Evaluación de modelos

PID_00284573

Raúl Montoliu Colás

Tiempo mínimo de dedicación recomendado: 2 horas

Universitat Oberta de Catalunya

Raúl Montoliu Colás

Ingeniero en Informática por la Universidad Jaume I (UJI) de Castellón. Doctor en métodos avanzados informáticos por la misma universidad. Actualmente trabaja como docente en el departamento de Ingeniería y Ciencia de los Computadores de la UJI y como investigador en el grupo de investigación Machine Learning for Smart Environments del Instituto de Nuevas Tecnologías de la Imagen (INIT). Desde el 2017 colabora como docente en la Universitat Oberta de Catalunya (UOC).

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por el profesor: Julià Minguillón Alfonso

Primera edición: septiembre 2021

Av. Tibidabo, 39-43, 08035 Barcelona
Autoría: Raúl Montoliu Colás
Producción: FUOC

Índice

Introducción

La evaluación de modelos es un proceso fundamental en la minería de datos. Se usa para comprobar en qué medida el modelo que hemos creado es capaz de resolver el problema planteado. Si no es el caso, deberemos tomar decisiones con el fin de obtener un nuevo modelo capaz de resolver de la mejor forma el problema en cuestión.

Este módulo se ha divido en tres partes, correspondientes a los tres tipos principales de problemas de minería de datos: problemas supervisados de clasificación, problemas supervisados de regresión y problemas no supervisados. Para cada tipo, se explicarán las técnicas más comunes usadas para evaluar de forma correcta los modelos creados.

1. Evaluación de modelos supervisados de clasificación

Imaginemos que nos han contratado para diseñar un algoritmo capaz de predecir si un cliente comprará o no un determinado producto. Para ello, necesitamos un conjunto amplio de muestras con los datos de los clientes y, además, para cada una de ellas, es necesario que un experto las etiquete con una de las dos posibilidades: comprará o no comprará el producto. En este caso, el experto etiqueta con +1 las muestras de los clientes que finalmente compraron el producto, y con –1 las muestras de los que no. Usando este conjunto de datos etiquetados, se creará un modelo de clasificación supervisada. Una vez que tengamos el modelo, y dados los datos de un nuevo cliente, podremos usar el modelo para predecir la etiqueta para ese cliente. Es decir, podremos predecir si el cliente comprará o no el producto.

Para saber si el modelo que hemos creado funcionará bien o no, únicamente podemos usar los datos que tenemos etiquetados, ya que, de esta forma, podemos comparar las etiquetas predichas con las reales.

1.1 Evaluación de problemas binarios

Un problema binario es aquel que tiene únicamente dos clases, como por ejemplo el problema planteado anteriormente, en el cual quiere predecirse si un cliente comprará o no un producto.

Para comprobar la calidad del modelo desarrollado, debemos dividir el conjunto original de muestras (las muestras que tenemos etiquetadas) en dos conjuntos: entrenamiento y test. El conjunto de entrenamiento nos servirá para obtener un modelo preliminar que tendrá un comportamiento similar al que podríamos obtener usando todo el conjunto de datos original, pero no igual, puesto que tiene menos muestras. El conjunto de test se usará para validar el modelo preliminar entrenado. El valor que obtengamos será una estimación optimista del resultado que se obtendrá cuando se predigan las futuras muestras reales.

Siguiendo con el ejemplo anterior, supongamos que tenemos una base de datos de 1,000 clientes, de los cuales 500 compraron el producto y 500 no lo compraron. Dividimos el conjunto en 800 muestras de entrenamiento y 200 de test. Es muy importante que el número de elementos de cada clase esté equilibrado en ambos conjuntos. Por ejemplo, un error importante sería que

el conjunto de entrenamiento fuera de 500 clientes que compraron el producto y 300 clientes que no lo compraron; por lo tanto, el conjunto de test tendría únicamente muestras de clientes que no compraron el producto y ninguna de clientes que sí lo compraron. Lo correcto sería, en este caso, que el conjunto de entrenamiento tuviera 400 de cada tipo y, por lo tanto, el de test 100 de cada tipo.

La medida más usada para evaluar la calidad de un modelo es la exactitud, que se define como el número de muestras para las que el modelo ha predicho bien su clase frente al número total de muestras. Formalmente, se define como se muestra a continuación:

$$exactitud = \frac{1}{N}\sum_{i=1}^{N}\lambda(f(x_i),l(x_i)) \quad (1)$$

donde x_i es una muestra del conjunto de muestras por evaluar (conjunto de test), $l(x_i)$ es la etiqueta verdadera de dicha muestra, f es la función de predicción del modelo que devuelve la etiqueta predicha de la muestra x_i y λ es una función que devuelve 1 si la etiqueta predicha $f(x_i)$ es igual a la verdadera $l(x_i)$ y 0 en otro caso. La exactitud obtiene un valor entre 0.0 y 1.0. Cuanto más cercano sea el valor obtenido a 1.0, mejor comportamiento presenta el modelo.

Supongamos que hemos creado un modelo con las 800 muestras de entrenamiento. Para obtener una medida de la calidad del modelo, predecimos la clase de las 200 muestras de test: obtenemos, por ejemplo, 180 muestras en las que se ha predicho la clase correcta y 20 en las que no. En este caso, la exactitud será 180/200 = 0.9. Es decir, el modelo acierta en el 90 % de los casos.

Una cuestión importante es cómo hacemos la división entre los dos conjuntos. Una posibilidad es realizar la división al azar. Sin embargo, podría ocurrir que justo un tipo de clientes muy particular no estuviera presente en uno de los conjuntos. Por ejemplo, imaginemos que da la casualidad de que los clientes de edad avanzada que sí compraron el producto caen todos en el conjunto de test, y ninguno en el de entrenamiento. El sistema podría confundirse al intentar predecir un cliente con esas características, puesto que no ha sido entrenado para ese tipo de muestras.

Para evitar este problema, se usa una técnica conocida como validación cruzada.

