla 3 se considera positiva, sin embargo, su etiqueta real es negativa, por lo que es un falso positivo. Por lo tanto, TP permanece igual y FP aumenta de manera que $\boldsymbol{FP = 1}$. El resto de los valores de la fila también se pueden calcular fácilmente, lo que arroja el punto (0.4, 0.2). El gráfico ROC resultante, al examinar cada tupla, es la línea dentada que se muestra en la Figura 8.19.

Hay muchos métodos para obtener una curva de estos puntos, el más común de los cuales es utilizar un casco convexo. La gráfica también muestra una línea diagonal donde, por cada positivo verdadero de tal modelo, es muy probable que encontremos un falso positivo. Para comparación, esta línea representa el adivinar al azar. **Fin Ejemplo.**

La Figura 8.20 muestra las curvas ROC de dos modelos de clasificación. También se muestra la línea diagonal que representa la adivinación aleatoria. Por lo tanto, cuanto más cercana esté la curva ROC de un modelo a la línea diagonal, menos preciso será el modelo. Si el modelo es realmente bueno, inicialmente es más probable que encontremos verdaderos positivos a medida que avanzamos en la lista clasificada. Por lo tanto, la curva se mueve abruptamente desde cero. Más adelante, a medida que comenzamos a encontrar cada vez menos positivos verdaderos, y cada vez más falsos positivos, la curva se suaviza y se vuelve más horizontal.

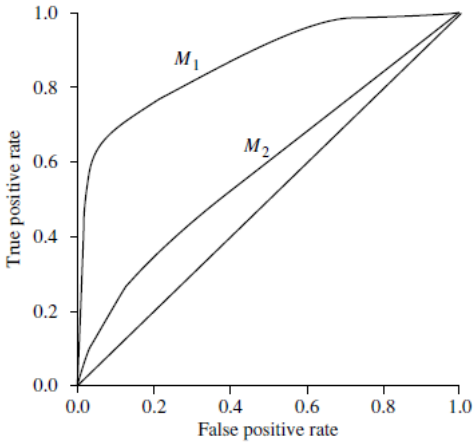

Figura 8.20: Curvas ROC de dos modelos de clasificación, $\boldsymbol{M_1}$ y $\boldsymbol{M_2}$. La diagonal muestra dónde, por cada positivo verdadero, es igualmente probable que encontremos un falso positivo. Cuanto más cerca esté una curva ROC de la línea diagonal, menos preciso será el modelo. Por lo tanto, $\boldsymbol{M_1}$ es más preciso aquí.

Para evaluar la precisión de un modelo, podemos medir el área bajo la curva. Varios paquetes de software son capaces de realizar dicho cálculo. Cuanto más cerca esté el área de 0.5, menos preciso será el modelo correspondiente. Un modelo con perfecta precisión tendrá un área de 1.0.

Técnicas para mejorar la precisión de la clasificación [8.6]

En esta sección, aprenderá algunos trucos para aumentar la exactitud de la clasificación. Nos centramos en los métodos de conjunto (**ensemble**). Un conjunto (**ensemble**) para la clasificación es un modelo compuesto, por una combinación de clasificadores. Los clasificadores individuales votan, y una predicción de etiqueta de clase es devuelta por el conjunto basado en la colección de votos. Los conjuntos tienden a ser más precisos que sus clasificadores de componentes. Comenzamos en la Sección 8.6.1 introduciendo métodos de conjunto (**ensemble**) en general. El **Bagging** (Sección 8.6.2), el **boosting** (Sección 8.6.3) y los bosques aleatorios (**random forests**) (Sección 8.6.4) son métodos de conjunto (**ensemble**) populares.

Los modelos tradicionales de aprendizaje suponen que las clases de datos están bien distribuidas. Sin embargo, en muchos dominios de datos del mundo real, los datos están desbalanceados por clases, donde la clase principal de interés está representada por unas pocas tuplas. Esto se conoce como el problema de desequilibrio de clase. También se estudian técnicas para mejorar la precisión de clasificación de datos desbalanceados por clase. Éstos se presentan en la Sección 8.6.5.

