Laboratiorio 5
Minería de datos
SMOTE

Andoni Martín Reboredo

7 de diciembre de 2014

1. Introducción

El algoritmo SMOTE otorga una manera de crear instancias artificiales de una clase estudiando los k vecinos más próximos y creando un número arbitrario de instancias aleatorias. Típicamente, este algoritmo se aplica con un factor de sobredimensionamiento inferior al que se ha aplicado para el estudio aquí realizado, sin embargo, vemos como estos factores tienen sentido sobre los archivos de la práctica con los datos obtenidos en este estudio.

2. Justificación

El problema que planteaba esta práctica partía de un archivo de datos desbalanceado, típicamente estamos acostumbrados a construir nuestros clasificadores sobre archivos con instancias balanceadas. Los métodos que usamos para evaluar el rendimiento son además inválidos ya que los estudios anteriores se basaban en la F-measure para comparar unos métodos con otros, en este caso, prima saber cuantos casos de la clase positiva han sido clasificados correctamente siendo esta una cota válida de evaluación.

Para entrenar los modelos se ve necesaria por tanto una forma de ampliar la clase minoritaria o de reducir la clase mayoritaria. En nuestro caso, Iñigo ha asumido la implementación del módulo Resample, que aborda este problema desde un punto de vista diferente al ofertado por el algoritmo SMOTE.

2.1. Codificación y puesta en funcionamiento

El programa creado para esta práctica hace uso de la implementación que viene ya realizada en la librería de weka haciendo solamente un recubrimiento sobre el código proporcionado.

Para reducir los tiempos de codificación, el programa no es tolerante a fallos ni a ejecuciones que violen las precondiciones de ejecución.

2.1.1. Precondiciones y postcondiciones

Recibe tres parámetros, el archivo sobre el que se quiere aplicar el algoritmo, en formato arff válido, el factor de sobredimensionamiento y el número de vecinos que se quiere estudiar para lo creación de nuevas instancias.

Devuelve un archivo de nombre similar al obtenido añadiendo SMOTED al final para indicar que se ha aplicado el algoritmo.

2.2. Pruebas realizadas sobre la codificación

El algoritmo ha sido probado sobre todos los archivos contenidos en la carpeta de muestra de weka aún sin que estos archivos requirieran ningún balanceo a fin de garantizar que funcionara correctamente en un entorno variado.

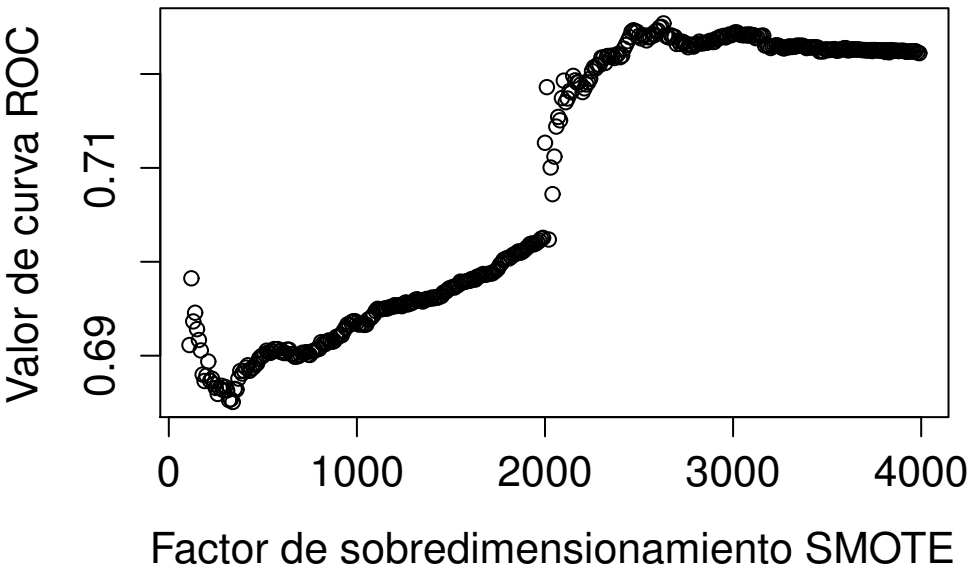
El resultado de la aplicación ha sido estudiado abriendo los archivos en la aplicación weka y observando que las instancias habían variado de número y haciendo un análisis visual comparativo de la disposición de las variables artificiales sobre el espacio muestral y sobre las variables no artificiales.