1.1.1 Validación cruzada

La validación cruzada (o *cross validation*) es una técnica para estimar el error que produce un modelo. La técnica consiste en dividir el conjunto de muestras en varias carpetas (o *folds*), cada una con un número similar de muestras de cada clase.

En el problema planteado, podríamos usar 10 carpetas. Por lo tanto, en cada carpeta habría 50 muestras de clientes que sí compraron el producto y 50 muestras de clientes que no lo compraron. Para cada carpeta, se entrena con las muestras pertenecientes a todas las carpetas menos la actual (es decir, con 900 muestras) y se valida con la actual (con 100 muestras), y se obtiene un valor de exactitud. De esta forma, para validar la primera carpeta F_1, el conjunto de entrenamiento estaría compuesto por las muestras pertenecientes al resto de carpetas, es decir, con $\{F_2,F_3,\ldots,F_{10}\}$. De forma similar, para validar la quinta carpeta F_5, el conjunto de entrenamiento estaría compuesto por las muestras pertenecientes a las carpetas $\{F_1,\ldots,F_4,F_6,\ldots,F_{10}\}$. Por último, para validar la última carpeta F_{10}, el conjunto de entrenamiento estaría compuesto por las muestras pertenecientes a las carpetas $\{F_1,\ldots,F_9\}$.

Para cada carpeta se obtiene un valor de exactitud. La exactitud total es el promedio de las exactitudes obtenidas en todas las carpetas.

Un caso particular es el método *leaving one out*, en el que las carpetas tienen únicamente una muestra. Para cada muestra, se entrena con el resto y se valida si acierta o no. La exactitud total será el número de aciertos dividido por el total de muestras.

1.2 Evaluación de problemas binarios no equilibrados

Un problema binario no equilibrado es aquel en el que el número de muestras de una clase es muy superior al número de muestras de la otra. Supongamos, por ejemplo, que existe un sistema de aprendizaje automático capaz de detectar de forma temprana una enfermedad mortal. El sistema toma como entrada un conjunto de datos del paciente y devuelve 1 si el paciente tiene la enfermedad, y –1 en el caso contrario. En el caso de detectar la enfermedad, podrán tomarse las medidas oportunas para aumentar las probabilidades de superarla. Sin embargo, la medicación tiene unos efectos secundarios muy desagradables, por lo que no se recomienda que un paciente sano la tome.

En este ejemplo, la clase objetivo es la clase positiva, es decir, detectar que el paciente tiene la enfermedad. La otra clase es la clase negativa.

Con el fin de comprobar si el sistema desarrollado funciona correctamente, se han realizado 10,000 predicciones, de las cuales 9,500 eran de pacientes sanos y 500 de pacientes con la enfermedad. El sistema ha predicho como sanos a 9,450 de los pacientes sanos, y como enfermos a 300 pacientes con la enfermedad. Por lo tanto, el sistema ha acertado en 9,450 + 300 = 9,750 casos de los 10,000. Una posible medida de la calidad del proceso es, tal como se ha explicado en el apartado anterior, calcular la exactitud del sistema. En este caso es 9,750/10,000 = 0.975. Es decir, el sistema acierta en un 97.5 % de los casos.

El valor obtenido para la exactitud podría llevarnos a confusión: aunque ciertamente es un valor muy alto, en la clase objetivo únicamente ha acertado el 60 % de los casos (300 de 500). En realidad, el resultado es muy malo, pues hay 200 pacientes (el 40 %) que tienen la enfermedad y el sistema no lo ha detectado. En este tipo de problemas, es crucial acertar la gran mayoría de los casos de la clase objetivo, aunque ello implique aumentar ligeramente los fallos en la clase negativa. En esta situación, resulta desagradable medicar a un paciente sano (errar en la clase negativa), pero puede resultar mortal no medicar a uno enfermo (errar en la clase positiva). En nuestro ejemplo, hay 50 personas que sufrirían los efectos secundarios por error, pero 200 personas que podrían fallecer por no medicarse.

En este apartado, se presentará una medida para evaluar este tipo de problemas llamada *F-measure*, *F-score* o *F1-score*, que tiene en cuenta lo bueno o malo que es el modelo a la hora de predecir correctamente la clase objetivo.

Previamente, hay que definir cuatro conceptos importantes:

- *Verdadero positivo* (TP): es una muestra positiva que el sistema ha predicho como positiva.
- *Verdadero negativo* (TN): es una muestra negativa que el sistema ha predicho como negativa.
- *Falso positivo* (FP): es una muestra negativa que el sistema ha predicho como positiva.
- *Falso negativo* (FN): es una muestra positiva que el sistema ha predicho como negativa.

En los casos de TP y TN el modelo acierta en la predicción, mientras que en los otros dos casos el modelo se equivoca.

A partir de las definiciones anteriores, se obtienen los siguientes valores:

- *Precisión*: también llamada *valor de la predicción positiva*, es la fracción de muestras positivas predichas como positivas (TP) entre el total de muestras

predichas como positivas (TP + FP) y se define como:

$$Precisión = \frac{TP}{TP + FP} \quad (2)$$

- *Sensibilidad*: también llamada *recall*, es la fracción de muestras positivas predichas como positivas (TP) entre el total de muestras realmente positivas (TP + FN) y se define como:

$$Sensibilidad = \frac{TP}{TP + FN} \quad (3)$$

Finalmente, la medida *F-measure* se obtiene mediante la siguiente ecuación:

$$2 \times \frac{Precisión \times Sensibilidad}{Precisión + Sensibilidad} \quad (4)$$

Esta medida obtiene un valor entre 0.0 (mal resultado) y 1.0 (buen resultado).

Volviendo al ejemplo planteado al inicio de este subapartado, los valores de TP, TN, FP, FN, Precisión, Sensibilidad y *F-measure* se muestran en la segunda columna de la tabla 1. Como puede comprobarse, el valor de *F-measure* no es muy elevado, debido a la deficiente capacidad del modelo a la hora de predecir los verdaderos positivos.

