

Balanceado en bases de datos

Pablo.Bermejo@uclm.es

Resumen

- Introducción
- Balanceado en clases binarias y clases con múltiples valores.
- Métricas
- Soluciones
- Conclusiones

Introducción

- Muchas bases de datos utilizadas para experimentación y entornos educativos no son realistas ya que, para no sesgar el modelo aprendido por un clasificador, los corpus se generan con el mismo número de muestras (registros, documentos, instancias) por clase:
 - MiniNewsGroup: clase de 20 valores, 100 documentos por cada uno.
 - Weka, Iris corpus: clase de 3 valores, 50 documentos por cada uno.
- Cuando se quieren aplicar tareas de Minería de Datos a un problema real, el corpus que representa dicha realidad puede presentar un reparto no equitativo en el número de muestras para cada clase.
- Cuando un corpus presenta este desequilibrio se dice que no está **balanceado**.

Introducción

- Un caso muy común son las bases de datos médicas donde para cada muestra (paciente) la clase puede ser '+' (enfermo) o '-' (sano). La mayoría de las muestras serán negativas, y esto hará difícil que nuestro clasificador aprenda un modelo correcto para predecir la enfermedad.
 - Por falta de muestras positivas
 - Por sesgo en el modelo
- El coste de clasificar una muestra positiva como negativa puede ser mucho mayor que al contrario.

Balanceado en clases binarias y clases con múltiples valores

Normalmente las clases **binarias** tiene uno de estos dos tipos de semántica:

1. Posesión de una propiedad (clase **positiva**) y falta de dicha propiedad (clase **negativa**).
 - La clase positiva suele ser poco frecuente y es difícil/caro conseguir muestras pertenecientes a dicha clase.
 - Aprender un modelo de la clase negativa no tiene sentido (las muestras no tienen relación entre sí) y basta con aprenderlo con la positiva.
 - Algunas veces el fallo en detectar una clase es más caro que en la otra:
 - Detección de cáncer.
 - Correo spam.
 - Cualquier tarea de Recuperación de la Información (ej: búsqueda de imágenes relevantes al término “agua”).

Balanceado en clases binarias y clases con múltiples valores

2. Pertenencia a dos grupos diferentes. Las muestras pueden pertenecer a la clase A o B
 - Se suele poder obtener con la misma facilidad muestras pertenecientes a ambos grupos.
 - Conviene aprender un modelo para cada clase.
 - A priori, el fallo en la detección de la clase correcta tiene el mismo coste.
 - Distinción automática de caracteres dibujados a mano ('u' o 'v').
 - Reconocimiento facial (hombre o mujer, adulto o niño,...)
 - La clase puede haber sido forzada a ser binaria y por lo tanto cada grupo puede esconder **disyunciones** (sub-conceptos). Esto hace más difícil aprender un modelo correcto para cada clase.
 - Distinción de imágenes de vehículos: reducir los grupos “Turismo”, “Furgoneta”, “Motocicleta” y “Bicicleta” en la clase binaria {“Cuatro Ruedas”, “Dos Ruedas”.}

Balanceado en clases binarias y clases con múltiples valores

- El valor de la clase binaria con menos muestras se conoce como clase **minoritaria**, y el otro clase **mayoritaria**.
- El grado de equilibrio en las clases binarias se suele expresar con el **ratio** **(1:M)**, indicando el número de muestras de la clase mayoritaria existentes por cada muestra de la clase minoritaria.

Balanceado en clases binarias y clases con múltiples valores

- Las clases binarias son un caso particular de clases **multinomiales**, las cuales pueden presentar una cardinalidad ≥ 2 .
- Cuanto mayor es la cardinalidad de una clase multinomial, más artificial resulta encontrar una base de datos equilibrada y procedente de un caso real.
- Un caso muy común es el de la categorización de texto, donde el número de posibles valores de la clase puede llegar a ser muy alto (género literario, clasificación de correo-e en carpetas,...).
- Cuanto mayor es la cardinalidad menos probabilidad hay de que una clase contenga disyunciones.
- Cuanto mayor es la cardinalidad, se encontrarán más casos en que una clase contiene muestras en el mismo espacio de dimensiones de otra clase (**overmapping**), pero puede ocurrir con cualquier cardinalidad.

Balanceado en clases binarias y clases con múltiples valores

- El grado de desequilibrio en clases multinomiales con cardinalidad >2 se suele expresar de dos formas diferentes:
 - Par (base:peak) , donde *base* es la cardinalidad de la clase minoritaria y *peak* es la cardinalidad de la mayoritaria. Sin embargo esta descripción no es muy informativa, ya que no se tiene información del resto de cardinalidades.
 - Par (μ, σ) , donde μ indica la media de las cardinalidades de todos los posibles valores de la clase, y σ la desviación típica de la media.

