Aplicación de diferentes clasificadores Bayesianos para el problema de detección de engaño en reseñas

Laritza Coello Guilarte, Jonathan Serrano Pérez

INAOE, Puebla, México,
{laritza.coello ,js.perez}@inaoe.mx

Resumen En este trabajo se han aplican diferentes clasificadores Bayesianos en el problema de detección de engaño en reseñas. Si bien dentro de la clasificación de textos, el Naïve Bayes es uno de los mas usados, en este trabajo se muestra que el uso de otros clasificadores Bayesianos pueden mejorar la exactitud obtenida por el Näive Bayes. Para la representación de los documentos se ha usado una representación de bolsas de palabras con pesado binario.

Keywords: detección de engaño, clasificadores Bayesianos

1. Introducción

El acceso a internet de las personas se ha incrementado en los últimos años. No solo es usado como una vía de comunicación sino también como medio de información. Las compras y reservaciones por internet son actividades que se realizan a diario y cada día aumenta el número de personas que acceden a estas operaciones. Muchas han sido las estrategias que se han implementado para que los clientes puedan decidir de manera rápida y fácil entre varios productos o servicios. Entre ellas se destaca el permitirle a otros clientes publicar sus opiniones e incluso darle calificación a un producto o servicio. Según el sitio *Pew Research Center* [1] el 82 % de las personas en Estados Unidos leen las opiniones de un producto antes de comprarlo. Lo que muestra que las opiniones que dejan compradores previos tiene un gran poder sobre la toma de decisiones de potenciales compradores. Aparejado a eso, los vendedores pueden tomar ventaja de esos criterios y mejorar o incrementar su trabajo para la satisfacción de los clientes, sin embargo, otros vendedores aprovechando este poder de las opiniones se dan a la tarea de publicar opiniones falsas, ya sea positivas a su favor o negativas para la competencia. El análisis de reseñas de opiniones es un área que ha sido de interés para muchos investigadores. Dentro del análisis se destaca la detección de la veracidad o no de las opiniones publicadas en diferentes sitios web. Según Ott [3] entre el 1 y el 6 % de las opiniones positivas acerca de hoteles son falsas. Por este motivo el objetivo de este trabajo es detectar engaño en reseñas de opiniones.

[1] http://www.pewresearch.org/

2. Trabajos relacionados

La opiniones engañosas ha cobrado gran auge en el mundo de web; en la mayoría de los casos no son detectadas por un lector humano. Por esa razón muchos investigadores han decidido estudiar la detección de opiniones engañosas en un conjunto de reseñas.

Sebastiani et al. [6]) hacen uso de la clasificación basada en n-gramas para detectar opiniones verdaderas o engañosas.

Hancock et al. [7] usaron la detección de engaño psicolingüístico, en el que las declaraciones engañosas ejemplifican los efectos psicológicos de la mentira, como el aumento de la emoción negativa y el distanciamiento psicológico.

Feng et al. [5] en el 2012 realiza un análisis de la estilometría sintáctica para la detección de engaño. Hacen uso de los árboles de análisis para mejorar los resultados alcanzado un 91.2 de exactitud. Realizando un análisis profundo de la sintaxis usan cuatro codificaciones diferentes de reglas de producción basadas en los árboles de análisis gramatical del contexto probabilístico (PCFG).

Ott et al. [2] en año 2013 realizan una comparación entre tres jueces humanos y un enfoque supervisado respecto a la detección de opiniones engañosas, y concluyen que el enfoque supervisado usando unigramas y bigramas de palabras, y haciendo distinción a tres características fundamentales del lenguaje como el cambio en el uso del singular en primera persona, disminución de la conciencia espacial y la escritura imaginativa, aumento de los términos de emoción negativa. Para la obtención de estos resultados lo hacen con el clasificador Máquina de Soporte Vectorial (SVM) y alcanzan un valor de 86.0 de exactitud.

3. Metodología

El presente proyecto tiene como objetivo usar un enfoque supervisado para la detección de opiniones engañosas haciendo uso de clasificadores bayesianos como son *Naïve Bayes* (NB), *Tree augmented Bayesian Classifier* (TAN), *Bayesian Network augmented Bayesian Classifier* (BAN) y *Semi-Naive* (SN).
A continuación se describe el corpus y los clasificadores utilizados (para mas detalle de los modelos revisar capitulo 4 de [4]).

