Solemne I – 3era Parte: Minería de Datos y Procesamiento de Lenguaje Natural

Profesor: ***Alejandro Figueroa***

Ponderación: ***2***

Fecha de entrega: ***viernes 26 de septiembre.***

Ayudante: ***Nicolás Olivares***

Método de entrega: ***mail al ayudante (nicolivares@gmail.com)***

Objetivo

En la tarea anterior, el alumno exploró cuatro espacios vectoriales diferentes. En esta tarea exploraremos otras dimensiones del aprendizaje supervisado. Cambiaremos el algoritmo de aprendizaje, utilizaremos métodos de sampling, y combinaremos clasificadores. En la tarea anterior utilizamos SVM, ahora también utilizaremos MaxEnt y Bayes. La implementación de ambos algoritmos es provista por MALLET[1].

Oversampling vs. Undersampling

En clases aprendimos que muchas veces tener conjuntos desbalanceados de datos generan problemas, ya que las funciones discriminantes aprendidas pueden tener un sesgo hacia la clase mayoritaria. En la primera parte de esta tarea se le pide al alumno hacer undersampling de la clase token y calcular el accuracy, precisión, recall y F1-Score respectivos.

Posteriormente, realice oversampling de las tres clases de entidades, calculando las métricas mencionadas anteriormente realizando 10-fold cross-validation, y considerando la clase entidad como positiva. Nótese que ambos métodos de sampling deben hacerse sobre los conjuntos de entrenamiento E_i, dejando el respectivo S_i intacto (cíñase a las definiciones de ambos conjuntos provistas en la tarea anterior). Es conveniente conservar los 10 splits producidos para la tarea anterior, y focalizarse sólo en el mejor de los cuatro modelos obtenidos. Los samplings deben hacerse sólo los respectivos S_i.

¿Qué puede concluir de ambos experimentos? ¿Qué funciona mejor? ¿En qué sentido? ¿Cómo mejora la detección de entidades? ¿Más información ayuda siempre? ¿Cómo son estos resultados a la luz de los obtenidos en la tarea anterior (mejor modelo)? Muestre un gráfico con la curva ROC para los tres casos.

[1] http://mallet.cs.umass.edu

Se pide un experimento adicional: ajustar el parámetro de penalización "c" de las SVM. Considere su caso de undersampling, y genere modelos con distintos valores de "c". Calcule la performance de cada uno de los modelos y haga un gráfico "c" versus alguna de las métricas de performance, e.g., F-Score. Compare la curva ROC generada por undersampling sin y con el ajuste de "c". ¿Qué observa en el gráfico? ¿Cuál de los dos clasificadores tiende a ser más "conservador" o "liberal"?

Bagging

El segundo ítem consiste en hacer **Bootstrap Aggregating**. Haciendo holdout evaluation, divida su conjunto de datos en 66% para entrenamiento y 34% para prueba. Esta división hágala de manera aleatoria y déjela fija en lo que resta de esta parte.

Procedimiento: Del 66% de entrenamiento genere una muestra del mismo tamaño escogiendo vectores de manera aleatoria. En esta nueva muestra puede haber elementos escogidos múltiples veces. Con esta nueva muestra genere un modelo y calcule su desempeño sobre el 34% de prueba apartado anteriormente. Utilice el mejor "c" determinado en el punto anterior.

Repita el procedimiento, generando un nuevo modelo y calculando su desempeño. Combine los resultados de ambos modelos utilizando el voto de la mayoría. En caso de empates, opte por el que tenga mayor valor de confiabilidad. ¿Cómo son los resultados combinados versus ambos modelos individuales?

Repita el procedimiento hasta que incorpore 10 modelos generados por bagging, y muestre cómo evoluciona el resultado mediante la combinación sistemática de los modelos. Para esta parte como métrica de desempeño puede utilizar el F-Score de la clase entidad, y utilice los vectores del mejor modelo de la tarea anterior. Muestre la matriz de confusión de la mejor combinación de clasificadores.

Modelos de Aprendizaje

Considere solamente el mejor modelo obtenido en la tarea anterior. Siendo consistente, considere los 10 splits de aquel modelo y utilice Mallet para hacer 10-fold cross-validation con MaxEnt y Bayes. Para importar los datos de entrenamiento del formato SVM_LIGHT a MALLET, la línea de comandos está dada por:

```
bin/mallet import-svmlight --input file --output file.mallet
```

Para entrenar, se utiliza la siguiente directiva:

```
bin/mallet train-classifier --input file.mallet --output-classifier my.classifier --
trainer MaxEnt
```

En caso de Bayes, se puede omitir "--trainer MaxEnt". Para evaluar,

```
bin/mallet classify-file --input testdata --output resultados --classifier my.classifier
```

Una vez obtenidos los resultados, calcular la accuracy, precisión, recall, F-Score y curva ROC para estos dos nuevos clasificadores y contrastar con la obtenida por SVM en la tarea anterior. Dado sus resultados, fundamente sus conclusiones. Estudie y compare las tres matrices de confusión. ¿Hay concordancia en los tipos de errores? ¿Qué puede concluir de los errores? ¿Qué clasificador gestiona mejor qué clase?

¿Qué sucede si utiliza el voto de la mayoría de los tres clasificadores? ¿A qué se debe la mejora o detrimento en los resultados? ¿Cómo sería la matriz de confusión de este meta-clasificador?

Modelos de Clasificación Binaria vs. Multi-Clase

Hasta este punto se han utilizado modelos de clasificación binarios. Ahora, se debe utilizar los mismos vectores anteriores, los mismos splits, pero en esta ocasión las etiquetas deben representar las cuatro clases en vez de sólo -1's y +1's. Usando un procedimiento similar al de SVM Light (10 fold-cross-validation), se debe utilizar SVM Multiclass:

http://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html

Con los resultados obtenidos, mostrar la matriz de confusión, calcular la accuracy del modelo. Para cada clase, calcule el Recall, Precisión y F-Score. Haga un análisis de error. ¿Cuál es la fracción de clases de entidades que más se confunde? ¿Es mejor el modelo Multi-Class o el binario? ¿Existe algún patrón en los errores? ¿Cómo se podría mejorar?