Trabajo de Minería de datos: Clasificación y Preprocesamiento

Javier Fumanal Idocin, Víctor ... y ...

4 de marzo de 2018

1. Preprocesamiento

1.1. Tratamiento de valores perdidos

Aunque el conjunto de datos no presenta muchos datos faltantes, sí que existen un número de valores faltantes suficientemente importantes como para tratarlos. La dimensionalidad del problema es demasiado alta como para comprobar si existe alguna relación entre los datos faltantes con alguna otra variable.

Es importante notar que los valores faltantes en los factores no se presentan como NA, sino que son leídos sin nombre, que hay que convertirlos previamente a NA antes de aplicar ningún método. Las siguientes opciones fueron evaluadas:

- Eliminación de observaciones con variables desconocidas.
- Imputación de variables con K-NN.
- Imputación de variables con MICE.

De los tres métodos probados, eliminar las observaciones no produjo ninguna mejor. Sí lo hacen imputar tanto con K-NN como con MICE, (ambos implementados ya en librerías de R), siendo este último el que ofrece mejores resultados. No obstante, es importante destacar que MICE tiene un costo en tiempo mucho mayor que la imputación de valores perdidos con K-NN.

En ambos casos de imputación, esta realiza conjuntamente en el conjunto de test y en el conjunto de train.

1.2. Normalización

En el caso de la normalización, se realiza de forma conjunta con el conjunto de test y el de train, luego de excluir las variables factores del conjunto de datos. La normalización a usar consiste en restar la media a cada variable y luego dividir por la desviación típica.

1.3. Filtro de Outliers

Para filtrar los datos atípicos o outliers se utiliza un filtro de comité. El proceso de aplicación del filtro es el siguiente:

1. Se genera k submuestras aleatorias del conjunto de datos original de tamaño n.
2. Se crea un clasificador para submuestra.
3. Se realiza una predicción de todo el conjunto de datos con cada clasificador.
4. Para cada observación, se decide si se etiqueta como ruido o no en función de si usa un criterio estricto (la mayoría de los clasificadores entrenados no pudieron clasificarlo correctamente) o suave (ninguno de los clasificadores pudieron clasificarlo correctamente).

En cuanto al algoritmo utilizado para entrenar los clasificadores, se han utilizado los 4 algoritmos propuestos por en el enunciado de la práctica, así como algunos adicionales. El algoritmo elegido finalmente es C5.0.

1.4. Balanceo de clases

Existen un número muy desigual de observaciones para cada clase en el conjunto de datos. Existen varias alternativas para solucionar el problema de la clasficación desbalanceada:

- Oversampling: se duplican las instancias de las clases con menos observaciones hasta que el conjunto de datos tenga un número de observaciones para cada una que parezca aceptable.
- SMOTE: se utiliza SMOTE para crear nuevas instancias de las clases minoritarias en función de las ya existentes.
- ROSE: ROSE crea nuevas instancias de la clase minoritaria. No obstante, ROSE sólo funciona en problemas clasificación binaria, y es necesario adaptar el problema actual, con 4 etiquetas, para que funcione.

Los resultados obtenidos muestran que SMOTE ofrece mejores resultados que los otros métodos aplicado una vez se han eliminado valores atípicos y desconocidos del conjunto de datos.

2. Clasificadores

2.1. RPART

Para aplicar el RPART primero se realiza una imputación de valores perdidos con MICE, a continuación se elimina ruido con el filtro de outliers y luego se utiliza SMOTE para balancear las clases (salvo que se piense utilizar posteriormente si se utiliza OVO).

En el caso de algoritmo RPART se ha utilizado la aproximación OVA de dos formas diferentes.

En ambos casos, se ha entrenado un modelo RPART para cada etiqueta del conjunto de datos. Únicamente aplicando el esquema de forma sencilla (escogiendo aquel que devuelva mayor valor) ya se obtiene una mejora respecto del RPART normal.

La primera aproximación implementada consiste en aplicar OVA en forma de árbol. Esto es, primero se clasifica con el modelo OVA que mayor % de acierto tiene. Si este devuelve positivo, se devuelve automáticamente la clase detectada. En caso contrario, se prueba con el siguiente con más % de acierto, y así sucesivamente. Si no se detecta positivamente ninguna clase, se utilizar un clasificador RPART entrenado para las cuatro clases.