3.1. Corpus y preprocesamiento

El corpus que ha sido usado es OpSpam [1,2] que tiene opiniones sobre hoteles de Chicago, este corpus esta dividido por polaridad, es decir, un grupo de opiniones positivas y el otro grupo de negativas, cada grupo tiene 800 opiniones, de las cuales 400 son opiniones verdaderas y 400 son de engaño (falsas). Las opiniones verdaderas fueron extraídas de opiniones de *TripAdvisor*, mientras que las de engaño fueron obtenidas vía *Amazon Mechanical Turk*. Sin embargo, para este trabajo la polaridad es ignorada, por lo tanto se tienen 800 opiniones verdaderas y 800 opiniones engañosas.

El preprocesamiento del corpus fue mínimo, es decir, solo se hizo reducción a minúsculas y se eliminaron signos de puntuación.

La representación de los datos que se uso es una representación vectorial, específicamente, bolsa de palabras (BoW) con pesado binario, la cual consiste en construir una matriz M de tamaño igual al número de documentos (de entrenamiento) por el tamaño del vocabulario (de los documentos de entrenamiento), donde a cada celda $M_{i,j}$ se le asigna 1 si el i-ésimo documento contiene a la j-ésima palabra o 0 en caso contrario. Esta representación es de las más usadas en el área de análisis de textos.

3.2. Naïve Bayes

Este algoritmo toma como suposición que todos los atributos son independientes dada la clase. Solo se requiere la probabilidad a priori de la clase y las n probabilidades condicionales de cada atributo dada la clase, como parámetros para el modelo.

$$P(C|A_1, A_2, ..., A_n) = P(C) * P(A_1|C) * P(A_2|C) * ... * P(A_n|C)/P(\mathbf{A}) \quad (1)$$

Por lo tanto, la probabilidad de una clase dada cierta configuración de atributos esta dada por la ecuación 1, donde $P(C)$ es la probabilidad a priori de la clase, $P(A_i|C)$ es la probabilidad del i-ésimo atributo dada la clase y $P(\mathbf{A})$ es una constante de normalización

3.3. Tree augmented Bayesian Classifier (TAN)

Incorpora algunas dependencias entre los atributos construyendo un árbol dirigido entre las variables de atributo. Los n atributos forman un grafo que está restringido a un árbol dirigido que representa las relaciones de dependencia entre los atributos. Además, hay un arco entre las variables de clase y cada atributo.

3.4. Bayesian Network augmented Bayesian Classifier (BAN)

Considera que la estructura de dependencia entre los atributos constituye un grafo acíclico dirigido. Al igual que con el clasificador TAN, hay un arco dirigido entre el nodo de la clase y cada atributo.

3.5. Semi-Naive Bayes (SN)

Este clasificador ha sido implementado en Python por los autores de este trabajo, tomando como base el algoritmo del capitulo 4.4 del libro [4]. la formula para medir la información mutua (IM) entre un atributo y la clase, se muestra en la ecuación 2, esta formula se ha usado con un umbral de 0.0035, es decir, aquellos atributos que no superen este umbral son eliminados.

$$IM(a, C) = \sum_{a} \sum_{C} P(a, C) log \left[\frac{P(a, C)}{P(a)P(C)} \right] \quad (2)$$

La formula para calcular la información mutua condicional (IMC) se muestra en la ecuación 3, esta formula se ha usado con un umbral de 0.01, es decir, de la pareja de atributos que superan este umbral dada la clase, se elimina el atributo que tiene la menor IM con respecto a la clase.

$$IMC(a_i, a_j|C) = \sum_{C} \sum_{a_i} \sum_{a_j} P(a_i, a_j, C) log \left[\frac{P(a_i, a_j, C)P(C)}{P(a_i, C)P(a_j, C)} \right] \quad (3)$$

Al aplicar este modelo de Semi-Naive, se ha reducido el numero de atributos como se muestran en la tabla 1, donde de mas de 8000 atributos quedan alrededor de 350 atributos. Además, en la tabla 2, se muestran los tiempos que le toma al Semi-Naïve clacular la IM y la IMC.