La introducción de métodos de conjunto (Ensemble) [8.6.1]

Bagging, **boostin**, y **random forest** son ejemplos de los métodos de conjunto (**ensemble**) (Figura 8.21). Un conjunto (**ensemble**) combina una serie de $\boldsymbol{k}$ modelos aprendidos (o

clasificadores base) $\boldsymbol{M_1, M_2, \ldots, M_k}$ con el objetivo de crear un modelo de clasificación compuesto mejorado $\boldsymbol{M*}$. Un conjunto de datos dado $\boldsymbol{D}$, se utiliza para crear $\boldsymbol{k}$ conjuntos de entrenamiento $\boldsymbol{D_1, D_2, \ldots, D_k}$, donde $\boldsymbol{D_i}$, $(\boldsymbol{1 < i < k - 1})$ se utiliza para generar el clasificador $\boldsymbol{M_i}$. Dada una nueva tupla de datos para clasificar, los clasificadores de base en cada voto devolviendo una predicción de clase. El conjunto devuelve una predicción de clase basada en los votos de las clasificadoras bases.

Un conjunto (**ensemble**) tiende a ser más preciso que sus clasificadoras bases. Por ejemplo, considere un conjunto (**ensemble**) que realice la votación por mayoría. Es decir, dada una tupla $\boldsymbol{X}$ para clasificar, recoge las predicciones de etiqueta de clase devueltas de las clasificadoras bases y emite la clase en su mayoría. Los clasificadores base pueden cometer errores, pero el conjunto clasificará erróneamente $\boldsymbol{X}$ sólo si más de la mitad de los clasificadores base están en error. Los conjuntos (**ensemble**) producen mejores resultados cuando existe una diversidad significativa entre los modelos. Es decir, idealmente, hay poca correlación entre los clasificadores. Los clasificadores también deben desempeñarse mejor que las suposiciones aleatorias. Cada clasificador de base se puede asignar a una CPU diferente y por lo que los métodos de conjunto (**ensemble**) son paralelizables.

Para ayudar a ilustrar el poder de un conjunto (**ensemble**), considere un problema simple de dos clases descrito por dos atributos, $\boldsymbol{x_1}$ y $\boldsymbol{x_2}$. El problema tiene un límite de decisión lineal. La figura 8.22 (a) muestra el límite de decisión de un clasificador de árbol de decisión sobre el problema. La figura 8.22 (b) muestra el límite de decisión de un conjunto (**ensemble**) de clasificadores de árboles de decisión sobre el mismo problema. Aunque el límite de decisión del conjunto (**ensemble**) sigue siendo constante por partes, tiene una resolución más fina y es mejor que el de un solo árbol.

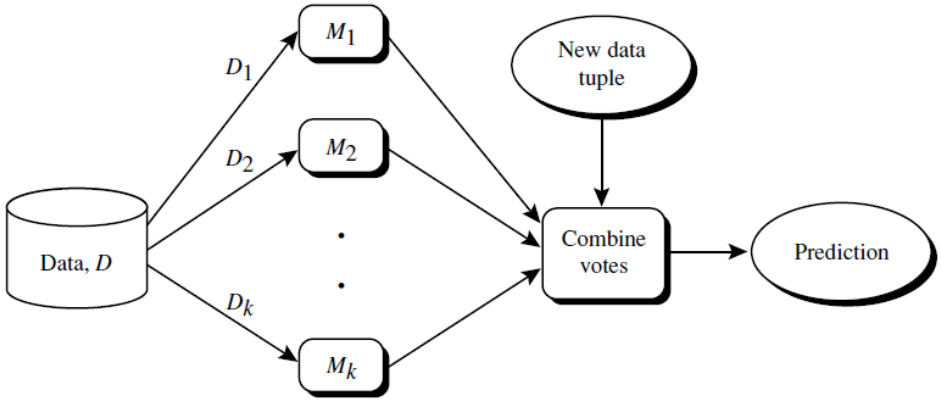

Figura 8.21: Aumento de la precisión del clasificador: Los métodos de conjunto (**ensemble**) generan un conjunto de modelos de clasificación $\boldsymbol{M_1, M_2, \ldots, M_k}$. Dada una nueva tupla de datos para clasificar, cada clasificador "vota" para la etiqueta de clase de esa tupla. El conjunto combina los votos para devolver una predicción de la clase.

