# Clasificación de texto mediante atributos probabilísticos de coocurrencia de palabras.

J. Fernando Sánchez Vega

[1] *Coordinación de Ciencias Computacionales*
*Instituto Nacional de Astrofísica Óptica y Electrónica, INAOE.*
*e-mail: fer.callotl@ccc.inaoep.mx*
*Luis Enrique Erro #1, CP 72840, Sta. Ma. Tonantzintla, Puebla, México.*

Comentario [S1]:

**Resumen.** En este artículo se describe la implementación de un método de pesado para clasificación de documentos basado en la probabilidad condicional de la coocurrencia de las palabras obteniendo mejores resultados que el pesado binario convencional aplicado a la clasificación de los documentos obtenidos por una etapa previa de búsqueda de documentos relevantes en bancos de información.

**Palabras claves:** Clasificación de documentos, pesado de atributos, atributos probabilísticos.

## 1 Introducción

En las investigaciones científicas, es de vital importancia obtener la mayor cantidad de información acerca del estado del arte del tema de interés. Se necesita una gran cantidad de datos y para obtenerlos se han hecho métodos de búsqueda automática en las grandes bases de datos, el inconveniente de estos métodos de búsqueda es que arrojan una gran cantidad de artículos y no todos son relevantes. Lo que se propone en este documento es un método que sirva para la clasificación posterior de artículos en dos categorías: los que son verdaderamente relevantes y los que no lo son; con esto se obtienen resultados más refinados.

El proceso de selección es una actividad artesanal que lleva varias horas hombre para la obtención de un *corpus* razonable de artículos del tema. Es conveniente automatizar esta selección, pues el método artesanal es muy costoso. Para esto se plantea un clasificador de textos, concretamente de *abstract* de artículos; debido a que en algunos casos no se cuenta con el artículo completo pues éste se debe de comprar a la

editorial, y no se desea comprar artículos que no sean útiles para los grupos de investigación.

### 1.1 Aplicación concreta

En el centro de Ciencias Genómicas de la UNAM, en el departamento de programación genómica computacional se ha creado una base de datos llamada Regulón DB. En un principio el Regulón DB [1] contaba con más de 2000 artículos clasificados manualmente relacionados a los genes de regulación de la bacteria Escherichia coli (que es de particular interés para las investigaciones genómicas).

La técnica de búsqueda que se desarrolló se obtuvo por la extracción de términos clave de los documentos que se tenían en el Regulón DB; originalmente esta técnica de búsqueda es un conjunto de disyunciones y conjunciones de los términos extraídos en campos particulares de los documentos como: el título, el abstrac o en todo el documento. Luego ellos aplicaron esta técnica de búsqueda al banco Medline [2]; con lo que recuperaron 12672 artículos de los cuales 1823 se encontraban originalmente en el Regulón DB. Curadores profesionales están en el proceso de clasificación manual de los artículos recuperados y, hasta el momento en que se escribió este artículo, ellos reportan un incremento del Regulón DB a 3785 artículos.

Las técnicas que se propondrán en el presente documento son desarrolladas en particular para probarse en el Regulón DB y poder obtener una mayos cantidad de artículos relevantes, y a su vez poder valorar el método de búsqueda desarrollado por el departamento de programación genómica computacional de la UNAM

### 1.2 Trabajo desarrollado

La tarea propiamente desarrollada es la clasificación de textos; en estas técnicas los atributos que se utilizan para describir un documento son las palabras sus propias palabras. La forma de medir estos atributos normalmente son la aparición de la palabra, el número de apariciones en el documento o el número de apariciones en el documento relativo al *corpus*.

La propuesta es medir los atributos utilizando representaciones probabilísticas anteriormente empleadas en el entorno de análisis de información.

## 2. Trabajo Relacionado

En el artículo [3], en su análisis cualitativo y cuantitativo de la información de los artículos de biomedicina, proponen un método de extracción de las palabras claves que representen la información abordada. Se asigna un puntaje a las palabras que les permite saber su relevancia.

La medida de relevancia propuesta se llama incidencia total. La incidencia (formula 1) es una relación entre dos palabras; la incidencia de A sobre B es simplemente la probabilidad condicional de que la palabra A se encuentre en la oración dado que B si está. El puntaje que utilizan es la incidencia total (Formula 2); la incidencia total de la palabra A es la suma de las incidencias de A sobre cada una de las palabras del resto del vocabulario del artículo.

$$\mu_{\tilde{I}_W}\left(w_i, w_j\right) = \frac{\left|W_i \cap W_j\right|}{\left|W_i\right|}$$

Formula 1. Incidencia de W*i* a W*j*.

$$K_i = \sum_{j \neq i} \mu_{\tilde{I}_W}\left(w_i, w_j\right)$$

Formula 2. Incidencia total de W*i*.