Tabla 1. Dos posibles resultados de un problema binario no equilibrado

Medida	Modelo original	Modelo mejorado
TP	300	450
TN	9,450	9,250
FP	50	250
FN	200	50
Exactitud	$\frac{9,450+300}{10,000} = 0.975$	$\frac{9,250+450}{10,000} = 0.97$
Precisión	$\frac{300}{300+50} = 0.86$	$\frac{450}{450+250} = 0.64$
Sensibilidad	$\frac{300}{300+200} = 0.6$	$\frac{450}{450+50} = 0.9$
F-measure	$2 \times \left(\frac{0.86 \times 0.6}{0.86+0.6}\right) = 0.71$	$2 \times \left(\frac{0.9 \times 0.64}{0.9+0.64}\right) = 0.75$

Tras ver los resultados, los expertos deciden modificar el modelo para que evite tantos falsos negativos. El nuevo modelo predice ahora como sanos a 9,250 de los pacientes sanos, y como enfermos a 450 pacientes con la enfermedad. Por lo tanto, con el nuevo modelo, el sistema ha acertado en 9,250 + 450 = 9,700 casos de los 10,000. En este caso, la exactitud es 9,700/10,000 = 0.97. Es decir, el sistema acierta en un 97.0 % de los casos, que, como puede comprobarse, es un poco menos que en el modelo original. Sin embargo, tal como muestra la tercera columna de la tabla 1, la *F-measure* es mejor en este caso y, por lo tanto, el nuevo modelo es preferible al original.

1.3 Evaluación de problemas multiclase

Frecuentemente, nos encontramos ante problemas de clasificación supervisada que tienen más de una clase. A este tipo de problemas les llamamos *problemas multiclase*. Por ejemplo, un posible problema multiclase es aquel en el que, dada una imagen donde aparece un coche, se predice la marca de este. Es un problema multiclase, puesto que hay más de dos posibles marcas.

Existen dos estrategias principales para validar problemas multiclase: *one versus one* y *one versus all*. En ambos casos, se transforma el problema multiclase en múltiples problemas binarios.

En la primera estrategia, *one versus one*, se calcula una medida de bondad del modelo (como la exactitud o la *F-measure* explicadas anteriormente) para cada par de clases. Por ejemplo, si el problema tiene cuatro clases $\{C_1,C_2,C_3,C_4\}$, se calculará la medida de bondad del modelo para todos los posibles problemas binarios que pueden plantearse con las cuatro clases, es decir: $\{C_1\}^+$ vs $\{C_2\}^-$, $\{C_1\}^+$ vs $\{C_3\}^-$, ..., $\{C_3\}^+$ vs $\{C_4\}^-$, donde $\{\cdot\}^+$ hace referencia a las etiquetas que forman parte de la clase positiva y $\{\cdot\}^-$ a las etiquetas que forman parte de la clase negativa. La medida de bondad final del problema multiclase será el promedio de las medidas de bondad obtenidas para todos los problemas binarios planteados.

La segunda estrategia, *one versus all*, consiste en formar tantos problemas binarios como clases existan, de forma que la clase positiva estará compuesta por las muestras de una de las clases existentes y la negativa con las muestras pertenecientes al resto de clases. Siguiendo con un ejemplo con cuatro clases, se calculará la medida de bondad del modelo para los siguientes cuatro problemas binarios: $\{C_1\}^+$ vs $\{C_2,C_3,C_4\}^-$, $\{C_2\}^+$ vs $\{C_1,C_3,C_4\}^-$, $\{C_3\}^+$ vs $\{C_1,C_2,C_4\}^-$ y $\{C_4\}^+$ vs $\{C_1,C_2,C_3\}^-$. Como en la estrategia anterior, la medida de bondad final del problema multiclase será el promedio de las medidas de bondad obtenidas para todos los problemas binarios planteados.

En realidad, la mayoría de las implementaciones de los algoritmos de clasificación supervisada son capaces de tratar con problemas multiclase de forma transparente para el usuario. Según el método, usan internamente una de las dos estrategias comentadas.

El usuario deberá seguir las mismas recomendaciones que se han comentado para los problemas binarios. Por un lado, a la hora de seleccionar la medida de bondad más adecuada, debe tenerse en cuenta si el número de muestras de alguna de las clases es muy diferente al del resto. Por otro lado, al aplicar la validación cruzada, habrá que tener especial cuidado de que el número de muestras de cada clase incluidas en cada carpeta sea similar.

1.4 Herramientas de interés

Además de las técnicas comentadas en los subapartados anteriores, existe un conjunto de herramientas que pueden ayudarnos a la hora de interpretar los resultados obtenidos en la validación de un modelo. Las más comunes son las matrices de confusión y las curvas ROC (*receiver operating characteristic*).

1.4.1 Matriz de confusión

La matriz de confusión es una forma gráfica de comprobar lo bien o mal que ha funcionado un modelo. Para los problemas binarios, es una matriz de dos por dos, donde en la primera fila se pondrán los verdaderos negativos y los falsos positivos, y en la segunda fila se pondrán los falsos negativos y los verdaderos positivos. Por lo tanto, la matriz de confusión puede escribirse tal como aparece en la tabla 2. La tabla 3 muestra la matriz de confusión del ejemplo correspondiente a la segunda columna de la tabla 1.

Tabla 2. Matriz de confusión para problemas binarios

		Predicción	
		Negativo	Positivo
Valor real	Negativo	TN	FP
	Positivo	FN	TP

Un buen modelo será el que tiene valores grandes en la diagonal principal y cercanos a cero en el resto de posiciones de la matriz.

Tabla 3. Matriz de confusión del ejemplo correspondiente a la segunda columna de la tabla 1

		Predicción	
		Negativo	Positivo
Valor real	Negativo	9,450	50
	Positivo	200	300

En problemas multiclase, la matriz de confusión tendrá tantas filas y columnas como número de clases haya. De forma similar al caso de los problemas binarios, obtendremos mejores modelos cuando la diagonal principal tenga valores altos y el resto de posiciones de la matriz estén cercanas a cero.

