Métodos de detección de spam: perceptrón multicapa versus Naive-Bayes

Andrés A. Gilli, Guido Ghisolfi y Julio A. Lucero
Trabajo práctico final de "Inteligencia Computacional", II-FICH-UNL

***Resumen*—Estadísticas indican que el porcentaje de correos no deseados que ingresa a cada cuenta supera claramente el setenta por ciento sobre el total de correos. Esto implica claramente un inconveniente para las empresas proveedoras de estos servicios ya que les genera un costo adicional. Actualmente existen varios métodos diseñados para la detección de spam, los mismos se dividen en dos ramas, los métodos basados en la fuente que genera el correo y los métodos basados en el contenido de los mismos; algunos, como el uso de "Listas Negras" ya fueron sorteados por las personas que envían spam (spammers). Los dos métodos que se comparan en este documento son métodos basados en el contenido. La implementación de ambos métodos consta de tres etapas: extracción de características, entrenamiento y prueba. Luego de evaluar el método implementado con un perceptrón multicapa y el implementado con el algoritmo de Naive-Bayes con la misma base de datos, se pudo concluir que con el perceptrón multicapa obtuvo mayor precisión en la clasificación, arrojando una tasa de acierto superior al noventa y seis por ciento (96 %).**

***Palabras clave*—correo no deseado, correo electrónico, spam, e-mail, Naive-Bayes, perceptrón multicapa.**

I. Introducción

Una de las consecuencias de la masiva utilización del correo electrónico es el envío de correo electrónico no deseado (spam). Las estadísticas indican que más del setenta por ciento de los correos electrónicos que ingresan en una cuenta son de tipo spam. Esto genera varios problemas tanto a las empresas que brindan servicios de correo electrónico como las empresas proveedoras de internet (ISP). Ante esto, dichas empresas se encuentran en una constante lucha por filtrar los spam.

Se han encontrado diversas soluciones a este inconveniente pero nos centraremos en las soluciones basadas en el análisis del contenido. En [2] se proponen varias soluciones al problema del filtrado de spam, algunas basadas en redes neuronales y otras basadas en el teorema de Bayes, obteniendo como resultados finales para ambos métodos una tasa de acierto del orden del noventa por ciento (90 %). Mientras que en [3], se propone una solución que conjuga la teoría de las probabilidades de Bayes con un perceptrón multicapa, lo cual aparenta ser muy prometedor, ya que ofrece como resultados tasas de acierto superiores al noventa y ocho por ciento (98 %). Por otro lado, en [6] se utilizan cuatro técnicas de filtrado utilizando el teorema de Bayes, entre las cuales se obtiene como mejor resultado una tasa de acierto algo superior al noventa por ciento (90 %), en técnica similar a la utilizada en este trabajo.

El presente trabajo compara método de detección de spam basado en un perceptrón multicapa y otro basado en el algoritmo de Naive-Bayes de manera independiente. De esta forma podremos extraer las principales diferencias, ventajas y desventajas de cada uno de ellos.

Con el objetivo de lograr una comparación equilibrada entre ambos métodos se realizaron algunas transformaciones sobre el texto de los correos y se establecieron algunas pautas de trabajo que se detallan a continuación: se analiza solo el contenido del asunto y el cuerpo del email; se considera las palabras con más de tres letras y menos de quince; se filtran caracteres especiales, números y tag de lenguaje html; todas las palabras se transforman a minúsculas.

A continuación, en la sección II se detallará el método basado en un perceptrón multicapa, en la sección III se detallará el método basado en el algoritmo de Naive-Bayes, en la sección IV se mostrarán los resultados obtenidos para los distintos métodos y en la sección V se establecen las conclusiones del presente trabajo.

II. Perceptrón multicapa

En este método se entrena una red neuronal implementada mediante el algoritmo perceptrón multicapa. Como primera medida se necesita encontrar alguna representación numérica para los datos de entrada. Una vez definidas cuales son las entradas a la red, se necesita definir una arquitectura para la misma, para luego entrenarla y evaluar los resultados.

A. Extracción de características

En esta etapa se propone representar un correo electrónico mediante la cantidad de ocurrencias de cada palabra, perteneciente a una lista de palabras frecuentes, en el correo en cuestión. Por lo tanto, cada correo estará representado por un vector v_i donde cada componente del vector indicará la cantidad de veces que apareció la palabra i en el mismo.

Para poder implementar esta representación se necesita, como primera medida, generar la lista de palabras frecuentes nombrada en el párrafo anterior. Para hacerlo, se leen todas las palabras de todos los correos que se tengan en la base de datos de entrenamiento. A partir de ello, y mediante la utilización de algunas técnicas computacionales , se genera una lista con las n palabras de mayor ocurrencia en los datos extraídos. A continuación se muestra un ejemplo de lectura para un correo.

