Proyecto Final de Sistemas de Información
Tema: Clasificador de Anuncios de Revolico

Luis A. Díaz Borge C-511
Alejandro Ramírez Comezañas C-511
Rayniel Ramos González C-512
Frank E. Rosales González C-512

Universidad de La Habana
Facultad de Matemática y Computación
2019-2020

Resumen

Revolico es un portal de anuncios clasificados que se ha convertido en el lugar de referencia para la compra-venta en el país. El presente trabajo se plantea la implementación de un *software* que permita clasificar anuncios de Revolico de manera automática. Para ello fue construido un Dataset conformado por más de 31000 anuncios clasificados. Se realizó una investigación profunda sobre el estado del arte de clasificación de texto. Posteriormente fueron implementados cuatro de los modelos más populares para este tipo de problema: Árbol de Decisión, KNN, Naive-Bayes y SVM. Además se realizaron comparaciones entre los modelos para determinar cúal es más eficiente en este tipo específico de problema de aprendizaje supervisado.

Palabras Clave: Revolico, anuncio, clasificación de texto, aprendizaje supervisado.

1 Introducción

La clasificación de texto es el acto de dividir un conjunto de documentos de entrada en dos o más clases, donde cada documento pertenece a una o más clases [8]. El enorme crecimiento de los flujos de información y especialmente, el explosivo crecimiento de Internet promovió el desarrollo de la clasificación automática de texto. El desarrollo de la computación proporcionó suficiente poder de cómputo como para que la clasificación automática de texto fuera usada en diferentes aplicaciones prácticas. La clasificación de texto es comunmente usada para detectar *spam*, clasificar grandes colecciones de documentos según el tema del cual tratan, administrar conocimiento y ayudar a los motores de búsqueda de Internet [5]. La clasificación automática de texto siempre ha tenido importantes aplicaciones y ha sido tema de investigación desde la introducción de los documentos digitales. Hoy en día es necesaria debido a la enorme cantidad de documentos que son manejados diariamente [3].

Una de las aplicaciones de esta rama de la investigación es la clasificación automática de anuncios publicitarios. El presente trabajo estará centrado en la clasificación automática de anuncios del sitio web Revolico.

1.1 Problema

El problema consiste en lo siguiente: Dado un anuncio de Revolico, determinar cuál es su clasificación. Teniendo en cuenta la naturaleza del problema, queda claro que se trata de un caso particular de clasificación de texto, por lo que el problema será atacado con las herramientas expuestas en el estado del arte de clasificación de texto.

1.2 Estado del Arte

La clasificación de texto ha sido atacada desde muchos puntos de vista y a través de diferentes modelos de aprendizaje. He aquí un resumen de los métodos y modelos más mencionados en el estado del arte de esta familia de problemas:

- Modelo Secuencial [2]: Un modelo de aprendizaje que utiliza clasificadores que restringen secuencialmente el número de clases, mateniendo con alta probabilidad la presencia de resultados verdaderos en el conjunto de candidatos.

- Modelo basado en Máquinas de Soporte Vectorial (**SVM**) [4]: Este trabajo desarrolla un modelo de aprendizaje estadístico basado en máquinas de soporte vectorial (**SVM** por sus siglas en inglés). Este estudio conecta las propiedades estadísticas de las tareas de clasificación de texto con la generalización del rendimiento de **SVM** de manera cuantitativa. A diferencia de los enfoques tradicionales para el aprendizaje de clasificadores de texto, los cuales confían principalmente en evidencia empírica, este modelo explica cuándo y por qué **SVM** realiza un buen trabajo en clasificación de texto.

- Revisión de métodos para clasificación de texto [5]: Este artículo proporciona una revisión de los procesos genéricos de clasificación de texto. En el texto son expuestos ejemplos de clasificación de páginas web y detección de spam. Además, son descritos los principios teóricos de los cuatro principales motores de clasificación de texto: Naive-Bayes, k *Nearest Neighbours* (**KNN**), **SVM** y redes neuronales.

- Comparación entre los dos enfoques del modelo de Naive Bayes [6]: Este artículo proporciona una detallada comparación entre los dos principales enfoques probabilísticos basados en el modelo de Naive-Bayes: el enfoque que asume total independencia entre los términos y aquél que tiene como primicia, la existencia de cierta dependencia entre las palabras y usa multigramas como representación de los términos. En el artículo se esclarece la confusión, describiendo las diferencias y detalles de estos dos modelos.