Tabla 4. Matriz de confusión de un hipotético problema multiclase con 4 clases

		Predicción			
		A	B	C	D
Valor real	A	50	5	5	40
	B	0	70	2	3
	C	5	0	50	0
	D	10	0	0	60

La tabla 4 muestra un hipotético resultado para un problema de clasificación multiclase con cuatro posibles clases: A, B, C y D. En este ejemplo, el modelo ha sido capaz de predecir correctamente (la diagonal principal)

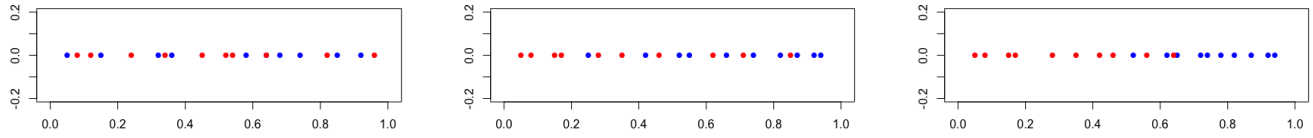
50 + 70 + 50 + 60 = 230 de las 300 existentes. Es decir, el modelo obtiene una precisión del 76.6 % (230/300 = 0.766). Un análisis más detallado de los resultados obtenidos nos permite comprobar que de las 100 muestras existentes de la clase A, únicamente 50 han sido predichas correctamente. De las 50 predichas incorrectamente, la gran mayoría (40) han sido predichas como pertenecientes a la clase D. Gracias a la matriz de confusión obtenida, podemos deducir que existe un problema de confusión entre las muestras de las clases A y D. El modelo tiene un comportamiento más correcto en las muestras del resto de clases. Por lo tanto, en este caso, y tras analizar la matriz de confusión resultante, deberíamos centrar nuestros esfuerzos en averiguar la razón por la que tantas muestras de la clase A se predicen como pertenecientes a la clase D.

1.4.2 Curvas ROC

Las curvas ROC (*receiver operating characteristic*) son un método muy efectivo para validar el funcionamiento de un modelo de clasificación supervisada en problemas binarios. Imaginemos que estamos usando un clasificador supervisado que, en vez de obtener un +1 cuando predice que la muestra es positiva y –1 cuando es negativa, nos proporciona un número entre 0.0 y 1.0 que indica la probabilidad de que la muestra pertenezca a la clase positiva. Por ejemplo, si el resultado es 0.2, significará que hay una probabilidad pequeña de que la muestra sea positiva. Sin embargo, si el resultado es 0.8, significará que hay una probabilidad alta de que la muestra sea positiva. En realidad, la gran mayoría de algoritmos de clasificación supervisada funcionan de esta forma.

Supongamos que, para un problema de clasificación supervisada binario, hemos entrenado tres algoritmos diferentes, y que para un conjunto de 20 muestras de test (10 de cada clase), hemos obtenido la probabilidad de que la muestra sea de la clase positiva. La figura 1 muestra el resultado obtenido para los tres modelos entrenados. Los puntos rojos son las muestras negativas y los puntos azules son las muestras positivas. La figura 1a muestra un mal resultado, puesto que resulta muy difícil encontrar una frontera entre las muestras de ambas clases. La figura 1b presenta un resultado intermedio y la figura 1c muestra el mejor resultado de los tres, ya que, en este caso, es más fácil establecer una frontera entre ambas clases.

Figura 1. Resultados de tres modelos diferentes de clasificación supervisada. Los puntos rojos son las muestras negativas y los puntos azules son las muestras positivas. Se presenta para cada muestra la probabilidad de que pertenezca a la clase positiva.



a. A la izquierda, mal resultado. **b.** En el centro, resultado intermedio. **c.** A la izquierda, buen resultado

Las tablas 5, 6 y 7 muestran, respectivamente, los TP, TN, FP y FN para los tres modelos entrenados, para un conjunto de umbrales de 0.0 a 1.0. Estas tablas también muestran la sensibilidad y 1 menos la especificidad de cada modelo.

La sensibilidad (o *recall*) cuantifica la proporción de muestras positivas (TP + FN) que son clasificadas como positivas (TP) y se ha definido anteriormente como:

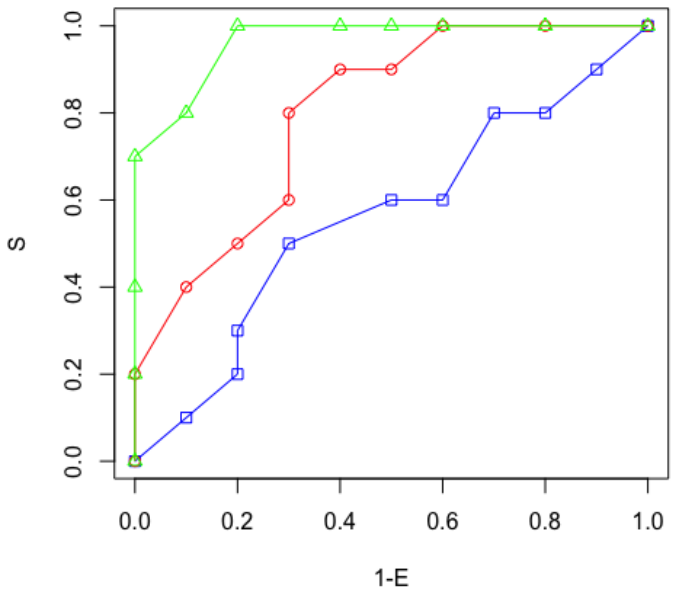
$$S = \frac{TP}{TP + FN} \tag{5}$$

Por otro lado, la especificidad cuantifica la proporción de muestras negativas (TN + FP) que son clasificadas como negativas (TN) y se define como:

$$E = \frac{TN}{TN + FP} \tag{6}$$

La figura 2 muestra las curvas ROC de los tres modelos. La curva ROC representa para cada umbral un punto donde la coordenada x es la sensibilidad y la coordenada y es 1 menos la especificidad. Una medida de la calidad del modelo es el área que queda debajo de la curva. Como puede comprobarse, dicha área será mayor en el modelo representado por la curva verde (muestras de la figura 1c y la tabla 7) que en el modelo intermedio representado por la curva roja (muestras de la figura 1b y la tabla 6), y el peor modelo será el representado por la curva azul (muestras de la figura 1a y la tabla 5).