Lista de Palabras = [“hello”,”free”,”webcam”]
Asunto = [“welcome”, ”free”, ”site”]
Cuerpo=[”welcome”,”our”,”free”,”webcam”,”site”]
Entrada a la Red Neuronal = [0, 2, 1]

B. Arquitectura y entrenamiento

Tal como lo indica el título del método, se utiliza para hacerlo efectivo un perceptrón multicapa. La cantidad de entradas del mismo quedan definidas por el tamaño de la lista de palabras frecuentes, en este caso *n*. El perceptrón estará compuesto por una capa de Entrada con *n* neuronas, una capa oculta con *r* neuronas y una capa de salida con una única neurona que indicará si el correo es o no de tipo spam.

El entrenamiento del mismo se efectúa mediante el algoritmo de retro propagación, utilizando como función de activación una función sigmoidea simétrica, ver (1). La ventaja de esta función, es que además de tener derivada continua, su derivada se puede expresar en función de si misma. Ver (2). Luego, se puede generalizar la expresión que regirá la actualización de pesos. En (3) se puede ver dicha expresión.

$$y_j = \phi(v_j) = \frac{2}{1+e^{-v_j}} - 1 \qquad (1)$$

$$\frac{\partial y(n)_j}{\partial v(n)_j} = \frac{1}{2}(y(n)_j + 1)(y(n)_j - 1) \qquad (2)$$

$$\Delta w_{ji}^{(p)}(n) = \eta < \delta^{(p+1)}, w_j^{(p+1)} > (1 + y_j^{(p)}(n))(1 - y_j^{(p)}(n)) y^{(p-1)} i(n) \qquad (3)$$

Donde *p* denota el número de capa, *w* denota un peso, *j* e *i* denota el número de neurona y por último *ji* indica un enlace entre la neurona *j* y la neurona *i*.

Entonces, el error de la última capa, que es la diferencia entre la salida deseada y la salida obtenida de la red, se propaga hacía atrás siguiendo la expresión que se muestra en (3). Esto se realiza iterativamente para cada correo perteneciente a la base de datos de entrenamiento mientras que no se alcance ningún criterio de finalización. En este caso se establecieron dos criterios de finalización para el entrenamiento. El primer criterio de finalización es un máximo de épocas de entrenamiento. El otro criterio de corte se da cuando la red neuronal supera la tasa de aciertos requerida.

III. Naive-Bayes

Este es un método probabilístico basado en el teorema de Bayes. El mismo se estructura en dos fases, el entrenamiento y la clasificación.

A. Entrenamiento

Este proceso consiste en armar dos listas conformadas con las palabras recurrentes en los correos del tipo spam y otra con los que no lo son. Las mismas están conformadas por la cantidad de ocurrencias de cada palabra y una probabilidad denominada "probabilidad a priori".

La probabilidad de que las palabras pertenezcan a un correo del tipo spam se expresa en (4), mientras que la probabilidad de que las palabras no pertenezcan a un correo del tipo spam, se expresa en (5).

$$P_{SPAM}(t_i) = \frac{P(SPAM \cap t_i)}{P(t_i)} = \frac{\frac{b_{t_i}}{B}}{\frac{b_{t_i}}{B} + \frac{g_{t_i}}{G}} \qquad (4)$$

$$P_{NSPAM}(t_i) = \frac{P(NSPAM \cap t_i)}{P(t_i)} = \frac{\frac{g_{t_i}}{G}}{\frac{g_{t_i}}{G} + \frac{b_{t_i}}{B}} \qquad (5)$$

Donde t_i indica una palabra o token, g_{ti} indica la frecuencia del token t_i en correos no spam, b_{ti} indica la frecuencia del token t_i en correos spam, G es la cantidad de correos no spam y B es la cantidad de correos spam.

Luego de obtener las bases de datos se seleccionan las palabras mas importantes, es decir, con probabilidad cercana a cero o cercana a uno; las demás palabras son descartadas; también se descartan las palabras que tengan poca cantidad de ocurrencias.