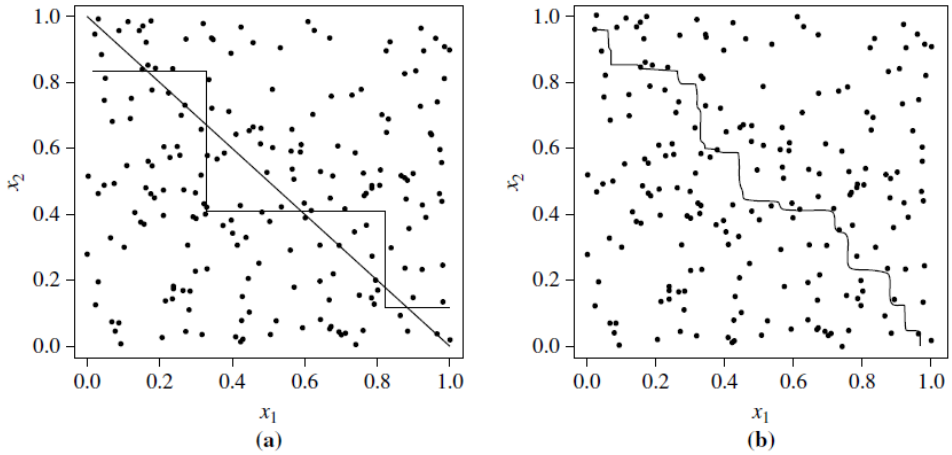

Figura 8.22 Límite de decisión por (a) un árbol de decisión único y (b) un conjunto de árboles de decisión para un problema linealmente separable (es decir, cuando el límite de decisión real es una línea recta). El árbol de decisión tiene problemas para aproximarse a un límite lineal. El límite de decisión del conjunto está más cerca del límite verdadero. Fuente: FromSeni y Elder [SE10]. c Editores de 2010Morgan y Claypool; usado con permiso.

Bagging [8.6.2]

Ahora tomamos una mirada intuitiva de cómo el **bagging** funciona como un método para aumentar la precisión. Supongamos que usted es un paciente y le gustaría tener un diagnóstico basado en sus síntomas. En lugar de preguntarle a un médico, puede elegir preguntarles a varios médicos. Si un cierto diagnóstico ocurre más que cualquier otro, usted puede elegir esto como el diagnóstico final o mejor. Es decir, el diagnóstico final se hace sobre la base de un voto mayoritario, donde cada doctor recibe un voto igual. Ahora reemplace a cada doctor por un clasificador, y usted tiene la idea básica detrás del **bagging**. Intuitivamente, un voto mayoritario hecho por un grupo grande de médicos puede ser más confiable que un voto de la mayoría hecho por un grupo pequeño.

Dado un conjunto $\boldsymbol{D}$, de $\boldsymbol{d}$ tuplas, **bagging** trabaja como sigue. Para cada iteración $\boldsymbol{i}$ $(\boldsymbol{i} = \mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{k})$, un subconjunto de entrenamiento $\boldsymbol{D_i}$, de $\boldsymbol{d}$ tuplas se muestrea con reemplazo del conjunto de datos original de tuplas $\boldsymbol{D}$. Tenga en cuenta que el término **bagging** significa (**bootstrap aggregation**) **agregación de bootstrap**. Cada conjunto de entrenamiento (training) es una muestra bootstrap, como se describe en la sección 8.5.4. Debido a que se usa muestreo con reemplazo, algunas de las tuplas originales de $\boldsymbol{D}$ pueden no ser incluidas en $\boldsymbol{D_i}$, mientras que otras pueden ocurrir más de una vez. Por cada subconjunto de entrenamiento $\boldsymbol{D_i}$, se crea un modelo clasificador $\boldsymbol{M_i}$. Para clasificar una tupla desconocida $\boldsymbol{X}$, cada clasificador $\boldsymbol{M_i}$, devuelve su predicción de clase, la cual cuenta como un voto. El clasificador bagging $\boldsymbol{M}*$, cuenta los votos y asigna la clase con la mayor cantidad de votos a $\boldsymbol{X}$. El bagging se puede aplicar a la predicción de valores continuos tomando el

valor promedio de cada predicción para una tupla de prueba dada. El algoritmo se resume en la figura 8.23.

El clasificador bagging a menudo tiene una precisión significativamente mayor que un único clasificador derivado de $\boldsymbol{D}$, los datos de entrenamiento original. No será considerablemente peor y es más robusto a los efectos de los datos ruidosos y del sobreajuste. La mayor precisión se produce porque el modelo compuesto reduce la varianza de los clasificadores individuales.

Algoritmo Bagging

Algoritmo: Bagging. El algoritmo bagging crea un conjunto de modelos de clasificación para un esquema de aprendizaje donde cada modelo da una predicción igualmente ponderada.

Entrada:

- $\boldsymbol{D}$, un conjunto de $\boldsymbol{d}$ tuplas de entrenamiento;
- $\boldsymbol{k}$, el número de modelos en el conjunto ensemble;
- Un esquema de aprendizaje de clasificación (algoritmo de árboles de decisión, naïve bayesiano, etc.).