Figura 2. Curvas ROC de los tres modelos entrenados (véase figura 1). Las curvas verde, roja y azul se corresponden con el mejor, medio y peor modelo, respectivamente.



En términos generales, cuanto más se parezca la curva ROC a una diagonal, peor modelo se habrá obtenido. Sin embargo, cuanto más a la izquierda y arriba esté la curva, mejor modelo se habrá obtenido.

Tabla 5. TP, TN, FP, FN, sensibilidad y 1 menos la especificidad del peor modelo (representado en la figura 1a), para cada uno de los valores del umbral.

Umbral	TP	TN	FP	FN	S	1-E
0.0	10	0	10	0	1.0	1.0
0.1	9	1	9	1	0.9	0.9
0.2	8	2	8	2	0.8	0.8
0.3	8	3	7	2	0.8	0.7
0.4	6	4	6	4	0.6	0.6
0.5	6	5	5	4	0.6	0.5
0.6	5	7	3	5	0.5	0.3
0.7	3	8	2	7	0.3	0.2
0.8	2	8	2	8	0.2	0.2
0.9	1	9	1	9	0.1	0.1
1.0	0	10	0	10	0.0	0.0

Tabla 6. TP, TN, FP, FN, sensibilidad y 1 menos la especificidad del modelo intermedio (representado en la figura 1b), para cada uno de los valores del umbral.

Umbral	TP	TN	FP	FN	S	1-E
0.0	10	0	10	0	1.0	1.0
0.1	10	2	8	0	1.0	0.8
0.2	10	4	6	0	1.0	0.6
0.3	9	5	5	1	0.9	0.5
0.4	9	6	4	1	0.9	0.4
0.5	8	7	3	2	0.8	0.3
0.6	6	7	3	4	0.6	0.3
0.7	5	8	2	5	0.5	0.2
0.8	4	9	1	6	0.4	0.1
0.9	2	10	0	8	0.2	0.0
1.0	0	10	0	10	0.0	0.0

Tabla 7. TP, TN, FP, FN, sensibilidad y 1 menos la especificidad del mejor modelo (representado en la figura 1c), para cada uno de los valores del umbral.

Umbral	TP	TN	FP	FN	S	1-E
0.0	10	0	10	0	1.0	1.0
0.1	10	2	8	0	1.0	0.8
0.2	10	4	6	0	1.0	0.6
0.3	10	5	5	0	1.0	0.5
0.4	10	6	4	0	1.0	0.4
0.5	10	8	2	0	1.0	0.2
0.6	8	9	1	2	0.8	0.1
0.7	7	10	0	3	0.7	0.0
0.8	4	10	0	6	0.4	0.0
0.9	2	10	0	8	0.2	0.0
1.0	0	10	0	10	0.0	0.0

2. Evaluación de modelos supervisados de regresión

Un problema supervisado de regresión se diferencia de un problema supervisado de clasificación en que lo que se predice es un valor continuo, y no una etiqueta entre un conjunto finito. Por ejemplo, un problema de regresión es predecir el valor de venta de una casa. El valor de la casa es un número real positivo que es una cantidad continua. Si se discretiza este valor en varios segmentos, entonces transformaríamos el problema en uno de clasificación supervisada.

Un concepto importante, a la hora de validar la calidad de un modelo de regresión, es el residuo o error de una muestra, que se define como la diferencia entre el valor real de la muestra y_i y el valor predicho por el modelo $\hat{y}_i$. A partir del residuo, pueden definirse las dos medidas más comunes que suelen usarse: el error cuadrático medio (ECM o RMS, en inglés) y la raíz del error cuadrático medio (RECM o RMSE, en inglés); se definen como se muestra a continuación:

$$ECM = \sum_{i=1}^{N} \frac{1}{N}(\hat{y}_i - y_i)^2 \quad (7)$$

$$RECM = \sqrt{ECM} \quad (8)$$

donde N es el número de muestras del conjunto de datos.

Para realizar correctamente la validación del modelo de regresión, puede usarse la validación cruzada. En este caso, también es muy importante distribuir correctamente las muestras entre las diferentes carpetas. Supongamos que nuestra base de datos tiene 800 muestras de casas con precios entre 100,000 y 500,000 euros. Para distribuir correctamente las muestras, podemos crear arbitrariamente tres grupos diferentes según el precio de las casas. Por ejemplo, el primer grupo estaría compuesto por las casas entre 100,000 y 200,000 euros; el segundo grupo, entre 200,000 y 300,000 euros, y el último grupo, con las de más de 300,000. El siguiente paso sería contar el número de muestras de cada grupo. Por ejemplo, supongamos que tenemos 400, 300 y 100 muestras en cada grupo. El siguiente paso es distribuir las muestras de cada grupo en las diferentes carpetas intentando que el número de muestras de cada grupo en cada carpeta sea similar. Siguiendo con el ejemplo y asumiendo que usamos 10 carpetas, pondríamos 40, 30 y 10 muestras de cada grupo en cada carpeta.

3. Evaluación de modelos no supervisados

Los modelos no supervisados, como los métodos de agrupamiento, agregación o *clustering*, también deben evaluarse para validar la calidad del agrupamiento obtenido. Sin embargo, al contrario de los métodos supervisados, en los métodos no supervisados resulta complicado definir cuál es el resultado correcto, puesto que los únicos datos de los que partimos son las propias muestras, sin la existencia de información adicional que pueda confirmarnos si cada muestra ha sido asignada al grupo o clúster correcto. Existe un conjunto de técnicas que pueden usarse con el objetivo de validar la calidad del agrupamiento o el método.

Si queremos evaluar la calidad del agrupamiento obtenido, debemos usar técnicas de validación interna. En la validación interna se utilizan un conjunto de índices de calidad que solo dependen de los datos de los que se parte. Lo que se busca es obtener un índice de cómo de bien o mal se han agrupado las muestras en los diferentes grupos. Estas técnicas también pueden usarse para obtener el número óptimo de grupos en algoritmos como el *k-means*.

