Procesamiento del Lenguaje Natural, Revista nº 35, septiembre de 2005 ISSN: 1135-5948

Sociedad Española para el Procesamiento del Lenguaje Natural

Artículos

XXI Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural

XXI Congreso de la SEPLN

Universidad de Granada, 14-16 de Septiembre de 2005

I Congreso Español de Informática (CEDI 2005)

Universidad de Granada

ISSN: 1135-5948
Depósito Legal: B:3941-91

Procesamiento del Lenguaje Natural, núm. 35 (2005), pp. 127-133 recibido 04-05-2005; aceptado 01-06-2005

Detección automática de Spam utilizando Regresión Logística Bayesiana

Antonio Jesús Ortiz Martos **María Teresa Martín Valdivia**

L. Alfonso Ureña López **Miguel Ángel García Cumbreras**

Grupo Sistemas Inteligentes de Acceso a la Información
Departamento de Informática
Universidad de Jaén
http://sinai.ujaen.es
e-mail:ajortiz@amsystem.es, {maite,laurena,magc}@ujaen.es

Resumen: Este artículo presenta un sistema de detección automática de Spam, o correo no deseado, aplicando Regresión Logística Bayesiana (BBR) como técnica de aprendizaje automático, sobre la colección de correos electrónicos SPAMBASE. A modo de comparativa se han aplicado otros dos algoritmos de aprendizaje: el algoritmo SVM (Support Vector Machine), y el algoritmo PLAUM (Perceptron Algorithm with Uneven Margins). La finalidad de este estudio es comprobar la eficiencia y efectividad del algoritmo BBR en la tarea concreta de filtrado de Spam. Como muestran los experimentos, el algoritmo BBR no solo obtiene unos resultados satisfactorios en cuanto a precisión y recall, sino que además es el algoritmo más rápido de los estudiados.
Palabras clave: filtrado de correo , Spam, SPAMBASE, BBR.

Abstract: This paper presents an Spam automatic detection system using Bayesian Logistic Regression (BBR) as machine learning algorithm, over the SPAMBASE collection. We have also used two machine learning algorithms: SVM and PLAUM, in order to compare the results. Our aim is to check the efficiency and effectiveness of the BBR method. The obtained results show good results in terms of precision and recall. We have also noticed that BBR is the faster algorithm.
Keywords: e-mail filter, Spam, SPAMBASE, BBR.

1 Introducción

El uso del servicio de correo electrónico ha sufrido en los últimos años un gran crecimiento. Actualmente existe un serio problema que afecta tanto al destinatario de los correos electrónicos como a las comunicaciones a través de Internet: el envío masivo de correo no deseado o Spam. Esta práctica fraudulenta proporciona a determinados publicistas un medio de llegar a miles de posibles clientes con un coste muy reducido.

Estudios recientes ponen en evidencia la importancia de los sistemas de detección y filtrado de dichos correos. Robert Horton, de la Australian Communication Commission, señala que el costo generado por toda esta basura electrónica a individuos y empresas se calcula en unos 10.000 millones de dólares anuales sólo en Europa y podría alcanzar en todo el mundo alrededor de los 25.000 millones de dólares.

Los términos asociados habitualmente en Internet a estos tipos de abuso son *spamming*, *mail bombing*, *unsolicited bulk email* (UBE), *unsolicited commercial email* (UCE) o *junk mail*.

Para solventar este problema se plantean diversas alternativas, unas preventivas, otras disuasorias, y, como medida final, alternativas denunciantes.

Las alternativas preventivas van orientadas a evitar, en la medida de lo posible, la recepción de Spam.

ISSN: 1135-5948

Las alternativas denunciantes pretenden identificar el agresor informático y denunciarlo ante la autoridad competente.

Por último, las alternativas disuasorias, más interesantes desde el punto de vista de la investigación, están enfocadas a la detección y rechazo del mayor número de mensajes utilizando diversas técnicas. Entre esas técnicas caben destacar: detección de palabras clave en el campo asunto del correo electrónico, limitación del tamaño del correo, selección de remitentes autorizados y, sobre todo, técnicas de filtrado o clasificación de correos.

El filtrado de Spam utilizando técnicas de categorización de texto permite introducir técnicas de Machine Learning [6] que, tras el adecuado entrenamiento, arrojan unos resultados muy interesantes.

Estas técnicas exigen la construcción de un sistema parcialmente automático de aprendizaje a partir de datos de entrenamiento (ejemplos etiquetados manualmente por un experto humano). Entre las colecciones de correos electrónicos disponibles en Internet destacan SPAMBASE (UCI Machine Learning Repository) [10], LingSpam [9] o PU1 [1].