Salida: El conjunto un modelo compuesto $\boldsymbol{M}$ $*$.

Método:

1. **Para** $\boldsymbol{i} = \mathbf{1}$ hasta $\boldsymbol{k}$ // crear $\boldsymbol{k}$ modelos:
2. Crear un muestra bootstrap $\boldsymbol{D_i}$, por muestreo $\boldsymbol{D}$ con reemplazo;
3. Use $\boldsymbol{D_i}$ y el esquema de aprendizaje para derivar un modelo $\boldsymbol{M_i}$;
4. **Fin Para**

Para utilizar el conjunto para clasificar un tupla $\boldsymbol{X}$: Que cada uno de los $\boldsymbol{k}$ modelos clasifiquen $\boldsymbol{X}$ y devuelvan el voto mayoritario;

Boosting y AdaBoost [8.6.3]

Ahora miramos del método de ensemble el **boosting**. Como en la sección anterior, suponga que, como paciente, tiene ciertos síntomas. En lugar de consultar a un médico, usted elige consultar varios. Supongamos que se asigna pesos al valor o el valor de diagnóstico de cada médico, basado en la precisión de los diagnósticos anteriores que han hecho. El diagnóstico final es entonces una combinación de los diagnósticos ponderados. Esta es la esencia detrás de **boosting**.

En **boosting,** los pesos son asignados también a cada tupla de entrenamiento. Una serie de clasificadores $\boldsymbol{k}$ se aprende de forma iterativa. Después de que se aprende un clasificador $\boldsymbol{M_i}$,

los pesos se actualizan para permitir que el clasificador subsiguiente M_{i+1}, "preste más atención" a las tuplas de entrenamiento que fueron mal clasificadas por M_i. El clasificador final reforzado $M*$, combina los votos de cada clasificador individual, donde el peso del voto de cada clasificador es una función de su exactitud.

AdaBoost (abreviatura de Adaptive Boosting) es un popular algoritmo de boosting. Supongamos que queremos aumentar la precisión de un método de aprendizaje. Dado D, un conjunto de datos de tuplas con etiqueta de clase d, $(X_1, y_1), (X_2, y_2), \ldots, (X_d, y_d)$, donde y_i es la etiqueta de clase de la tupla X_i. Inicialmente, AdaBoost asigna a cada tupla de entrenamiento un peso igual a $1/d$. Generar k clasificadores para el conjunto (**ensemble**) que requiere k rondas a través del resto del algoritmo. En la ronda i, las tuplas de D se muestrean para formar un conjunto de entrenamiento, D_i, de tamaño d. Se usa el muestreo con reemplazo: la misma tupla se puede seleccionar más de una vez. La probabilidad de que cada tupla sea seleccionada se basa en su peso. Un modelo clasificador, M_i, se deriva de las tuplas de entrenamiento de D_i. Su error luego se calcula usando D_i como un conjunto de prueba. Los pesos de las tuplas de entrenamiento se ajustan según la clasificación.

Si una tupla fue clasificada incorrectamente, su peso aumenta. Si una tupla se clasificó correctamente, su peso disminuye. El peso de una tupla refleja cuán difícil es clasificarlo: cuanto mayor es el peso, más veces se clasifica erróneamente. Estos pesos se usarán para generar las muestras de entrenamiento para el clasificador de la siguiente ronda. La idea básica es que cuando construimos un clasificador, queremos que se centre más en las tuplas mal clasificadas de la ronda anterior. Algunos clasificadores pueden ser mejores para clasificar algunas tuplas "difíciles" que otros. De esta manera, construimos una serie de clasificadores que se complementan entre sí. El algoritmo se resume en la figura 8.24.

Ahora, veamos algunas de las matemáticas que están involucradas en el algoritmo. Para calcular la tasa de error del modelo M_i, sumamos los pesos de cada una de las tuplas en D_i que M_i clasificó incorrectamente. Es decir,

$$error(M_i) = \sum_{j=1}^{d} w_j \times err(X_j) \qquad (8.34)$$

donde $err(X_j)$ es el error de clasificación errónea de la tupla X_j: si la tupla se clasificó erróneamente, entonces $err(X_j)$ es 1; de lo contrario, es 0. Si el rendimiento del clasificador M_i es tan pobre que su error excede 0.5, entonces lo abandonamos. En cambio, lo intentamos de nuevo generando un nuevo conjunto de entrenamiento D_i, del cual obtenemos un nuevo M_i.