Si lo que queremos es hacer una comparativa entre varios algoritmos de agrupamiento, podemos usar técnicas de validación externa. Para ello, será necesario disponer de información adicional, como por ejemplo la etiqueta de clase de cada muestra. En la práctica, no se dispone de esta información, puesto que estamos tratando con problemas no supervisados. Sin embargo, y con el único fin de comprobar si el algoritmo de agrupamiento que hemos desarrollado obtiene buenos resultados, podemos usar una base de datos que incluya información de la clase a la que pertenece cada muestra (definida para problemas supervisados), y usar esa información para comprobar la calidad del agrupamiento obtenida con nuestro algoritmo.

3.1 Técnicas de validación interna

Las técnicas de validación interna se fundamentan en el cálculo de un conjunto de índices basados únicamente en información obtenida de los propios datos. En términos generales, estas técnicas se basan en los dos criterios siguientes:

- *Cohesión*: una muestra perteneciente a un grupo debe estar cerca del resto de muestras del mismo grupo.

- *Separación*: una muestra perteneciente a un grupo debe estar lejos de las muestras pertenecientes a los otros grupos.

Por lo tanto, basándose en estos dos criterios, un buen agrupamiento es el que maximiza tanto la cohesión como la separación. Es decir, cuando las muestras de un grupo están cerca de las muestras de su propio grupo y lejos de las muestras del resto de grupos.

A continuación, se presenta un conjunto de índices que pueden usarse para medir la calidad de un agrupamiento. En un caso real, es recomendable probar varios índices diferentes para poder obtener más información sobre la calidad del agrupamiento conseguido.

Figura 3. Dos ejemplos de posibles resultados de un modelo de agrupamiento. El resultado de la izquierda (**a**) es claramente mejor que el de la derecha (**b**).

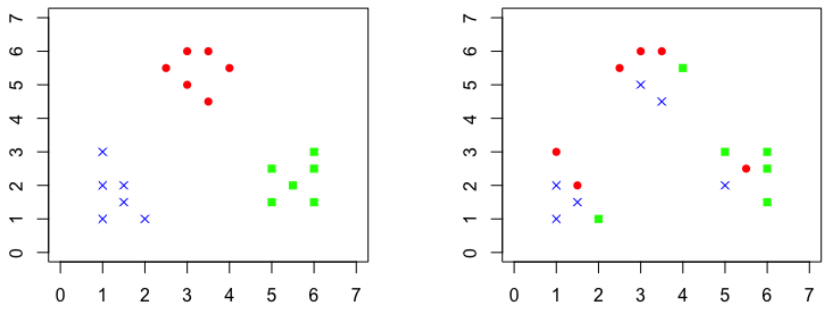
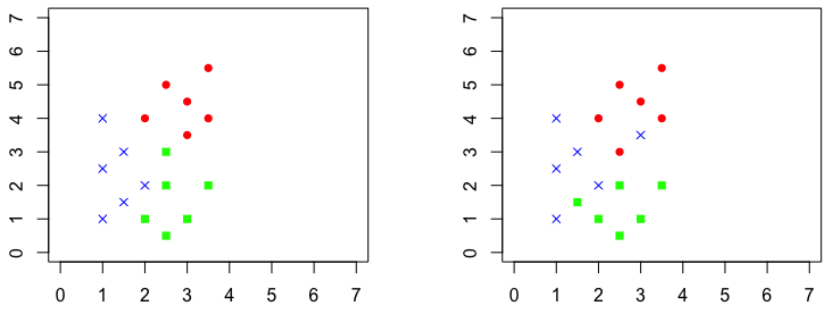


Figura 4. Dos ejemplos de posibles resultados de un modelo de agrupamiento. No está tan claro, como en el caso mostrado en la figura 3, qué resultado es mejor.



Para facilitar la comprensión de cada técnica, se presentan dos resultados diferentes de un modelo de agrupamiento para dos conjuntos de datos diferentes. Las figuras 3a y 3b muestran el primer conjunto de datos. La figura de la izquierda (figura 3a) presenta un resultado de mejor calidad que la figura de la derecha (figura 3b), por lo que es de esperar que los índices que se obtengan sean mejores en el caso de la izquierda que en el de la derecha. Las figuras 4a y

4b muestran el segundo conjunto de datos. En este caso, no está tan claro qué modelo de agrupamiento ha conseguido mejores resultados. Serán los valores que obtengamos de los índices los que nos resolverán la duda.

3.1.1 *Sum of squared within* (SSW)

Este índice se usa para medir la cohesión de los grupos obtenidos. Se obtiene mediante el uso de la siguiente ecuación:

$$SSW = \sum_{i=1}^{k} \sum_{x_j \in G_i} (x_j - \mu_i)^2 \tag{9}$$

donde k es el número de clústers, x_j una muestra del grupo G_i y μ_i es el centroide del i-ésimo grupo G_i.

Cuanto menor sea el número obtenido, más cohesionados estarán los grupos, puesto que las distancias entre las muestras y su centroide serán menores. Hay que tener en cuenta que el valor conseguido depende del número de muestras de cada grupo. En el ejemplo mostrado en las figuras 3 y 4, todos los grupos tienen el mismo número de muestras, lo que facilita la comparación. Pero en un caso real no tiene por qué suceder lo mismo. De hecho, lo más normal será que no ocurra.

La primera fila de la tabla 8 muestra los valores obtenidos del índice *SSW* para los ejemplos mostrados en las figuras 3 y 4. Tal como se esperaba, el resultado conseguido confirma que el mejor agrupamiento de los cuatro ejemplos es el presentado en la figura 3a, que es muy superior al valor obtenido para la otra agrupación con el mismo conjunto de datos, mostrada en la figura 3b. En el caso de las dos agrupaciones presentadas en las figuras 4a y 4b, el resultado está más igualado, y es ligeramente preferible el ejemplo de la figura 4a.