En este trabajo se estudia el comportamiento del algoritmo BBR [4] en la tarea de filtrado de Spam sobre la colección experimental SPAMBASE. Con el fin de comparar los resultados obtenidos, se han aplicado otros dos algoritmos de aprendizaje sobre dicha colección, concretamente el algoritmo SVM y el algoritmo PLAUM.

El resto del trabajo está organizado de la siguiente forma: en la sección 2 se presentan los algoritmos utilizados en esta investigación. La sección 3 describe la colección SPAMBASE, su contenido, organización, y los resultados obtenidos. La sección 4 muestra las conclusiones y los trabajos futuros.

2 *Algoritmos de aprendizaje automático*

Los sistemas comerciales disponibles actualmente utilizan algoritmos de aprendizaje bayesianos simples. Los sistemas experimentales suelen utilizar otros algoritmos mas complejos basados tanto en aprendizaje bayesiano como en otros sistemas automáticos de clasificación. Trabajos recientes en el filtrado de spam, hacen uso del algoritmo de aprendizaje SVM con unos resultados satisfactorios.

Nuestra investigación se ha centrado principalmente en la utilización del algoritmo de aprendizaje basado en regresión logística bayesiana BBR. Por otra parte, a modo de comparativa, se ha aplicado el algoritmo SVM y el algoritmo PLAUM. A continuación, se describen brevemente dichos algoritmos.

2.1 BBR (Bayesian Binary Regression)

Se trata de una implementación de la regresión logística bayesiana, aplicada a la clasificación binaria. La clave de este algoritmo es la utilización de una distribución de probabilidad previa (ver ecuación 1) y algoritmos de optimización sucesiva de los ejemplos de entrenamiento suministrados.

$$p(y=1|\beta,x)=\psi(\beta^T x)$$

Ecuación 1

Este algoritmo inicialmente realiza una regresión logística de los datos de entrenamiento a partir de la distribución de probabilidad elegida (Gausiana o Laplace), por medio de una función de enlace ψ (ver ecuación 2).

$$\psi(z)=\exp(z)/(1+\exp(z))$$

Ecuación 2

Una vez obtenido el modelo de regresión, se va optimizando sucesivamente a través de la aplicación de un algoritmo de regresión logística en cadena. Se trata de un algoritmo de optimización de coordenada cíclica descendente. Se comienza poniendo todas las variables a algún valor inicial, y se busca qué valor de la primera variable minimiza la función objetivo, asumiendo que todas las otras variables mantienen constantes sus valores iniciales. Este es un problema de optimización unidimensional. El mismo método se lleva a cabo con la segunda variable, y así sucesivamente hasta que se han cruzado todas las variables. Este proceso se repite varias pasadas hasta encontrar un criterio de convergencia.

De forma resumida, el algoritmo funciona como sigue [4]:

1. Inicializar las variables $\beta_j=0$ (vector de parámetros), $\Delta_j=0$ (conjunto de datos, o región explorada), para j=1 hasta d (número de parámetros del modelo); $\tau_i=0$ (estimador previo de confianza), para i=1 hasta n (número de ejemplos).

2. Desde k=1, 2, ... hasta que se produzca la convergencia, hacer
   2.1. Para j=1 hasta d, hacer:
      2.1.1. Calcular

$$\Delta\upsilon_j = -\frac{\left(\sum_{i=1}^{n} x_{ij} y_i \frac{1}{1+\exp(\tau_i)}\right) + 2\beta_j / \tau_j^2}{\left(\sum_{i=1}^{n} x_{ij}^2 F(\tau_j; \Delta_j | x_{ij}|)\right) + 2/\tau_j}$$

      2.1.2. Calcular
      $\Delta\beta_j = mín(máx(\Delta\upsilon_j - \Delta_j), \Delta_j)$, en los datos spam del subconjunto tratado.

      2.1.3. Calcular
      $\Delta\tau_i = \Delta\beta_j x_{ij} y_i$, con $\tau_i = \tau_i + \Delta\tau_i$, para i=1,...,n.

      2.1.4. Calcular $\beta_j = \beta_j + \Delta\beta_j$

      2.1.5. Calcular
      $\Delta_i = máx(2|\Delta\beta_j|, \Delta_i / 2)$, ampliando el tamaño del subconjunto de spam tratado.

