Práctica 5

Minería de datos

Curso: 4º.

Fecha: 7/12/2014

Jonathan Castro

Índice de contenido

Parámetros de entrada....4
Modo de funcionamiento....4
Batería de pruebas....6
Problemas obtenidos....7
Bibliografía....8

Índice de ilustraciones

Ilustración 1: Cómo buscar outliers.....5
Ilustración 2: Resultados con todos los atributos salvo uno y la clase.....6

One-class

Repisotorio: https://github.com/spolex/datamining-modules

Parámetros de entrada

Lo que he realizado es un programa que, dependiendo del número de ficheros con conjuntos de instancias de entrada hace una cosa u otra:

- De introducir un único fichero, realizará la búsqueda de outliers y sacará como resultado un fichero con el conjunto de instancias sin outliers.
- De introducir dos, realizará la búsqueda del mejor parámetro un para entrenar un modelo clasificador y luego se evaluará, dando como resultado el fichero .model que contendrá el modelo óptimo y la tabla de resultados en un fichero de texto.
- De introducir tres, realizará lo miso que el segundo punto y, además, realizará la evaluación de un test para ver si clasifica bien o el modelo o no. Por lo tanto, a parte de los dos ficheros anteriores, se obtendrá, a su vez, un fichero .arff con lo que se supone que el clasificador a evaluado como clase positiva (outliers de la manera en el que he realizado el ejercicio).

Modo de funcionamiento

La manera en el que he realizado el ejercicio ha sido, en todo momento, buscar outliers, puesto que es una de las funciones por la que son usados los clasificadores One-class.

Outliers

Para la búsqueda de outliers, he tratado la clase binaria como si fuera una única, añadiendo un parámetro artificial que hará de clase durante el proceso. Se entrenará el modelo con el parámetro nu = 0.1 y se hará una evaluación no honesta con los mismos datos. Se considerarán outliers aquellas instancias que no se han logrado clasificar, como bien se puede ver en la Ilustración 1.

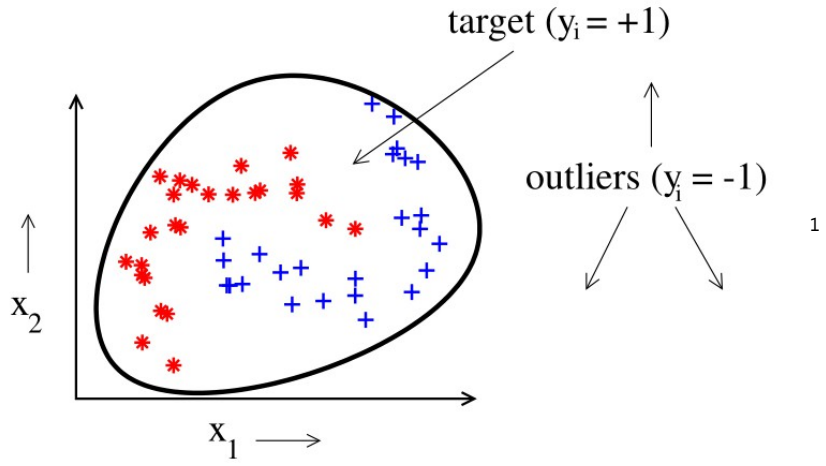

Ilustración 1: Cómo buscar outliers.

Sinceramente, no creo que sea la manera correcta de realizar la búsqueda de outliers, pero tampoco he encontrado en ningún sitio la manera de hacerlo o, de haberla encontrado, no la he entendido, puesto que la mayoría de textos que he encontrado acerca de One-class estaban en inglés y con terminología matemática.

Entrenar un modelo

Para entrenar el modelo, he optado por eliminar todo registro de instancias de clase positiva (minoritaria) porque con pocos datos no es posible entrenar bien el modelo. Se entrena el modelo únicamente con las instancias negativas y, una vez evaluando el modelo, las instancias con clase positiva deberían ser las que el clasificador no logre clasificar.

Para obtener la lista de instancias, para el conjunto de test no se elimina ninguna instancia, de ahí que las positivas tengan se acaben tratando como outliers.

El único parámetro que se recorre en busca de mejores resultados en nu. El parámetro del kernel RBF, C, la impresión que me dio era que distorsionaba mucho los resultados y los perjudicaba, por lo que en todo momento es considera como 0.0, tal y como hace Weka.

Este parámetro se recorre desde 0.001 hasta 0.1, en saltos de 0.001. Si no se logra mejorar el resultado tres veces seguidas, es decir, no logre realizar una cima secundaria, se sale del bucle y se ajusta el modelo con la mejor nu obtenida hasta el momento.

1 Imagen obtenida de la tesis de David Martinus Johannes realizada el 19 de junio de 2001.

Batería de pruebas

La batería de pruebas que he realizado es, la verdad, escasa. Este tipo de clasificador consume más memoria RAM de la que tengo disponible en el ordenador, por lo que no he podido comprobar los resultados con los datos intactos suministrados.

En cambio, he realizado pruebas con los mismos ficheros eliminando instancias y comprobando los resultados con Weka.

La verdad es que los datos son un tantos malos, por llamarlos de alguna manera, además de no poder comprobar la f-measure, puesto que siempre es 1. La manera en la que he comprobado todos los datos ha sido usando el número de instancias correctamente clasificadas y el número de no clasificadas.

Usando los ficheros con los atributos eliminados que me suministró mi compañero Alberto, train.e2e.wOOV.obfuscated.arffattSel.arff y dev.e2e.wOOV.obfuscated.arffattSel.arff, en los que se seleccionó un único atributo, a parte de la clase, con más información. Los resultado que fueron:

Correctly Classified Instances	23175	98.9835 %
Incorrectly Classified Instances	0	0 %
Kappa statistic	1	
Mean absolute error	0	
Root mean squared error	0	
Relative absolute error	NaN %	
Root relative squared error	NaN %	
UnClassified Instances	238	1.0165 %
Total Number of Instances	23413	

Ilustración 2: Resultados con todos los atributos salvo uno y la clase.

Problemas obtenidos

A parte del problema del consumo, que debe ser propio de LIBSVM, hay otro tipo de problemas que aparecen con algún fichero en concreto y saltan excepciones. Con algunos ficheros todo va bien, pero hay otros que da error a la hora de filtrar la clase y no llego a entender el problema completamente.

Bibliografía

- Problema con el parámetro nu y el overfitting: http://www.researchgate.net/post/Increasing_nu_parameter_in_One-Class_SVM_Im_using_LIBSVM_causes_underfitting_and_a_small_value_for_nu_causes_overfitting_Am_I_right
- Efecto de nu: http://stats.stackexchange.com/questions/72515/training-one-class-svm-using-libsvm
- Introducción a One-class: http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/
- Tesis de David Martinus Johannes: http://homepage.tudelft.nl/n9d04/thesis.pdf
- Detección de outliers con One-class: http://scikit-learn.org/stable/modules/outlier_detection.html
- Usos de One-class: http://www.academia.edu/165419/An_Evaluation_of_One-Class_Classification_Techniques_for_Speaker_Verification