En la segunda aproximación, se combina el esquema OVA con el esquema ONE-VS-ONE. Primero se calcula la respuesta de todos los clasificadores OVA para cada muestra, y a continuación se utiliza un clasificador OVO para diferenciar entre las dos clases de respuesta mayoritaria (es posible aplicar SMOTE o ROSE en algunos de estos modelos OVO de forma individual).

Es la segunda aproximación la que en principio ofrece un mayor % de acierto tanto en los datos de entrenamiento como de test.

2.2. GLM

Para aplicar el GLM primero se realiza una imputación de valores perdidos con MICE, a continuación se elimina ruido con el filtro de outliers y luego se utiliza SMOTE para balancear las clases.

En el caso de de GLM, al no soportar directamente problemas multiclase, es necesario descomponer el problema actual en un esquema ONE-VS-ALL, de modo que se entrena un clasificador para cada clase distinta en el conjunto de datos.

Para entrenar cada clasificador, es posible utilizar SMOTE en vez de en el conjunto de datos inicial en el conjunto de entrenamiento de cada clasificador en particular. Es posible también utilizar ROSE en este caso ya que el clasificador para cada clase resuelve un problema de clasificación binario. No obstante, esta aproximación no parece mejorar en tasa de acierto a la original, y la aplicación de ROSE lejos de mejorar los resultados finales, los empeora.

Además de aplicar OVA, se utiliza también One-Versus-One, de manera que se tiene un clasificador por cada par de etiquetas distintas. Para saber que clasificador OVO utilizar, se utiliza primero el clasificador OVA sobre una muestra. Las dos clases que más probabilidad devuelvan de ser ciertas indican qué modelo OVO se utilizará para clasificar la muestra.

Para entrenar cada modelo OVO se hace una selección de variables específica para cada uno, usando una metodología Top-Down para la selección de las mismas. Esto ahorra tiempo de entrenamiento para los modelos OVO, y evita problemas de separabilidad entre clases.

Esta aproximación tiene el problema de que la tasa de acierto entre modelos OVO es muy distinta. Por ejemplo, el modelo OVO que distingue entre clases 0 y 1 tiene una

tasa de acierto cercana al 90 %, mientras que el modelo que distingue entre 2 y 3 está en torno al 55 % de acierto.

2.3. 1-NN

1-NN que es una variante del famoso método KNN (K Nearest Neighbours) en el que se tiene en cuenta únicamente el vecino más cercano para realizar la predicción. El método funciona, básicamente, calculando las distancias del punto a clasificar con todos los puntos de entrenamiento y devolviendo como resutlado aquel que esté más cercano.

Lo primero es decidir la función de distancia. Es decir, la función que decidirá si un punto está más cerca o más lejos de otros. Hay varios tipos de distancias; Euclídea, Mahalanobis, Manhattan, etc. En este caso, se va a utilizar la distancia de Mahalanobis, que es la que se utiliza por defecto en la función *knn.cv*.

Empezaremos primero por diferenciar entre variables numéricas y no numéricas. En este caso, los factores se reconvierten en variables dummy, donde cada factor se convierte en tantas variables como niveles tenga, siendo estas nuevas variables 0 ó 1, indicando qué valor tenía el facotr original.

Esto se hace debido a la naturaleza de KNN y de cómo el clasificador se basa en las distancias entre las observaciones para clasificarlas. Está claro que es fácil calcular distancias entre variables numéricas, usando la distnacia Euclídea, pero en el caso de las variables categóricas sólo podemos hacerlo de esta manera, ya que nos falta información de orden en cada factor. (Por ejemplo, si tenemos valores 'Alto', 'Medio' y 'Bajo', bajo está más cerca de medio que de alto), Como en nuestro nuestro caso no disponemos de esa información, se recurre a la solución mentada.

Como es necesario utilizar normalización en 1-NN para evitar problemas de distintas escalas entre variables, se utiliza la normalización. La normalización escogida es la z-score, ya que permite dejar a los outlier como tales (algo en principio deseable por nosotros).

Ningún tratamiento especial se hace para este método en los outliers. La imputación de valores se realiza con MICE.

3. Reparto del trabajo

- Javier Fumanal Idocin (Kaggle: fuminides): Imputación con k-nn y MICE, normalización, filtro de comité de outliers, oversampling, SMOTE y ROSE. RPART y GLM.
- Victor 1NN.