Tabla 1. Reducción de atributos al aplicar el Semi Naïve

Tam. Diccionario	**IM**	**IMC**
8469.6±(114.67)	374.2±(12.31)	353±(12.5)

Tabla 2. Tiempos que le toma al Semi Naïve (segundos)

	IM	**IMC**
Fold 1	19.5	4344.98
Fold 2	20.13	4805.7
Fold 3	19.32	4911.15
Fold 4	19.74	4517.46
Fold 5	19.61	5233.22
Promedio	19.66±(0.30)	4762.502±(346.37)

Si bien, el algoritmo menciona 3 opciones si el umbral de la IMC es superado (eliminar alguno de los dos atributos o unirlos en una sola varible), lo que se ha hecho en esta implementación es eliminar aquel atributo que tenga la menor información mutua con respecto a la clase. Esto se ha hecho por las siguientes razones, de hacer la unión de atributos haría que el espacio de búsqueda sea extremadamente grande, además de que esto implica hacer varias itereraciones, ya que atributos nuevos serían agregados, por lo que el tiempo tomado por los 5 pliegues crecerá con el numero de iteraciones.

4. Experimentos y Resultados

Para la realización de los experimentos se procedió a particionar los datos en 5 ficheros de entrenamiento y 5 ficheros de prueba, usando el criterio de

validación cruzada con 5 pliegues. Para los clasificadores NB, BAN y TAN se han usado las implementaciones de WEKA [2]. La tabla 3 muestra los resultados de la clasificación obtenidos por los diferentes clasificadores.

Tabla 3. Exactitud obtenida por los diferentes clasificadores para cada pliegue

	NB	**TAN**	**BAN**	**SN**
Fold 1	86.2	80.6	**88.1**	78.4
Fold 2	85.9	85.3	**87.2**	81.2
Fold 3	**83.1**	81.6	**83.1**	80.9
Fold 4	**88.7**	87.8	88.1	83.7
Fold 5	**84.3**	80.3	82.5	75.6
Promedio	85.64	83.12	**85.8**	79.96

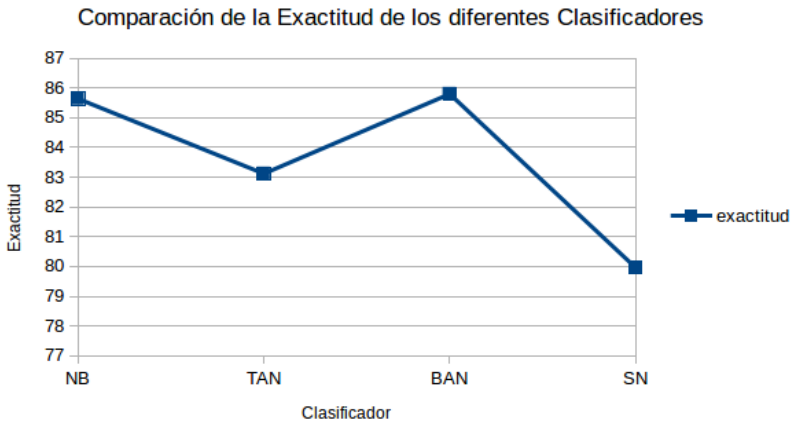

Figura 1. Gráfica con la exactitud obtenida por cada clasificador

En la Figura 1 se muestran los resultados de la exactitud de clasificación que obtuvo cada clasificador. Los dos mejores resultados corresponden al algoritmo BAN con un 85.8 seguido del Naive Bayes con un 85.6 de exactitud. Posteriormente, se hizo un análisis de los tiempos de ejecución para cada uno de los algoritmos, nuevamente por pliegue, y al final el promedio de los tiempos con su desviación estándar. Estos resultados se muestran en la tabla 4.

[2] https://www.cs.waikato.ac.nz/ml/weka/

Tabla 4. Tiempo de ejecución por clasificador (en segundos)

	NB	**TAN**	**BAN**	**SN**
Fold 1	0.29	10210.08	758.68	4376.54
Fold 2	0.25	11416.11	569.55	4833.97
Fold 3	0.22	11317.9	565.23	4938.61
Fold 4	0.25	12318.18	622.51	4545.26
Fold 5	0.25	11994.98	580.78	5261.03
Promedio	**0.25**±(0.02)	11451.45±(807.26)	619.35±(81.11)	4791.08±(345.22)

Los tiempos de ejecución para los dos algoritmos que arrojaron mejores resultados son muy distantes. El Naive Bayes demora un tiempo promedio de 0.25s con una desviación estándar de 0.02s, mientras que el BAN demora un promedio de 619.35s (10.32 min) con una desviación estándar de 81,12s (1.35 min).