B. Clasificación

En esta etapa se calcula la probabilidad de que el correo que ingresa sea del tipo spam. Como primer medida se recupera la información correspondiente a cada palabra o token del correo ingresado, luego se seleccionan las palabras más importantes dentro del correo. Éstas darán lugar al cálculo de la probabilidad combinada total. La cual se puede expresar de la siguiente manera:

$$P_{SPAM}(t_1, t_2, \ldots, t_n) = \frac{\prod_{i=1}^{n} P_{SPAM}(t_i)}{\prod_{i=1}^{n} P_{SPAM}(t_i) + \prod_{i=1}^{n} P_{NSPAM}(t_i)} \qquad (6)$$

En (6), se expresa la probabilidad de que un correo determinado sea del tipo spam de acuerdo a las palabras que contiene. Ésta es igual a la probabilidad de que las palabras que aparecen en el correo sean de de tipo spam dividida la probabilidad de que las palabras aparezcan en cualquier correo. En esta etapa puede que aparezcan palabras que no existan en la base de datos de entrenamiento y por tal motivo no se contará con ninguna información sobre las probabilidades de dicha palabra. Este inconveniente se soluciona asignando a la palabra en cuestión una probabilidad muy baja de que sea de tipo spam o no spam. En (7) se muestra la expresión para que se utiliza para definir esa probabilidad.

$$P_{SPAM}(t_i) = \frac{P(SPAM \cap t_i)}{P(t_i)} = \frac{\frac{b_{t_i}}{B}}{\frac{b_{t_i}}{B} + \frac{0{,}01}{(CpS+1)}} \qquad (7)$$

Los términos de (7) se detallan luego de (5) excepto el término CpS que se define como la cantidad de correos del tipo spam. Como vemos esta expresión relaciona la probabilidad que se asignará a las palabras no existentes con la cantidad de palabras del tipo spam o no spam.

IV. Resultados y Discusión

Para generar los resultados se utilizo una base de datos de mil correos electrónicos donde quinientos eran del tipo spam. A partir de los mismos se generaron cinco particiones de datos, cada una con el ochenta por ciento (80 %) de los correos para entrenamiento y el veinte por ciento (20 %) restante para la etapa de pruebas.

A. Perceptrón multicapa

Se han probado distintas configuraciones, tanto para la red neuronal como para el tamaño de la lista de palabras frecuentes. Sin embargo, se ha comprobado que al aumentar la cantidad de entradas a la red, aumenta demasiado el costo computacional y no aumenta en la misma medida la calidad de la solución.

A continuación se muestran los resultados obtenidos con una lista de palabras frecuentes de cien palabras, que representarán la cantidad de entradas a la red. Luego, se define una capa de entrada con cincuenta neuronas, una capa oculta de cuatro neuronas y una capa de salida de una neurona.

TABLA I

Resultados parciales del error de clasificación, falsa alarma y acierto (validación cruzada sobre 5 particiones de entrenamiento y prueba).

Validación Cruzada	Nº Partición				
	1	2	3	4	5
% Error	2	1	1	0	0
% Falsa Alarma	1	1	2	5	3
% Acierto	97	97	97	94	96

TABLA II

Resultados finales de la media y varianza del error de clasificación, falsa alarma y acierto (validación cruzada sobre 5 particiones de entrenamiento y prueba).

Indicadores	Media	Varianza
% Error	0.8	0.75
% Falsa Alarma	2.4	1.5
% Acierto	96.2	1.17

B. Naive-Bayes

En este caso también se han probado distintas configuraciones. Los parámetro ajustables fueron la cantidad de correos destinados al entrenamiento y la variación en el umbral para la clasificación de un correo como spam. Experimentalmente se estableció que si un correo tenía una probabilidad mayor a 0.8 de ser spam, entonces se lo consideraba como tal.

TABLA I

Resultados parciales del error de clasificación, falsa alarma y acierto (validación cruzada sobre 5 particiones de entrenamiento y prueba).

Validación Cruzada	Nº Partición				
	1	2	3	4	5
% Error	6	3.5	5.5	4	4
% Falsa Alarma	8	8.5	9	9.5	10
% Acierto	86	88	85.5	86.5	86

TABLA II

Resultados finales de la media y varianza del error de clasificación, falsa alarma y acierto (validación cruzada sobre 5 particiones de entrenamiento y prueba).

Indicadores	Media	Varianza
% Error	4.6	0.97
% Falsa Alarma	9	0.71
% Acierto	86.4	0.86

C. Discusión

Bajo las condiciones fijadas sobre los datos, que se plantean al inicio de esta sección, se puede decir que el método basado en un perceptrón multicapa presenta mejores resultados que el método algoritmo de Naive-Bayes.

Esto se puede explicar debido a que el método basado en el teorema de Bayes se ve muy influenciado por la cantidad de correos que se tienen en la base de datos de entrenamiento. Se muestra en la Figura 1 como disminuye el error a medida que crecen los datos para el entrenamiento.