Métricas

Cuando se realiza clasificación supervisada en bases de datos muy poco balanceadas, es necesario tener cuidado con la métrica utilizada en la evaluación:

- Accuracy
- Precision
- Recall
- F_n-measure
- Curvas ROC

Métricas

Accuracy

- Es el porcentaje de clasificaciones correctas, teniendo en cuenta todas las clases.
- No recomendable para bases de datos muy desbalanceadas.
- Supongamos que nuestro test set contiene 950 instancias negativas y 50 positivas. Si el clasificador asigna todas las instancias a la clase negativa, tendría un 95% de accuracy. Sin embargo, lo que más nos interesa es que se clasifiquen correctamente las instancias positivas.

$$Accuracy = \frac{\#Clasificaciones\ Correctas}{\#Clasificaciones}$$

Métricas

Precisión

- Es el porcentaje de aciertos para una clase determinada.
- Es muy útil cuando la clase es muy desbalanceada y la clase de interés es la minoritaria.
- Normalmente, si modificamos nuestro clasificador para aumentar la precisión de una clase, el porcentaje de aciertos aumentará a costa de reducir el número de muestras identificadas para dicha clase (recall).
- En un sistema de Recuperación de la Información, cuanto mayor es la precisión de documentos relevantes, menos documentos se encontrarán en el ranking pero habrá mas certeza de que son relevantes.

$$Precision(c) = \frac{\#Instancias\ Con\ Clase\ c\ Clasificas\ Con\ Clase\ c}{\#Instancias\ Clasificadas\ Con\ Clase\ c}$$

Métricas

Recall

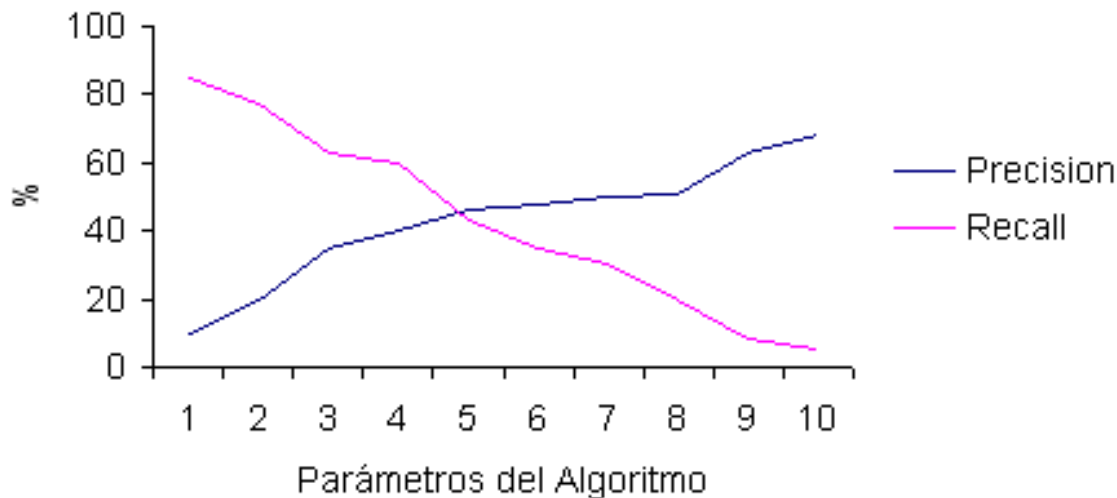
- Dada una cantidad N de documentos pertenecientes a una clase determinada c, Recall(c) es el porcentaje de instancias en N identificadas como pertenecientes a la clase c.
- Es útil para controlar que, por aumentar la precisión, estemos identificando muy pocas instancias.
- Al trabajar con clases binarias, también se le conoce como “sensibilidad”.
- En un sistema de Recuperación de la Información, cuanto mayor es el Recall de los documentos relevantes más resultados mostrará el ranking.
- También se conoce como *True Positive rate(c)*.

$$\text{Recall}(c) = \frac{\# \text{Instancias Con Clase } c \text{ Clasificas Con Clase } c}{\# \text{Instancias Con Clase } c}$$

Métricas

- Si asignamos todas las muestras a una misma clase c , $\text{Recall}(c)$ será 100% pero $\text{Precision}(c)$ será muy baja. Conviene presentar conjuntamente los valores para Precision y Recall.
- Las curvas en gráficas $\text{Precision}(c)$ Vs. $\text{Recall}(c)$ suelen cortarse en un punto llamado **breakeven point**, siendo este el punto de configuración de nuestro algoritmo en que las dos métricas se igualan.