2.2 SVM

El algoritmo SVM (Support Vector Machine) fue utilizado por primera vez en la Clasificación de Texto en 1998 por T. Joachims [7]. En términos geométricos, se puede ver como el intento de encontrar un espacio n-dimensional, que permita separar los ejemplos positivos de entrenamiento de los negativos, permitiendo especificar el margen más amplio posible.

El objetivo perseguido por este algoritmo es encontrar el hiperplano óptimo que maximice la distancia entre los casos positivos y los casos negativos.

Como argumenta Joachims [7], las máquinas de vectores de soporte ofrecen dos grandes ventajas para la categorización de texto:

- Evita los problemas de sobrecarga de pruebas en espacios de grandes dimensiones.
- Realiza una optimización global, sin óptimos locales.

Estos son los pasos principales del algoritmo SVM [2]:

1. Seleccionar el parámetro C como representante de la compensación entre la reducción al mínimo del error de clasificación del conjunto de entrenamiento y maximizar el margen.
2. Seleccionar la función de kernel y cualquier parámetro del mismo.
3. Solucionar el problema cuadrático dual resultante de la ecuación 3 o una formulación alternativa que use la programación cuadrática de forma apropiada o un algoritmo de programación lineal.

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{m} \alpha_i$$
$$s.t. \quad \sum_{i=1}^{l} y_i \alpha_i = 0$$
$$C \geq \alpha_i \geq 0 \quad i = 1,...,m$$

Ecuación 3

4. Extraer la variable de umbral b, utilizando los vectores de apoyo.
5. Clasificar un punto nuevo x, siguiendo la ecuación 4:

$$f(x) = sign(\sum_i y_i \alpha_i K(x, x_i) - b)$$

Ecuación 4

donde, x_i (i=1,...,m) son los miembros del conjunto de puntos de entrenamiento; y_i=±1 son las etiquetas de la clase; y α_i son los vectores de soporte.

En la figura 1 se pueden apreciar ejemplos de entrenamiento positivos y negativos. Las líneas representan los hiperplanos de separación. La línea más gruesa es la que minimiza la distancia para cualquier ejemplo de entrenamiento.

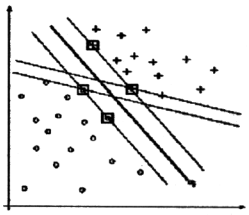

Figura 1: Representación de puntos en SVM

2.3 PLAUM (Perceptron Algorithm with Uneven Margins)

Se trata de algoritmo rápido y eficaz para realizar clasificaciones lineales. El algoritmo PLAUM es una extensión del algoritmo del

Perceptron, adaptada para tratar problemas de separación lineal de datos a través de un hiperplano [14]. Tal como SVM se basa en la idea de encontrar un margen entre hiperplanos, y sus autores aseguran que funciona mejor que SVM para tareas de clasificación de texto. Este algoritmo requiere:

- Un conjunto de entrenamiento linealmente separable de la forma: $z = (x, y) \in (X \times \{-1,+1\})^m$.
- Un índice de aprendizaje $\eta \in \Re^+$.
- Un número máximo de iteraciones T.
- Dos parámetros que limitan los ejemplos negativos y positivos: $\tau_{-1}, \tau_{+1} \in \Re^+$.

El algoritmo funciona de acuerdo a los siguientes pasos [11]:

1. Inicialización de variables:
   $iteración = 0$; i = 1; paso = m; $\vec{w} = \vec{0}$; b = 0;
   $R = máx_{z_i \in X} \|\vec{x}_i\|$
2. Repetir.
   2.1. Si $y_i((\vec{w}, \vec{x}_i) + b) \leq \tau_i$ entonces
      2.1.1. $\vec{w} = \vec{w} + \eta y_i \vec{x}_i$.
      2.1.2. $b = b + \eta y_i R^2$.
      2.1.3. $paso = i$.
   2.2. $i = i + 1$.
   2.3. Si $(i > m)$ entonces
      2.3.1. $i = 1$.
      2.3.2. $iteración = iteración + 1$.
   Hasta $(i = paso)$ o $(iteración \geq T)$.
3. Devolver (w,b).

3 *Experimentos*

3.1 Metodología de Experimentación

Los experimentos se han realizado utilizando como colección de entrenamiento y prueba de correos electrónicos SPAMBASE[1]. Esta colección fue creada en Julio de 1999 por Mark Hopkins, Erik Reeber, George Forman, y Jaap Suermondt, a partir de una recopilación de mensajes donada por George Forman. Está compuesta por 4601 casos, de los cuales 1813 (39.4%) son clasificados como "Spam".