3.1.2 *Sum of squared between* (SSB)

Este índice se usa para medir la separación entre los grupos obtenidos. Se obtiene mediante el uso de la siguiente ecuación:

$$SSB = \sum_{i=1}^{k} |G_i|(\mu - \mu_i)^2 \tag{10}$$

donde k es el número de clústers, $|G_i|$ es el número de muestras del grupo G_i, μ_i es el centroide del i-ésimo grupo G_i y μ es la media de todo el conjunto de datos.

Cuanto mayor sea el número, más separación habrá entre los grupos. Como ocurría en el caso del índice SSW, el valor obtenido depende del número de muestras.

La segunda fila de la tabla 8 muestra los valores obtenidos del índice *SSB* para los ejemplos mostrados en las figuras 3 y 4. Al igual que ocurría con el índice SSW, el mejor agrupamiento de los cuatro ejemplos es el presentado en la figura 3a. De forma similar, en el caso de las dos agrupaciones mostradas en las figuras 4a y 4b, el resultado está de nuevo más igualado, y también es ligeramente preferible el ejemplo de la figura 4a.

3.1.3 Índices basados en SSW y SSB

A partir de los dos índices anteriores SSW y SSB, puede obtenerse otro conjunto de índices para valorar la calidad del agrupamiento conseguido por un modelo. De forma general, cuanto menor sea el valor de SSW y mayor el de SSB, ello significará un mejor agrupamiento. El artículo de Zhao y Fränti (2014) presenta un estudio muy completo de índices que pueden usarse para valorar la calidad del agrupamiento. A continuación, se presenta un subconjunto de los más utilizados:

- *Ball and Hall* (Ball y Hall, 1965): $\frac{SSW}{k}$
- *Caliński and Harabasz* (Caliński y Harabasz, 1974): $\frac{SSB}{k-1}$
- *Hartigan* (Hartigan, 1975): $\log\left(\frac{SSB}{SSW}\right)$
- *XU-index* (Xu, 1997): $d \times \log\left(\sqrt{\frac{SSW}{dN^2}}\right) + \log(k)$
- *WB-index* (Zhao y otros, 2009): $k \times \frac{SSW}{SSB}$

donde k es el número de grupos, N es el número total de muestras y d es la dimensión del problema. En los ejemplos mostrados en las figuras 3 y 4, $k = 3$, $N = 18$ y $d = 2$.

Para los índices *Ball and Hall*, *XU-index* y *WB-index*, son preferibles los valores bajos. Sin embargo, en el caso de los índices *Caliński and Harabasz* y *Hartigan*, los valores altos indican una mejor calidad del agrupamiento.

La tabla 8 muestra los valores que se obtienen de los índices anteriores para los dos ejemplos mostrados en las figuras 3 y 4. Como puede comprobarse, y tal como era esperado, el primer ejemplo alcanza siempre los mejores resultados. En el caso de las dos agrupaciones mostradas en las figuras 4a y 4b, el resultado está siempre muy igualado, y es ligeramente preferible el ejemplo de la figura 4a.

Tabla 8. Resultados obtenidos usando los índices más habituales y los datos mostrados en las figuras 3 y 4. El mejor resultado para cada índice se muestra en negrita.

Índice	Figura 3a	Figura 3b	Figura 4a	Figura 4b
SSW	9.38	82.20	16.63	19.04
SSB	102.72	27.64	34.56	32.13
BH	3.78	27.40	5.54	6.34
CH	78.34	2.52	15.59	12.65
H	2.34	–1.09	0.73	0.52
XU	8.47	10.60	9.00	9.13
WB	0.29	8.92	1.44	1.78
DB	1.35	18.60	9.51	9.76
SC	0.72	–0.01	0.30	0.22

3.1.4 Davies Bouldin

El índice de Davies Bouldin (DB) no está directamente relacionado con los índices SSW y SSB, aunque los principios en los que se basa son similares. Para calcular este índice, se usa la siguiente ecuación:

$$DB = \frac{1}{k}\sum_{i=1}^{k} R_i \tag{11}$$

$$R_i = \max_{j=1,\ldots,k} R_{ij} \tag{12}$$

$$R_{ij} = \frac{S_i + S_j}{d_{ij}} \tag{13}$$

donde d_{ij} es la distancia entre los centroides de los grupos G_i y G_j, S_i es la distancia promedio entre cada punto del grupo G_i y su centroide μ_i y S_j es la distancia promedio entre cada punto del grupo G_j y su centroide μ_j.

Cuanto más bajo sea el valor de este índice, ello indicará grupos más compactos cuyos centroides están bien separados los unos de los otros.

La penúltima fila de la tabla 8 muestra el valor obtenido para los ejemplos mostrados en las figuras 3 y 4. El resultado es el mismo que cuando se han usado los otros índices.

3.1.5 Coeficiente de Silhouette

El coeficiente de Silhouette tampoco está directamente relacionado con los índices SSW y SSB y se calcula para cada muestra del conjunto de datos. Proporciona un valor entre –1.0 y 1.0 que indica lo bien o mal que está agrupado dicho punto en su grupo. Los valores cercanos a 1.0 indican que la muestra está en el grupo correcto. Los valores cercanos a –1.0 indican que la muestra está en el grupo incorrecto.

Esta medida se calcula para cada punto x_i del conjunto de datos, como se muestra a continuación:

$$s(x_i) = \frac{b(x_i) - a(x_i)}{max(a(x_i), b(x_i))} \tag{14}$$

donde $a(x_i)$ es la distancia promedio de una muestra a todas las de su mismo grupo y $b(x_i)$ es la distancia mínima de la muestra a cualquier otra muestra del resto de grupos. $a(x_i)$ es una medida de la cohesión, mientras que $b(x_i)$ es una medida de la separación.

Una medida de la calidad del agrupamiento puede obtenerse calculando el promedio de los coeficientes de Silhouette de todas las muestras del conjunto de datos (SC), como se muestra en la siguiente ecuación:

$$SC = \frac{1}{N} \sum_{i=1}^{N} s(x_i) \tag{15}$$

La última fila de la tabla 8 muestra el valor obtenido para los ejemplos mostrados en las figuras 3 y 4. De nuevo, el resultado coincide con el alcanzado cuando se han usado los otros índices.