La tasa de error de M_i afecta cómo se actualizan los pesos de las tuplas de entrenamiento. Si una tupla en la ronda i fue clasificada correctamente, su peso se multiplica por $\frac{error(M_i)}{(1-error(M_i))}$. Una vez que se actualizan los pesos de todas las tuplas correctamente clasificadas, los pesos para todas las tuplas (incluidos los mal clasificados) se normalizan de modo que su suma permanezca igual que antes. Para normalizar un peso, lo multiplicamos por la suma de los

pesos viejos, dividido por la suma de los nuevos pesos. Como resultado, los pesos de las tuplas mal clasificadas se incrementan y los pesos de las tuplas correctamente clasificadas se reducen, como se describió anteriormente.

"Una vez que se completa el boosting, ¿cómo se usa el conjunto de clasificadores para predecir la etiqueta de clase de una tupla, $\boldsymbol{X}$?" A diferencia del bagging, donde a cada clasificador se le asignó un voto igual, el aumento asigna un peso al voto de cada clasificador, según cuán bien el clasificador realizado. Cuanto menor es la tasa de error de un clasificador, más precisa es, y, por lo tanto, mayor debe ser su peso para votar. El peso del clasificador $\boldsymbol{M_i}$ es el voto

$$\log\left(\frac{1 - error(M_i)}{error(M_i)}\right) \qquad (8.35)$$

Para cada clase, $\boldsymbol{c}$, sumamos los pesos de cada clasificador que asignó la clase $\boldsymbol{c}$ a $\boldsymbol{X}$. La clase con la suma más alta es el "ganador" y se devuelve como la predicción de clase para la tupla $\boldsymbol{X}$.

"¿Cómo se compara boosting con el bagging?" Debido a la forma en que el boosting se centra en las tuplas mal clasificadas, corre el riesgo de sobreajustar el modelo compuesto resultante a tales datos.

Por lo tanto, a veces el modelo resultante "boosted" puede ser menos preciso que un modelo único derivado de los mismos datos. El bagging es menos susceptible al sobreajuste del modelo. Si bien ambos pueden mejorar significativamente la precisión en comparación con un solo modelo, aumentar el boosting tiende a lograr una mayor precisión.

Algorithm: AdaBoost. A boosting algorithm—create an ensemble of classifiers. Each one gives a weighted vote.

Input:

- D, a set of d class-labeled training tuples;
- k, the number of rounds (one classifier is generated per round);
- a classification learning scheme.

Output: A composite model.

Method:

(1) initialize the weight of each tuple in D to $1/d$;
(2) **for** $i = 1$ to k **do** // for each round:
(3) sample D with replacement according to the tuple weights to obtain D_i;
(4) use training set D_i to derive a model, M_i;
(5) compute $error(M_i)$, the error rate of M_i (Eq. 8.34)
(6) **if** $error(M_i) > 0.5$ **then**
(7) go back to step 3 and try again;
(8) **endif**
(9) **for** each tuple in D_i that was correctly classified **do**
(10) multiply the weight of the tuple by $error(M_i)/(1 - error(M_i))$; // update weights
(11) normalize the weight of each tuple;
(12) **endfor**

To use the ensemble to classify tuple, X:

(1) initialize weight of each class to 0;
(2) **for** $i = 1$ to k **do** // for each classifier:
(3) $w_i = log\frac{1-error(M_i)}{error(M_i)}$; // weight of the classifier's vote
(4) $c = M_i(\boldsymbol{X})$; // get class prediction for $\boldsymbol{X}$ from M_i
(5) add w_i to weight for class c
(6) **endfor**
(7) return the class with the largest weight;

Figure 8.24 AdaBoost, a boosting algorithm.

Random Forests [8.6.4]

Ahora presentamos otro método de conjunto (**ensemble**) llamado bosques aleatorios (**Random Forest**). Imagine que cada uno de los clasificadores en el conjunto es un clasificador de árbol de decisiones, de modo que la colección de clasificadores es un "bosque". Los árboles de decisión individuales se generan utilizando una selección aleatoria de atributos en cada nodo para determinar la división. Más formalmente, cada árbol depende de los valores de un vector aleatorio muestreado de forma independiente y con la misma distribución para todos los árboles del bosque. Durante la clasificación, cada árbol vota y se devuelve la clase más popular.

