DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico Número 1: Spam Filter

27 de Septiembre de 2016 — Aprendizaje Automático

Integrante	LU	Correo electrónico
Abdala, Leila Yasmín	950/12	abdalaleila@gmail.com
Costa, Manuel José Joaquín	035/14	manucos94@gmail.com
Bertero, Federico		federico.bertero@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
http://www.fcen.uba.ar

1. Extracción de atributos

Para esta primer etapa planteamos dos tipos de atributos, automáticos y no automáticos. Los atributos no automáticos son 4. HTML, indica si un mail contiene código HTML. SUBJ, indica si un mail contiene el campo subject. LEN, retorna la cantidad de caracteres del mail. Y finalmente SPACES, cuenta la cantidad de espacios " " que parecen en el mail.

Por otro lado, los atributos automáticos se crean haciendo uso de la librería SKLearn[I]. Con la misma, creamos a partir de la base de datos un vocabulario de palabras frecuentes, usando como métrica la cantidad de apariciones de cada una de ellas en ese mismo mail. Con el fin de eliminar las palabras repetidas en los headers y los conectores de lenguaje (por ejemplo The, He, Why, To, ect.), hicimos la suposición de que no es posible que una *keyword* aparezca en mas del 60 % de los mails[1].

Motivados por el fuerte impacto del código HTML en la extracción de atributos automáticos, nos planteamos implementar un procedimiento para preprocesar los datos de entrada. Finalmente, dado que el elevado tiempo de computo que requería esta tarea, que las herramientas que encontramos para quitar el código HTML presentaban errores en algunos casos y, la despreciable diferencia hallada entre el accuracy de los mails que fueron preprocesados y los que no, es que se determino omitir esta tarea.

2. Modelos

A continuación, listamos los clasificadores usados y los parámetros que probamos sobre cada uno de ellos. La selección de los mismos fue realizada con el objetivo de explorar lo mas posible espacio de búsqueda y esta basada en algunos papers y artículos[II] leidos.

1. Decision Tree: Los parámetros usados fueron max_depth con los valores: None, 3, 10, 100, 300. max_features con los valores: None, 1, 3, 10. max_leaf_nodes con los valores: None, 25, 50, 100, 1000. min_samples_split con los valores: 2, 3, 4, 10. Y criterion con los valores: "gini" y "entropy".

2. Gaussian Naive Bayes: Probamos sin variar los parámetros. (...) (tiene parametros??)

3. Multinomial Naive Bayes: alpha = 0.25 fit_prior = True

4. Bernoulli Naive Bayes: binarize = 0.0 alpha = 0.25 fit_prior = False

5. K Nearest Neighbors: Los parámetros usados fueron n_neighbors con los valores: 1, 3, 5, 7, 10. weights con los valores: 'distance' y 'uniform'. Y algorithm con los valores: 'ball_tree', 'kd_tree', 'brute'.

[1] Llegamos a este porcentaje mediante experimentación manual que no agregamos porque consideramos que no aporta al objetivo de este informe.

6. Support Vector Machines: Los parámetros se separaron en dos casos para este clasificador. Cuando definimos kernel con el valor 'linear' usamos C con los valores: 1, 10, 100, 1000. Para kernel con el valor 'rbf' definimos C con los valores: 1, 10, 100, 1000 y gamma con los valores: 0.001, 0.0001, 0.1, 0.2, 0.5, 1.0.

7. Random Forest: Los parámetros usados fueron max_depth con los valores: None, 3, 10, 100. max_features con los valores: None, 1, 3, 10, 100. min_samples_split con los valores: 1, 3, 10, 100. min_samples_leaf con los valores: 1, 3, 10, 100. bootstrap con los valores: True y False. Y criterion con los valores: "gini" y "entropy".

3. Reducción de dimensionalidad

3.1. Seleccion de atributos

3.2. Transformacion de atributos

4. Resultados

Para evaluar los métodos usamos como métrica la media armónica[2] con $\beta = 0,5$ ya que deseamos enfatizar la precisión. Esto es porque, al estar clasificando Spam, se considera un grave error que un mail Ham sea clasificado como Spam[3] mientras que clasificar un Spam como Ham es aceptable[4].

A continuación, listamos los parámetros para los cuales obtuvimos los mejores resultados por cada clasificador. Para todos los clasificadores se usan los mismos atributos, que aparecen descriptos en la sección 1. Cabe destacar que se considero otro set de atributos, pero dado que obtuvo resultados peores en todos los casos, fue descartado. Este consideraba como métrica la aparición de una palabra en el mail, usando para ello un vocabulario extraído de manera no automática[5] y clasificando como relevantes las λ palabras con mas apariciones en la base de datos de Spam.

Decision Tree

- max_features = None
- max_leaf_nodes = 100
- min_samples_split = 2
- criterion = gini

Obtuvimos $f_{0,5} = 0{,}992519162723$.

[2] $f_{0,5}$
[3] false positives
[4] true negative
[5] Llamamos no automático a todo proceso realizado íntegramente por nosotros, es decir, sin recurrir a librerías tales como SKLearn.

Gaussian Naive Bayes Obtuvimos $f_{0,5} = 0{,}564002009787$.

Multinomial Naive Bayes

- alpha = 0.25
- fit_prior = True

Obtuvimos $f_{0,5} = 0{,}624500833551$.

Bernoulli Naive Bayes

- binarize = 0.0
- alpha = 0.25
- fit_prior = False

Obtuvimos $f_{0,5} = 0{,}942918912295$.

K Nearest Neighbors

- n_neighbors = 1
- weights = uniform
- leaf_size = 15
- algorithm = kd_tree

Obtuvimos $f_{0,5} = 0{,}919134264754$.

Support Vector Machines

- kernel = rbf
- C = 1
- gamma = 1.0

Obtuvimos $f_{0,5} = 0{,}838130487534$.

Random Forest

- max_features = 0.5
- max_leaf_nodes = None
- min_samples_split = 4
- criterion = gini
- n_estimators = 20

Obtuvimos $f_{0,5} = 0{,}996898685035$.

5. Discusión

Bibliografía

[I]http://scikit-learn.org/stable/modules/feature_extraction.html

[II]L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and Regression Trees", Wadsworth, Belmont, CA, 1984.

L. Breiman, and A. Cutler, "Random Forests", http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

V. Metsis, I. Androutsopoulos and G. Paliouras (2006). Spam filtering with Naive Bayes – Which Naive Bayes? 3rd Conf. on Email and Anti-Spam (CEAS).

"Automatic Capacity Tuning of Very Large VCdimension Classifiers" I Guyon, B Boser, V Vapnik - Advances in neural information processing 1993.