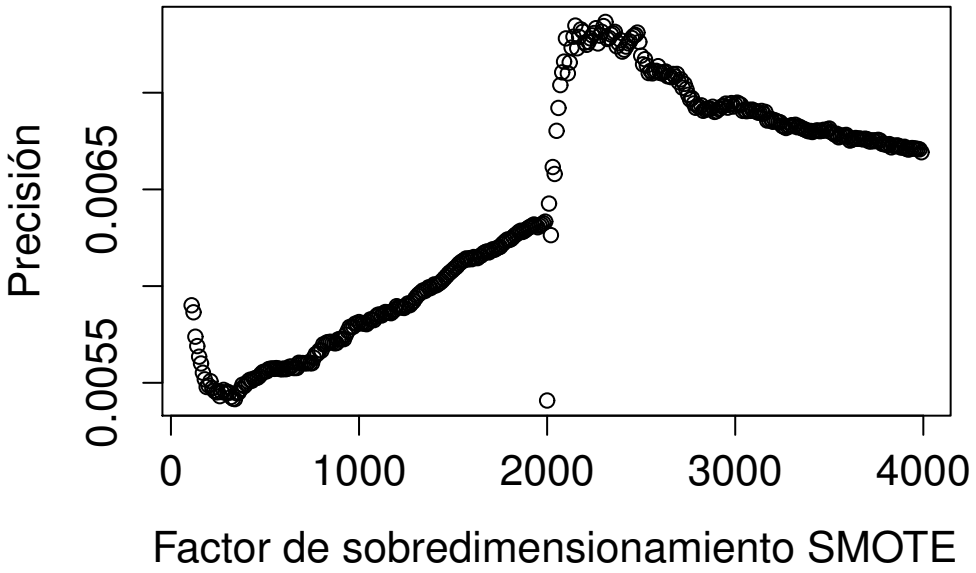
3. Análisis de combinación con resample

Para aplicar el algoritmo a la práctica que nos concierne he creado un entorno de prueba que sobre el archivo original de entrenamiento aplica los filtros SMOTE primero y resample después variando el factor de sobredimensionamiento evaluando el rendimiento sobre el archivo de test mediante un evaluador arboreo tipo J48 entrenado en cada iteración.

Los resultados obtenidos son de carácter empírico y son aplicables solo al modelo evaluado, sin embargo si nos sirve para que nos podamos hacer una idea de como varía el rendimiento de un modelo cuando se aplican filtros generadores de instancias. El factor de sobredimensionamiento estudiado va desde 100 hasta 4000 en intervalos de 10 porciento en diez porciento.

Para estudiar el rendimiento, he extraido dos indicadores, la curva ROC y la precisión sobre la clase minoritaria.





Ambas figuras representan valores altamente correlacionados [1] y se observa una tendencia ascendente continua hasta el factor de sobredimensionamiento 2000, en este punto, ambos indicadores experimentan un ascenso más pronunciado llegando a alcanzar un valor de curva de ROC de 0.7253 y una precisión de 0.0073.

Cuando el algoritmo comienza a redimensionar las muestras por encima de 2500 esta tendencia se invierte demostrándose continuamente negativa hasta la cota de finalización de este estudio situada en un 4000 %

La diferencia de precisión entre el valor más alto obtenido y el más bajo es de un 2 % siendo esta cantidad despreciable en algunas aplicaciones pero bastante significativa en la aplicación actual si tenemos en cuenta que el estudio se ha realizado sobre un árbol que no está optimizado para el conjunto de datos de prueba.

4. Conclusión

En vista a las posibilidades que ofrece este algoritmo para mejorar el resultado del entrenamiento de distintos modelos de clasificación y de la posibilidad de aplicar otras técnicas, tanto de sobremuestreo como de eliminación de instancias, conjuntamente con este algoritmo puedo concluir que en ciertas aplicaciones concretas el resultado se va a ver mejorado tras aplicar SMOTE.

[1] 94.94995 de correlación