Los bosques aleatorios se pueden construir usando el bagging (Sección 8.6.2) junto con la selección aleatoria de atributos. Se proporciona un conjunto de entrenamiento, $\boldsymbol{D}$, de $\boldsymbol{d}$ tuplas. El procedimiento general para generar $\boldsymbol{k}$ árboles de decisión para el conjunto es el siguiente. Para cada iteración, $\boldsymbol{i}(\boldsymbol{i} = \mathbf{1}, \mathbf{2}, .., \boldsymbol{k})$, se muestrea un conjunto de entrenamiento, $\boldsymbol{D_i}$, de $\boldsymbol{d}$ tuplas con reemplazo de $\boldsymbol{D}$. Es decir, cada $\boldsymbol{D_i}$ es una muestra de arranque de $\boldsymbol{D}$ (Sección 8.5. 4), de modo que algunas tuplas pueden aparecer más de una vez en $\boldsymbol{D_i}$, mientras que otras pueden ser excluidas. Sea $\boldsymbol{F}$ el número de atributos que se utilizarán para determinar la división en cada nodo, donde $\boldsymbol{F}$ es mucho más pequeño que el número de atributos disponibles. Para construir un clasificador de árbol de decisión, $\boldsymbol{M_i}$, seleccione aleatoriamente, en cada nodo, $\boldsymbol{F}$ atributos como candidatos para la división en el nodo. La metodología CART se usa para hacer crecer los árboles. Los árboles crecen al tamaño máximo y no se podan. Los bosques aleatorios formados de esta manera, con selección de entrada aleatoria, se denominan Forest-RI.

Otra forma de bosque aleatorio, llamada Forest-RC, usa combinaciones aleatorias lineales de los atributos de entrada. En lugar de seleccionar al azar un subconjunto de los atributos, crea nuevos atributos (o características) que son una combinación lineal de los atributos existentes. Es decir, se genera un atributo especificando $\boldsymbol{L}$, el número de atributos originales que se combinarán. En un nodo dado, los atributos $\boldsymbol{L}$ se seleccionan aleatoriamente y se suman junto con los coeficientes que son números aleatorios uniformes en [-1, 1]. Se generan combinaciones $\boldsymbol{F}$ lineales, y se realiza una búsqueda sobre estas para la mejor división. Esta forma de bosque aleatorio es útil cuando solo hay unos pocos atributos disponibles, a fin de reducir la correlación entre los clasificadores individuales.

Los bosques aleatorios son comparables en precisión a AdaBoost, pero son más robustos a los errores y valores atípicos. El error de generalización para un bosque converge siempre que la cantidad de árboles en el bosque sea grande. Por lo tanto, el sobreajuste no es un problema. La precisión de un bosque aleatorio depende de la fuerza de los clasificadores individuales y una medida de la dependencia entre ellos. Lo ideal es mantener la fortaleza de los clasificadores individuales sin aumentar su correlación. Los bosques aleatorios son insensibles al número de atributos seleccionados para su consideración en cada división. Normalmente, se eligen hasta $\boldsymbol{log_2 d} + \mathbf{1}$. (Una observación empírica interesante fue que el

uso de un único atributo de entrada aleatorio puede dar como resultado una buena precisión que a menudo es mayor que cuando se usan varios atributos). Debido a que los bosques aleatorios consideran muchos menos atributos para cada división, son eficientes en bases de datos muy grandes. Pueden ser más rápidos que bagging o boosting. Los bosques aleatorios dan estimaciones internas de la importancia de las variables para el modelo.

Mejora de la precisión de clasificación de datos desbalanceados de clase [8.6.5]

En esta sección, revisamos el problema de desequilibrio de clase. En particular, estudiamos enfoques para mejorar la precisión de clasificación de datos desequilibrados por clase.

Dada la información de dos clases, los datos están desequilibrados por clase si la clase principal de interés (la clase positiva) está representada por unas pocas tuplas, mientras que la mayoría de las tuplas representan la clase negativa. Para datos desequilibrados multiclase, la distribución de datos de cada clase difiere sustancialmente donde, nuevamente, la clase principal o clases de interés son raras. El problema del desequilibrio de clase está estrechamente relacionado con el aprendizaje sensible a los costos, donde los costos de los errores, por clase, no son iguales. En el diagnóstico médico, por ejemplo, es mucho más costoso diagnosticar falsamente a un paciente canceroso como saludable (un falso negativo) que diagnosticar erróneamente que un paciente sano tiene cáncer (un falso positivo). Un error falso negativo podría llevar a la pérdida de vidas y, por lo tanto, es mucho más costoso que un error falso positivo. Otras aplicaciones que incluyen datos desequilibrados por clase incluyen la detección de fraudes, la detección de derrames de petróleo a partir de imágenes de radar por satélite y el monitoreo de fallas.