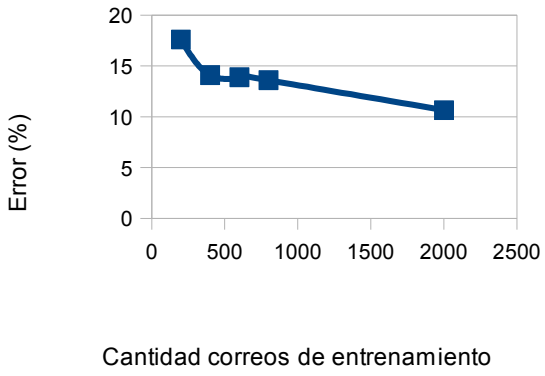

Fig. 1: Error versus cantidad de correos de entrenamiento

Esto implica claramente una desventaja respecto al método basado en un perceptrón multicapa. Otra desventaja de este método es que es muy susceptible a correos en los cuales se incluyan muchas palabras de tipo no spam para enmascarar le correo. La gran ventaja que presenta es que es fácilmente actualizable, y por tal motivo, a medida que ingresan más correos a la base de datos se puede ir actualizando y aumentando de esa forma su tasa de aciertos, de esta forma se va reduciendo la primer desventaja planteada.

El mayor inconveniente del método basado en un perceptrón multicapa es que al tener mayor cantidad de correos se hace necesario aumentar la cantidad de palabras frecuentes con las que trabaja y esa cantidad de palabras define la cantidad de entradas que tendrá la red, lo cual implica un gran crecimiento en la cantidad de pesos de la red. Otra desventaja de este método es que generar un algoritmo para mantener actualizada la red es más complejo que generar uno para actualizar el otro método. Dentro de las ventajas de este método es que no se ve tan influenciado como Naive-Bayes al analizar correos

enmascarados y la más importante es que no necesita gran cantidad de correos para producir buenos resultados.

V. Conclusiones

El trabajo pretende comparar dos métodos muy usados para la detección de correo electrónico no deseado y determinar cual es mas efectivo y en que circunstancias.

Bajo las mismas condiciones notamos que ambos métodos son eficientes, sin embargo el perceptrón multicapa presenta mejores resultados (alcanzando un 97% de aciertos en el mejor de los casos). Aun así, el algoritmo de Naive-Bayes disminuye en gran medida su tasa de error cuando se incrementan la cantidad de correos electrónicos para su entrenamiento, conformando así una base de datos con mas palabras, (90% de aciertos utilizando 2000 correos como entrenamiento).

Una propuesta futura para mejorar aún mas estos métodos consistiría en implementar un híbrido entre ambos métodos. La misma se organizaría en tres fases. En la primera se obtendrían las palabras mas frecuentes utilizando Naive-Bayes y seleccionando aquellas que tengan mayor probabilidad, la segunda fase sería el entrenamiento de los pesos del perceptron multicapa tomando como entradas las ocurrencias de las palabras obtenidas en la fase 1, y la fase 3 sería la de test, que determinaría la clasificación del correo. La solución mejoraría considerablemente y se lograrían tasas de aciertos mayores con menos entradas en la red, lo que reduciría el costo de entrenamiento.

Referencias

[1]C. M. Bishop (1995). Neural Network for Pattern Recognition. Oxford: Claredon Press.

[2]L. Özgür, T. Güngör y F. Gürgen. “Adaptive anti-spam filtering for agglutinative languages: a special case for Turkish”. Pattern Recognition Letters, Volume 25, Issue 16, December 2004, Pages 1819-1831.

[3]O. A. S. Carpinteiro; I. Lima; J. M. C. Assis; A. C. Zambroni de Sousa; E. M. Moreira y C. A. M. Pinheiro (2006). “A Neural Model in Anti-spam Systems”. Artificial Neural Networks – ICANN 2006. Springer Berlin / Heidelberg: Volume 4132, Start Page 847, End Page 855, URL: http://dx.doi.org/10.1007/11840930_88

[4]R. Duda, P. Hart y D. Stork (2000). *Pattern Classification, Second Edition*. Reino Unido: John Wiley & Sons.

[5]S. Haykin (1999). Neural Networks: A Comprehensive Foundation, Second Edition. Hamilton, Ontario, Canadá: Prentice Hall International.

[6]V. P. Deshpande, R. F. Erbacher y C. Harris (2007). “An Evaluation of Naïve Bayesian Anti-Spam Filtering Techniques”. Proceedings of the 2007 IEEE - Workshop on Information Assurance. United States Military Academy, West Point, NY 20-22 June 2007

[7]Y. Song; A. Kolcz y C. Lee Giles (2009). “Better Naive Bayes classification for high-precision spam detection”. Wiley InterScience. Software Practice Experience.