Teniendo encuenta los modelos mayormente utilizados en el estado del arte de clasificación de texto, para resolver el problema planteado en la sección 1.1 serán utilizados los modelos: **KNN**, **SVM**, Naive-Bayes y Árbol de Decisión.

1.3 Objetivos

El objetivo general de este trabajo es la implementación de un *software* que permita clasificar el texto de un anuncio del sitio web Revolico, asignándole la categoría a la cual pertenece.

Como objetivos específicos, se encuentran:

1. Descargar de manera manual o a través de un *crawler* los anuncios Revolico, para utilizarlos como Dataset.
2. Procesar el texto contenido en los archivos HTML descargados, para formar el Dataset.
3. Implementar varios modelos de aprendizaje supervisado, que sean entrenados con el Dataset y sean capaces de predecir con cireto nivel de precisión la clase a la que pertenece algún anuncio.
4. Realizar una comparación entre dichos modelos y resaltar cúal es más conveniente usar en este tipo de problemas.
5. Implementar una interfaz gráfica de usuario que permita seleccionar un modelo y predecir la categoría de un anuncio cualquiera.

2 Contrucción del Dataset

Para la contrucción del Dataset no fue necesaria la implementación de un crawler ya que los anuncios fueron extraidos de una imagen ISO del servidor de Revolico. Los archivos HTML fueron separados en cada una de las 41 subcategorías de anuncios. El texto de los archivos HTML fue extraido y procesado, pasando por un proceso de limpieza que consistió en:

- Eliminación de símbolos y números.
- Sustitución de las vocales con tilde por su homóloga sin tilde.
- Normalización de las palabras con letras repetidas usando expresiones regulares.
- Tokenización

- Eliminación de *Stopwords*
- Lamatización

Una vez "limpio.[el] texto, los tokens fueron guardados en archivos JSON que además contienen la categoría a la que pertenece cada anuncio. En total, el Dataset cuenta con 31934 anuncios clasificados en 41 categorías diferentes. El 20 % de los anuncios de cada categoría se separó aleatoriamente para formar el *Test Set*. Posteriormente los diccionarios contenidos en los archivos JSON fueron serializados en dos listas para tener el dataset en un solo archivo y por tanto facilitar su carga en memoria.

3 Implementación de los Modelos

Los modelos seleccionados en la sección 1.2 fueron implementados usando las bibliotecas de Python *scikit-learn* y *keras* (este último para el caso del modelo de redes neuronales).

Para el caso **KNN** no se usa *cross validation* ya que la precisión del modelo depende de la cantidad de vectores vecinos (K), por tanto, al disminuir el tamaño del Training Set, se pierde preseción. En el resto de los modelos implementados sí se hace *cross validation*.
En el caso de **SVM**: Se usa un optimizador para el hiper-parámetro del modelo llamado *RandomizedSearchCV*, el cual apoyado en la distribución del parámetro, selecciona una cantidad determinada de valores y realiza *cross validation* para cada uno de ellos, calcula el promedio de estos valores y finalmente selecciona el valor del hiper-parámetro cuyo promedio de *cross validation* sea mejor.

3.1 Representación de los Documentos

Según lo expuesto en [3], [5] y [1] existen varias formas de representación de los documentos, entre ellas: representación binaria; el vector de frecuencia de las palabras considerando los unigramas, bigramas y trigramas o cualquiera de sus combinaciones; la matriz de *tfidf* calculada a partir de la representación anterior y el promedio de los *embbedings*. Fue seleccionada la representación de *tfidf* basada en unigramas y bigramas debido a que es la más recomendada en el estado del arte, para ello se escojen solo las 10 mil palabras con mayor frecuencia en el Dataset.