Los algoritmos de clasificación tradicionales pretenden minimizar el número de errores cometidos durante la clasificación. Suponen que los costos de los errores falsos positivos y falsos negativos son iguales. Al asumir una distribución equilibrada de clases y costos de error iguales, por lo tanto, no son adecuados para datos desequilibrados de clase. Las partes anteriores de este capítulo presentaron formas de abordar el problema del desequilibrio de clase. Aunque la medida de precisión asume que el costo de las clases es igual, se pueden usar métricas de evaluación alternativas que consideren los diferentes tipos de clasificaciones. La sección 8.5.1, por ejemplo, presentó sensibilidad o recuperación (la tasa positiva real) y especificidad (la tasa negativa verdadera), lo que ayuda a evaluar qué tan bien un clasificador puede predecir la etiqueta de clase de datos desequilibrados. Las medidas relevantes adicionales discutidas incluyen $\boldsymbol{F_1}$ y $\boldsymbol{F_B}$. La Sección 8.5.6 mostró cómo las curvas ROC grafican la sensibilidad versus a la especificidad de 1 (es decir, la tasa de falsos positivos). Dichas curvas pueden proporcionar información al estudiar el rendimiento de los clasificadores en datos desequilibrados por clase.

En esta sección, observamos los enfoques generales para mejorar la precisión de clasificación de los datos desequilibrados por clase. Estos enfoques incluyen (1) sobremuestreo, (2) submuestreo, (3) movimiento de umbral y (4) técnicas de conjunto (**ensemble**). Los primeros tres no implican ningún cambio en la construcción del modelo de clasificación. Es decir, el

sobremuestreo y la falta de muestreo cambian la distribución de las tuplas en el conjunto de entrenamiento; el movimiento del umbral afecta la forma en que el modelo toma decisiones al clasificar nuevos datos. Los métodos de conjunto (**ensemble**) siguen las técnicas descritas en las Secciones 8.6.2 a 8.6.4. Para facilitar la explicación, describimos estos enfoques generales con respecto al problema de datos de desequilibrio de dos clases, donde las clases de mayor costo son más raras que las clases de menor costo.

Tanto el sobremuestreo como la falta de muestreo cambian la distribución de datos de entrenamiento de modo que la clase rara (positiva) está bien representada. El sobremuestreo funciona volviendo a muestrear las tuplas positivas para que el conjunto de entrenamiento resultante contenga un número igual de tuplas positivas y negativas. El submuestreo funciona disminuyendo el número de tuplas negativas. Elimina al azar las tuplas de la clase mayoritaria (negativa) hasta que haya una cantidad igual de tuplas positivas y negativas.

Ejemplo 8.12 Sobremuestreo y submuestreo. Supongamos que el conjunto de entrenamiento original contiene 100 tuplas positivas y 1000 negativas. En el sobremuestreo, replicamos las tuplas de la clase más rara para formar un nuevo conjunto de entrenamiento que contiene 1000 tuplas positivas y 1000 tuplas negativas. Al hacer un submuestreo, eliminamos al azar de las tuplas negativas de modo que el nuevo conjunto de entrenamiento contenga 100 tuplas positivas y 100 tuplas negativas.

Existen varias variaciones de sobremuestreo y submuestreo. Pueden variar, por ejemplo, en cómo se agregan o eliminan las tuplas. Por ejemplo, el algoritmo SMOTE usa sobremuestreo donde se agregan tuplas sintéticas, que están "cerca de" las tuplas positivas dadas en el espacio tuple.

El enfoque de movimiento de umbral para el problema de desequilibrio de clase no implica ningún muestreo. Se aplica a los clasificadores que, dada una tupla de entrada, devuelven un valor de salida continuo (como en la Sección 8.5.6, donde discutimos cómo construir curvas ROC). Es decir, para una tupla de entrada, $\boldsymbol{X}$, dicho clasificador devuelve como salida una asignación, $\boldsymbol{f}(\boldsymbol{X}) \rightarrow [\mathbf{0}, \mathbf{1}]$. En lugar de manipular las tuplas de entrenamiento, este método devuelve una decisión de clasificación basada en los valores de salida. En el enfoque más simple, tuplas para las cuales $\boldsymbol{f}(\boldsymbol{X}) \geq \boldsymbol{t}$, para un cierto umbral, $\boldsymbol{t}$, se consideran positivos, mientras que todas las demás tuplas se consideran negativas. Otros enfoques pueden implicar manipular las salidas por ponderación. En general, el movimiento del umbral mueve el umbral, t, de modo que las tuplas de clases raras son más fáciles de clasificar (y, por lo tanto, hay menos posibilidades de costosos errores falsos negativos). Ejemplos de tales clasificadores incluyen naïve clasificadores Bayesianos (Sección 8.3) y clasificadores de redes neuronales como retropropagación (Sección 9.2). El método de desplazamiento de umbral, aunque no es tan popular como el sobremuestreo y el exceso de muestreo, es simple y ha mostrado cierto éxito para los datos desequilibrados de dos clases.