3.2 Técnicas de validación externa

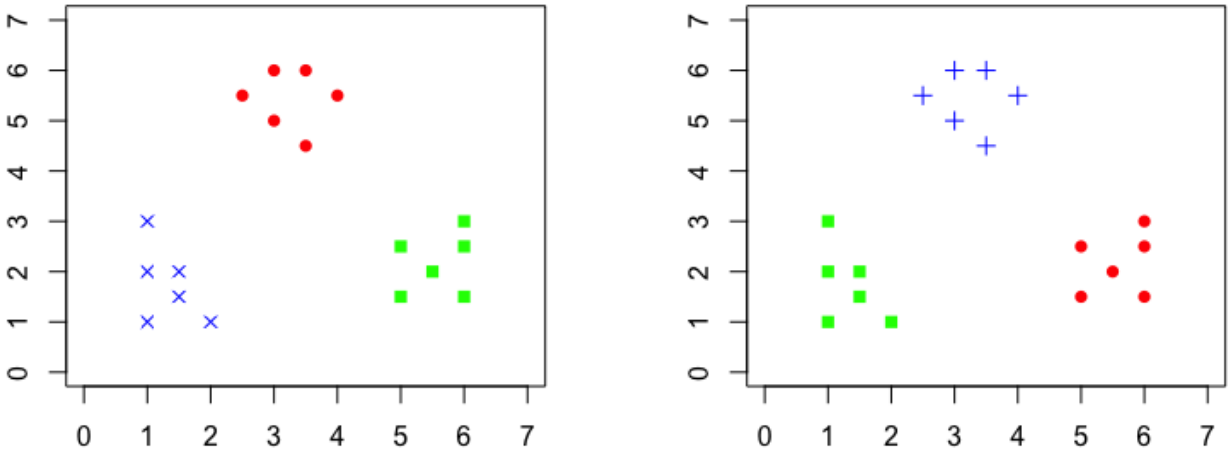
Cuando el objetivo es validar la calidad de un algoritmo de agrupamiento o comparar el funcionamiento de varios algoritmos, podemos usar, además de las técnicas de validación interna descritas, un conjunto de datos etiquetados previamente por un experto. En primer lugar, se procederá a ejecutar el algoritmo de agrupamiento usando los datos de entrada pero excluyendo la variable que etiqueta las muestras. Una vez que el algoritmo ha obtenido una partición de las muestras en los diferentes grupos, se procederá a comparar si el grupo en el que ha sido clasificada cada muestra coincide con la etiqueta original.

Una posible medida de la calidad del agrupamiento es la exactitud, definida como el total de muestras bien clasificadas dividido por el número total de muestras. También podrían usarse otras medidas, como las comentadas en el apartado 1.

Es importante tener en cuenta que el método de agrupamiento proporciona una agrupación de las muestras en los diferentes grupos, pero no identifica la clase a la que pertenecen. La figura 5 presenta este problema. Imaginemos un problema de clasificación de imágenes de animales. Las muestras originales están etiquetadas como *Gatos*, *Perros* o *Vacas*. En la figura 5a aparecen las muestras de la base de datos usando aspas azules para representar a los gatos,

círculos rojos para representar a los perros y cuadrados verdes para representar a las vacas.

Figura 5. A la izquierda, conjunto de datos original (**a**). A la derecha, un posible resultado de un algoritmo de agrupamiento (**b**).



La figura 5b muestra un posible resultado de la aplicación de un algoritmo de agrupamiento. El algoritmo de agrupamiento agrupa las muestras en tres grupos, y con el fin de informar al usuario del grupo al que pertenece cada clase, asigna a las muestras un número entre 1 y el número de grupos (en este caso, 3). Este identificador del grupo no proporciona ninguna información de carácter semántico. Es simplemente un identificador para poder diferenciar a qué grupo pertenece cada muestra. Para representar el resultado, se han usado los cuadrados verdes para mostrar las muestras del primer grupo, los círculos rojos para el segundo y las aspas azules para el tercero.

Antes de poder comprobar si el agrupamiento obtenido ha clasificado las muestras de forma correcta, debemos interpretar los grupos obtenidos. Es decir, debemos llegar a la conclusión de que el grupo 1 (cuadrados verdes en la figura 5b) corresponde a los perros (aspas azules en la figura 5a), que el grupo 2 (aspas azules en la figura 5b) corresponde a los gatos (círculos rojos en la figura 5a) y que, finalmente, el grupo 3 (círculos rojos en la figura 5b) corresponde a las vacas (cuadrados verdes en la figura 5a). En este ejemplo, es relativamente sencillo al tener solo dos dimensiones, pero en otros problemas con más dimensiones este proceso puede ser muy complejo.

Bibliografía

Ball, G.; Hall, D. (1965). *ISODATA. A novel method of data analysis and pattern classification*. Stanford Research Institute.

Caliński, T.; Harabasz, J. (1974). «A dendrite method for cluster analysis». *Communications in Statistics* (vol. 3, núm. 1, págs. 1-27).
<https://www.tandfonline.com/doi/abs/10.1080/03610927408827101>

Hartigan, J. A. (1975). *Clustering algorithms*. Nueva York: Wiley.

Xu, L. (1997). «Bayesian Ying–Yang machine, clustering and number of clusters». *Pattern Recognition Letters* (vol. 18, núm. 11, págs. 1167-1178).
<http://www.sciencedirect.com/science/article/pii/S0167865597001219>

Zhao, Q.; Fränti, P. (2014). «WB-index: A sum-of-squares based index for cluster validity». *Data and Knowledge Engineering* (vol. 92, págs. 77-89).
<http://www.sciencedirect.com/science/article/pii/S0169023X14000676>.

Zhao, Q.; Xu, M.; Fränti, P. (2009). «Sum-of-Squares Based Cluster Validity Index and Significance Analysis». En: M. Kolehmainen; P. Toivanen; B. Beliczynski. *Adaptive and Natural Computing Algorithms* (págs. 313-322). Berlín / Heidelberg: Springer.