Cada ejemplo está representado por un vector de 58 atributos, donde el último contiene la clase a la que pertenece el documento ("Spam" o "NO Spam"). El resto de atributos representan la aparición de determinados símbolos o palabras.

La colección SPAMBASE no es linealmente separable, y la selección de atributos claves, permite la separación, casi perfecta de las dos clases.

La tabla 1 muestra los tiempos medios expresados en segundos, empleados en el entrenamiento con las particiones tratadas. Los resultados de prueba son inmediatos una vez entrenado el modelo.

	ENTRENAMIENTO
BBR	**5,1 seg.**
SVM	13396,37 seg.
PLAUM	10,61 seg.

Tabla 1: Tiempos de entrenamiento sobre la colección SPAMBASE

Cabe destacar que el algoritmo BBR es el más rápido de los utilizados para entrenamiento, con un margen muy amplio respecto del algoritmo SVM.

3.2 Resultados obtenidos

Como método de evaluación de los experimentos realizados hemos utilizado Validación Cruzada o *"10-fold Cross Validation"* [13], consistente en hacer diez particiones iguales de la colección y utilizar de forma alternativa nueve partes para entrenamiento y la parte restante para prueba. Este proceso se repite 10 veces variando la partición utilizada en la evaluación. El resultado final obtenido es una media de los diez resultados parciales.

Para cada una de las ejecuciones, hemos obtenido una tabla de contingencia, del estilo mostrado en la tabla 2, donde:

- A es el número de documentos Spam clasificados como Spam;
- B es el número de documentos No Spam, clasificados como Spam;
- C es el número de documentos Spam, clasificados como No Spam;
- D es el número de documentos No Spam, clasificados como No Spam.

[1]Disponible en el ftp: *ftp://ftp.ics.uci.edu/pub/machine-learning-databases/spambase*

Partición P_i	SPAM	NO SPAM
Asigna SPAM	A	B
Asigna NO SPAM	C	D

Tabla 2: Tabla de contingencia.

A partir de estos datos se obtienen las métricas de **precisión** y **recall** y **F medida**, expresadas en las ecuaciones 5, 6 y 7 [12]:

$$P = \text{Precisión} = \frac{A}{A+B}$$

Ecuación 5: Precisión

$$R = \text{Recall} = \frac{A}{A+C}$$

Ecuación 6: Recall

$$F1 = \frac{2*P*R}{P+R}$$

Ecuación 7: Métrica F1

La tabla 3 muestra los resultados obtenidos en términos de precisión y recall. Se ha incluido también la métrica F1 (ver ecuación 7) con el fin de tener una valoración global del comportamiento de los algoritmos

	Precisión	Recall	F1
BBR	87,10%	88,93%	87,99%
SVM	80,52%	40,68%	54,05%
PLAUM	58,10%	67,52%	62,46%

Tabla 3: Resultados sobre la colección SPAMBASE

Como se puede observar el algoritmo BBR supera ampliamente a los otros dos algoritmos presentados. La mejora de precisión obtenida con el algoritmo BBR sobre los algoritmos SVM y PLAUM es de un 7,55% y 33,30%, respectivamente. En cuanto al recall, la mejora es aún mayor (54,26% sobre SVM y 24,08% sobre PLAUM).

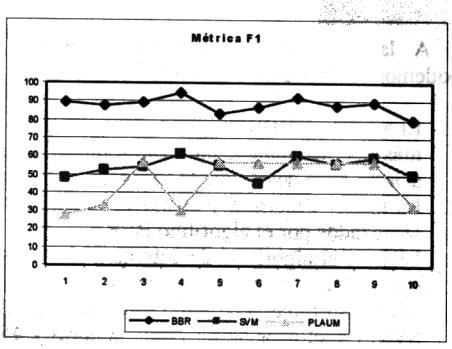

Figura 2. Comportamiento global de los algoritmos utilizando la métrica F1.

Desde un punto de vista global (es decir, utilizando la métrica F1), el algoritmo que presenta el mejor comportamiento es con diferencia el algoritmo BBR. La figura 2 muestra los valores obtenidos en cada uno de los 3 algoritmos para la métrica F1 en cada una de las 10 particiones utilizadas durante la evaluación. Se puede observar como en todos los casos (en cada una de las 10 particiones) la regresión logística bayesiana supera en más de 20 puntos a los otros dos algoritmos. Concretamente, promediando los resultados, el algoritmo BBR mejora el algoritmo SMV en un 38,57% y el algoritmo PLAUM en un 29,02%.

Por último, cabe destacar además que el algoritmo BBR es el más rápido de los algoritmos presentados tanto en el entrenamiento como en la evaluación.