Los métodos de conjunto (**ensemble**) (Secciones 8.6.2 a 8.6.4) también se han aplicado al problema de desequilibrio de clase. Los clasificadores individuales que componen el conjunto pueden incluir versiones de los enfoques descritos aquí, como sobremuestreo y movimiento de umbral.

Estos métodos funcionan relativamente bien para el problema de desequilibrio de clase en tareas de dos clases. Se observó empíricamente que los métodos de movimiento de umbral y conjunto (**ensemble**) superaban el sobremuestreo y la submuestreo. El movimiento de umbral funciona bien incluso en conjuntos de datos extremadamente desequilibrados. El problema del desequilibrio de clase en las tareas multiclase es mucho más difícil, donde el sobremuestreo y el desplazamiento del umbral son menos efectivos. A pesar de que los métodos de desplazamiento de umbral y conjunto (**ensemble**) son prometedores, la búsqueda de una solución para el problema del desequilibrio multiclase sigue siendo un área de trabajo futuro.

Resumen [8.7]

- La **clasificación** es una forma de análisis de datos que extrae modelos que describen clases de datos. Un clasificador o modelo de clasificación predice etiquetas categóricas (clases). La predicción numérica modela las funciones de valor continuo. La clasificación y la predicción numérica son los dos tipos principales de problemas de predicción.
- La **inducción del árbol de decisión** es un algoritmo de inducción de árbol recursivo descendente, que utiliza una medida de selección de atributo para seleccionar el atributo probado para cada nodo no hoja en el árbol. ID3, C4.5 y CART son ejemplos de tales algoritmos que utilizan diferentes medidas de selección de atributos. Los algoritmos de poda de árboles intentan mejorar la precisión eliminando las ramas de los árboles que reflejan el ruido en los datos. Los algoritmos iniciales del árbol de decisión suelen suponer que los datos residen en la memoria. Se han propuesto varios algoritmos escalables, como RainForest, para la inducción de árboles escalables.
- La clasificación Naïve Bayesian se basa en el teorema de probabilidad posterior de Bayes. Asume la independencia condicional de clase, que el efecto de un valor de atributo en una clase dada es independiente de los valores de los otros atributos.
- Un **clasificador basado en reglas** usa un conjunto de reglas IF-THEN para la clasificación. Las reglas se pueden extraer de un árbol de decisiones. Las reglas también se pueden generar directamente a partir de datos de entrenamiento utilizando algoritmos de cobertura secuencial.
- Una **matriz de confusión** se puede usar para evaluar la calidad de un clasificador. Para un problema de dos clases, muestra los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Las medidas que evalúan la capacidad predictiva de un clasificador incluyen precisión, sensibilidad (también conocida como memoria), especificidad, precisión, $\boldsymbol{F}$ y $\boldsymbol{F_\beta}$. La confianza en la medida de precisión puede engañar cuando la clase principal de interés es minoritaria.
- La construcción y la evaluación de un clasificador requieren la división de datos etiquetados en un conjunto de capacitación y un conjunto de prueba. **Holdout**, **muestreo aleatorio**, **validación cruzada** y **bootstrapping** son métodos típicos utilizados para dicha partición.

- Las pruebas de significación y las curvas ROC son herramientas útiles para la selección del modelo. Las pruebas de significancia se pueden usar para evaluar si la diferencia de precisión entre dos clasificadores se debe a la posibilidad. Las curvas ROC grafican la tasa positiva real (o sensibilidad) frente a la tasa de falsos positivos (o 1-especificidad) de uno o más clasificadores.
- Los métodos de conjunto (**ensemble**) se pueden usar para aumentar la precisión general aprendiendo y combinando una serie de modelos de clasificadores individuales (base). **Bagging**, **boosting** y bosques aleatorios (**random forest**) son métodos de conjunto (**ensemble**) populares.
- El problema de desequilibrio de clase ocurre cuando la clase principal de interés está representada por solo unas pocas tuplas. Las estrategias para abordar este problema incluyen sobremuestreo, submuestreo, movimiento de umbral y técnicas de conjunto.