4 Evaluación y Resultados

Para cada modelo se realizaron 35 iteraciones de entrenamiento, realizando *cross validation* en cada una de ellas (salvo **KNN**) y comparando el resultado de *cross validation* con la precisión del modelo sobre el Test Set. Nótese que en cada iteración se le hizo *shuffle* al DataSet, separándolo en Training Set y Test Set, esto garantiza que se obtengan resultados similares con cualquier Dataset de Revolico. Después de la fase de entrenamiento, estos fueron los resultados más relevantes:

Como puede observarse en la tabla 1 la precisión del modelo KNN es inferior al 50 %, realmente mala, esto no quiere decir del todo que KNN sea un mal modelo para este tipo de problemas, sino que debido al elevado número de dimensiones en los vectores

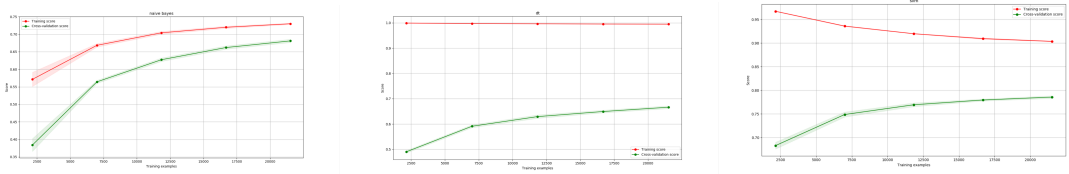
Modelo	Cross Avg	Test Avg
Naive-Bayes	0.6813513756665353	0.6942837457372502)
Árbol de Decisión	0.6668406934979795	0.6727546074377215
KNN	-	0.4846087286621637
SVM	0.7827139612431993	0.7884421567726132

Cuadro 1: Valores promedios de la precisión en el *cross validation* y en el Test Set de cada modelo

(10000) la métrica utilizada, en este caso la distancia euclideana, no funciona.
Con los modelos de Naive-Bayes y Árbol de decisión, se obtienen resultados similares tanto en el *cross validation* como en el test set, alrededor de 70 % de precisión.
Sin embargo, es SVM el modelo que mejores números arroja: una precisión de casi 80 % tanto en el *cross validation* como en el test set, lo que demuestra sin lugar a dudas que es el mejor de estos modelos.

4.1 Curvas de Aprendizaje

Como puede observarse en la Fig 1, los resultados mostrados en la tabla 1 son compatibles con la información que nos muestran las curvas de aprendizaje de estos modelos, ratificando una vez más que **SVM** es el mejor de estos modelos. La gráficas de la Fig 1, muestran la relación precisión-tamaño del training set de los modelos especificados. La línea roja representa la precisión en el training set y la verde refleja la precisión del *cross validation*.



(a) Naive-Bayes (b) Árbol de Decisión (c) SVM

Figura 1: Curvas de aprendizaje obtenidas a partir de los modelos de Naive-Bayes, Árbol de Decisión y SVM

5 Conclusiones

Después de realizado el trabajo propuesto en la sección 1.3, puede llegarse a la conclusión de que SVM es el mejor modelo de los estudiados en esta investigación para atacar este tipo de problemas. Si bién un modelo no es mejor que ninguno de manera general (*No Free Launch Theorem*), en el caso particular de la clasificación de anuncios puede recomendarse que SVM es el más acertado de los cuatro.

6 Trabajo Futuro

A partir del estudio, modelación e implementación de soluciones para el problema planteado en la sección 1.1, se plantea como línea futura de investigación, el análisis de multi-clasificadores gerárquicos que sean capaces no solo de detectar a qué categoría específica pertenece un anuncio, sino también detectar relaciones gerárquicas entre las categorías, por ejemplo: un anuncio de sub-categoría "Laptops"pertenece también a la categoría Çomputadoras".

Referencias

[1] Baharudin, B; Khan, A; Khan, K & Lam-Hong, L. *A Review of Machine Learning Algorithms for Text-Documents Classification*

[2] Even-Zohar, Y & Roth, D. *A Sequential Model for Multi-Class Classification*

[3] Ikonomakis, M; Kotsiantis, S & Tampakas, V. (2005). *Text Classification Using Machine Learning Techniques*

[4] Joachims, T. *A Statistical Learning Model of Text Classification for Support Vector Machines*

[5] Mahinovs, A & Tiwari, A. (2007). *Text Classification Method Review*

[6] McCallum, A & Nigam, K. *A Comparison of Event Models for Naive Bayes Text Classification*

[7] Nasser, M. *Ad classification using the text description on Divar dataset*

[8] Sebastiani, F. (2002). *Machine learning in automated text categorization*