4.1. Prueba estadística entre NB y BAN

La exactitud para el problema de detección de engaño es de 85.64 usando un clasificador NB, sin embargo, 5 pruebas usando un clasificador BAN (tabla 3 columna de BAN), muestran que la exactitud es de 85.8, Por lo que se quiere analizar si la exactitud del NB ha sido superada usando una significancia del 0.05.

De este modo la hipótesis nula H_0 indica que la exactitud no ha cambiado, y la hipótesis alternativa H_a indica que la exactitud ha sido superada.

$$H_0 : \mu = 85{,}64 \tag{4}$$

$$H_a : \mu > 85{,}64 \tag{5}$$

Dado que se tienen 5 pruebas y no se conoce la desviación estándar, el estadístico de prueba que se usa es el *t de Student* (ecuación 6),

$$\frac{\overline{X} - \mu}{\frac{S}{\sqrt{n}}} \sim t_{n-1} \tag{6}$$

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \overline{X}\right)^2 \tag{7}$$

El valor de t_4 con una significancia del 0.05 es igual a 2.132 (este valor es obtenido de la tabla de distribución de una *t de Student*), por lo que las decisiones quedan de la siguiente manera:

- Se rechaza H_0 si el valor observado del estadístico de prueba es mayor que 2.132.
- No se rechaza H_0 si el valor observado del estadístico de prueba es menor que 2.132.

Calculando el valor del estadístico con la ecuación 6 y tomando como $\overline{X}$ =85.8, se obtiene un valor de 0.12910. Por lo tanto, no se rechaza H_0.

Esto indica que la exactitud obtenida por el clasificador BAN no es significativamente mayor que la del NB con una significancia del 0.05.

5. Conclusiones

El algoritmo que mayor exactitud arroja para la representación BoW con pesado binario es el BAN con una exactitud del 85.8, seguido del Naive Bayes con una exactitud de 85.6, sin embargo, como se mostró en la prueba estadística, el resultado obtenido por el BAN no es estadísticamente mejor que el de NB, además de que calcular el BAN es mucho más tardado que el NB, esto es, el clasificador NB es más de 2000 veces mas rápido que el BAN.

El clasificador peor evaluado fue el Semi-Naïve, esto se puede deber a la eliminación de atributos que podrían ayudar a discriminar entre las clases, por lo que umbrales menos estrictos podrían mejorar la exactitud. Sin embargo, se debe considerar que tener más atributos implica mas tiempo de ejecución.

6. Trabajo Futuro

Probar los diferentes clasificadores Bayesianos, pero usando diferentes representaciones de los documentos, por ejemplo, bolsa de palabras con pesado por frecuencias o tfidf; usar una representación tipo word2vec. etc. Una implementación completa del Semi-Naïve en un lenguaje compilado como C. Un ensamble con los 4 clasificadores sería interesante.

Referencias

1. Myle Ott, Yejin Choi, Claire Cardie, Jeffrey T. Hancock
   Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. (2011)
2. Myle Ott, Claire Cardie, Jeffrey T. Hancock
   Negative Deceptive Opinion Spam. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. (2013)
3. Myle Ott, Claire Cardie, and Jeff Hancock.
   Estimating the prevalence of deception in online review communities. Proceedings of the 21st international conference on World Wide Web. (2012)
4. L. Enrique Sucar
   Probabilistic Graphical Models Principles and Applications. Advances in Computer Vision and Pattern Recognition Springer (2015)
5. Song Feng, Ritwik Banerjee, Yejin Choi
   Syntactic Stylometry for Deception Detection. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. (2012)

6. Fabrizio Sebastiani
Machine Learning in Automated Text Categorization ACM Computing Surveys, Vol. 34. (2002)
7. Jeffrey T. Hancock, Lauren E. Curry, Saurabh Goorha
On Lying and Being Lied To: A Linguistic Analysis of Deceptionin Computer-Mediated Communication. Discourse Processes A Multidisciplinary Journal. (2008)