Precision Vs. Recall



Métricas

F n -measure

- Otra forma de presentar los resultados *Precision Vs. Recall* es la métrica F n -measure, la cual computa la relación entre ambas.
- El parámetro n indica la importancia de recall sobre la precisión.
 - $n=1$ indica misma importancia. Se dice que F1-measure es la **media armónica** de la precisión y el recall; también se conoce como F-measure ó F1-Score.
 - $n=2$ indica que F-measure se calcule dando a recall el doble de peso que a la precisión.
 - $n=0.5$ indica que F-measure se calcule dando a la precisión el doble de peso que al recall.

$$F_n - measure(c) = \frac{(1+n^2) \times Precision(c) \times Recall(c)}{(n^2 \times Precision(c)) + Recall(c)}$$

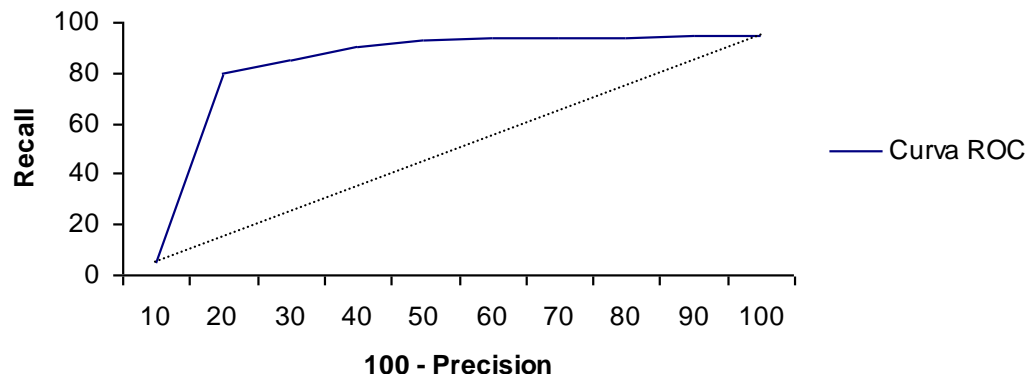
$$F1 - measure(c) = \frac{2 \times Precision(c) \times Recall(c)}{Precision(c) + Recall(c)}$$

Métricas

Curvas ROC

- Junto a la precisión el recall, las curvas ROC también son recomendables para bases de datos muy desbalanceadas.
- En el eje de abcisas se indica la False Positive Rate(c) ($100 - \text{Precision}(c)$), y en el eje de coordenadas la True Positive Rate (c) ($\text{Recall}(c)$) para el correspondiente punto en el eje de abcisas.
- Cuanto más cerca está la curva de la esquina superior izquierda, mejor es la clasificación.
- La herramienta Weka puede dibujar curvas ROC

Curva ROC para la clase c



Métricas

AUC

- AUC son las siglas de *Area Under the Curve*, refiriéndose a las curvas ROC.
- La curva tiende naturalmente al punto (100,100). Si este punto no existe hay que añadirlo para calcular el AUC. AUC es el área encerrada entre los dos puntos de la curva y el segmento que los une.
- Si seleccionamos aleatoriamente una instancia con clase c y otra con clase c' , $AUC(c)$ indica la probabilidad de que la primera instancia se asigne a la clase c antes que la segunda instancia.

Soluciones

Se pueden tomar 2 tipos de acciones ante un corpus desbalanceado:

1. Soluciones a nivel de algoritmos
2. Soluciones a nivel de datos

1. Las **soluciones algorítmicas** consisten en diseñar o modificar clasificadores para combatir la pobreza del modelo aprendido para las clases minoritarias. Normalmente los clasificadores que implementan estas soluciones se conocen como *meta-clasificadores*.
 - Dar pesos desiguales a cada clase en la hora del aprendizaje (clasificador AdaBoost).
 - Dar distinta importancia al fallo de reconocer clases (clasificador CostSensitiveClassifier).
 - Utilizar combinaciones de clasificadores a los que el desbalanceado les afecta de forma distinta (clasificador Vote)

Soluciones

2. Las **soluciones a nivel de datos** realizan modificaciones en el conjunto de entrenamiento para mejorar el modelo aprendido.
- *Modificar atributos*: selección o construcción de atributos. Por ejemplo, la selección por InfoGain suele mejorar la precisión a costa del recall.
 - *Modificar a nivel de instancias* muestreando un nuevo training set.
 - Se pueden generar nuevas instancias: *over-sample*, o eliminar instancias existentes: *under-sample*.
 - Si se realiza over-sample, se puede realizar con *reemplazo* o *sin reemplazo* de muestras en el conjunto original.
 - Over-sampling y under-sampling pueden realizarse de modo *dirigido* (inteligente) o aleatorio.
 - La combinación óptima de over-sample, under-sample, reemplazo, dirigido y aleatoriedad depende de la base de datos con la que se trabaja.

Soluciones

- SMOTE: realiza over-sample dirigido de la clase minoritaria sin reemplazo, y under-sample aleatorio de la mayoritaria. Las muestras creadas son instancias obligadas a estar en el espacio entre una instancia de la clase minoritaria y su vecino más cercano de la misma clase.

Conclusiones

- Las bases de datos procedentes del mundo real pueden estar muy desbalanceadas.
- Estas bases de datos presentan varios problemas:
 - Disyunciones.
 - Overmapping.
 - Pocas muestras para aprender un modelo de la clase de interés.
- Accuracy no es una métrica apropiada para estos casos
 - Precision
 - Recall
 - F_n-measure
 - ROC y AUC
- Soluciones a nivel algorítmico y a nivel de datos.