4 Conclusiones y trabajo futuro

La tarea de filtrado de correo electrónico no deseado o Spam, aplicando métodos de categorización de texto y usando métodos de aprendizaje automático es una tarea muy interesante e importante actualmente, tanto desde el punto de vista de la investigación como desde una perspectiva comercial o empresarial.

Una cuestión importante a la hora de elegir el algoritmo más interesante para tratar el filtrado de Spam es determinar la estrategia más adecuada: filtrar el mayor número posible de correos válidos, o relajar las condiciones para dejar pasar más correos y evitar el rechazo de correos correctos.

También es importante el tiempo en el que se procesa y filtra toda la información y el sistema toma la decisión de clasificarlo como Spam o no.

A la vista de los resultados obtenidos, podemos adelantar las siguientes conclusiones:

1. El algoritmo BBR consigue unos resultados más que aceptables, en torno a un 90% de acierto en la detección de Spam.
2. La mayor Precisión y el mayor Recall son alcanzados por el algoritmo BBR.
3. BBR es el algoritmo de entrenamiento más rápido (en el peor de los casos, 102 seg.), mientras el más lento, con diferencia, es el SVM (en el peor de los casos, 2 horas y 45 minutos).

Podemos concluir que el método de Regresión Logística Bayesiana es muy adecuado para esta tarea de filtrado de Spam, ya que se obtienen unos resultados muy satisfactorios y en unos tiempos realmente interesantes.

De cara a futuros trabajos, planteamos la posibilidad de tratar estos algoritmos con otras colecciones disponibles, y la incorporación de otras técnicas de clasificación de texto, estableciendo un sistema de voto para determinar si un correo es o no spam, en función de la fiabilidad de cada uno de los algoritmos utilizados.

Agradecimientos

Este trabajo ha sido financiado con el proyecto (MCYT) TIC-2003-07158-C04-04.

Bibliografía

[1]. I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, and C.D. Spyropoulos, "An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-mail Messages". In Belkin, N.J., Ingwersen, P. and Leong, M.-K. (Eds.), Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000), Athens, Greece, pp. 160-167, 2000.

[2]. Kristin P. Bennet, Colin Campbell. Support Vector Machines: Hype or Hallelujah?. SIGKDD Explorations. ACM SIGKDD 2000.

[3]. Pai-Hsuen Chen, Chih-Jen Lin, and Bernard Schölkopf. A Tutorial on ν–Support Vector Machines. 2001.

[4]. Alexander Genkin, David D. Lewis, and David Madigan.Large-Scale Bayesian Logistic Regression for Text Categorization. Dimacs. 2004.

[5]. José María Gómez Hidalgo. Evaluating cost-sensitive unsolicited bulk email categorization. Proceedings of the ACM Symposium on Applied Computing. 2002.

[6]. José María Gómez Hidalgo, Enrique Puertas Sanz, Francisco Carrero García, Manuel de Buenaga Rodríguez. Categorización sensible al coste para el filtrado de contenidos inapropiados en Internet. Universidad Europea de Madrid. 2003.

[7]. T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[8]. Bryan Klimt, Yiming Yang. The Enron Corpus: A New Dataset for Email Classiffication Research. Proceedings of the 2004 European Conference on Machine Learning (ECML).

[9]. I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, George Paliouras, and C.D. Spyropoulos, "An Evaluation of Naive Bayesian Anti-Spam Filtering". In Potamias, G., Moustakis, V. and van Someren, M. (Eds.), Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML 2000), Barcelona, Spain, pp. 9-17, 2000.

[10]. SPAMBASE, creada por Mark Hopkins, Erik Reeber, George Forman, y Jaap Suermondt Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304. Donada por Donor: George Forman (gforman at nospam hpl.hp.com). Generado: Junio-Julio 1999.

[11]. S. E. Robertson, S. Walker, H. Zaragoza, R. Herbrich. Filtering track. Microsoft Cambridge at TREC 2002.

[12]. G. Salton, M.J. McGill. Introduction to modern informtion retrieval. McGraw-Hill 1983.

[13]. Stone, M.: Cross-validatory choice and assessment of statistical predictions (with discussion). Journal of the Royal Statistical Society B 36 1974.

[14]. Miguel Ángel García Cumbreras, L. Alfonso Ureña López, Fernando Martínez

Santiago y Arturo Montejo Raez: Búsqueda de Respuestas Multilingüe: Clasificación de preguntas en español basada en aprendizaje. Revista SEPLN nº 34. 2005.