Entonces, en [4], se reportan los resultados de la clasificación de documentos recuperados por la estrategia de búsqueda ECOLI utilizando el esquema de pesado binario, aplicando *stemming* y reducción de atributos por frecuencia y ganancia de información. El clasificador que presentó mejores resultados fue el Bayes multinominal, obteniendo una precisión para la clase positiva de 0.503 y recuerdo de 0.787 y para la clase negativa una precisión de 0.96 y un recuerdo de 0.869.

Por lo tanto, en este proyecto se propone utilizar el puntaje de la incidencia total de las palabras como el valor del atributo en un esquema de clasificación de texto con un modelo vectorial de bolsas de palabras como representación.

## 3 Metodología y Desarrollo

### 3.1 Preprocesamiento de los documentos del Regulón DB

Los documentos del Regulón DB contienen varios campos del PUBMED. Debido a que el método está basado únicamente en la extracción de información únicamente en el título y el *abstrac* de los artículos es necesario primero extraer estos campos de interés. Se eliminaron las palabras consideradas vacías tomadas de la conjunción de las listas de *Cornell SMART project* [6], [5] (utilizada en áreas medicas) y de Escobedo [4]. Posteriormente, se hizo la separación de las oraciones bajo los siguientes criterios:

1. El punto y coma separa las oraciones siempre (esto debido a que como el valor tiene cierta dependencia con el tamaño de la oración; y si un autor escribiera usando mucho el punto y coma y poco el punto podrían obtener valores mas alto, ya que no es de interés parametrizar el estilo, se ha optado esta decisión).
2. El punto separa las oraciones si:
   a) La siguiente palabra está en mayúscula (para que no separe con los puntos de las abreviaciones).
   b) No debe de haber un punto en las próximas 5 palabras (para que listas de iniciales o nombres con iniciales no separen y para que no existan oraciones muy pequeñas).
   c) No se dividen las oraciones en paréntesis con menos de 10 palabras.

Se separaron las palabras que estaban unidas por guiones debido a que había una gran cantidad de compuestas, ya que la medición de los atributos mantiene la relación entre las palabras que coocurren en un mismo enunciado. Por ello la información que nos pueden dar si estas palabras forman una palabra compuesta o se encuentran separadas no se verán mermadas.

**3.2 Instanciación.**

Se tomó como la lista de vocabulario las palabras presentes en los campos de interés con más de dos ocurrencias y se aplicó el algoritmo de instanciación (Tabla 1) presentado en la siguiente tabla para obtener el valor (K) de incidencia total para cada palabra.

| Algoritmo de instanciacion |
|---|
| 1. Se calcula el vector binario de cada oración.<br>1.1 En una Matriz de vocabulario X vocabulario se introduce el vector en todos las filas de los términos en el vocabulario<br>M[vector]=vector<br>1.1 Se suma la matriz de 2.1 a una matriz principal<br>MP[]=MP[]+M[]<br>2. Se obtiene un tiene el valor K de cada palabra sumando todos los valores en la fila excepto el de misma palabra de la fila, y se divide por el valor de la palabra de la fila.<br>$K[i]=\Sigma j \neq i\ MP[j] / MP[i]$ |

Tabla 1. Algoritmo de instanciacion

En el primer paso se obtiene el vector binario de los atributos en los documentos como se hace tradicionalmente pero sólo para una oración haciéndolo en cada oración del documento, con este vector se llena una matriz auxiliar de dimensiones del vocabulario de columnas por el vocabulario de filas y se introduce el vector de cada oración en las filas en las que el término de la columna haya aparecido en la oración. Posteriormente, esta matriz se suma a otra matriz que llevará el conteo de todo el documento. Al terminar de pasar por todas las oraciones del documento, es ahora que ya se puede obtener la incidencia total de todas las palabras; se obtiene el valor K de cada palabra sumando todos los valores en la fila i-esima excepto el valor de la columna i-esima, y se divide entre el valor de la columna i-esima y fila i-esima.

## 4 Experimentos y resultados

Los experimentos fueron realizados sobre los documentos recuperados con la estrategia de búsqueda ECOLI. Se tomaron como ejemplos positivos los 1823 documentos que recuperó ECOLI y que estaban en el Regulón DB, el resto de los documentos recuperados por el Regulón DB se tomaron como ejemplos negativos. En la etapa de clasificación se entrenaron a los clasificadores Bayes simple y Bayes multinominal con dos tercios de los ejemplos positivos y negativos; se probaron con el tercio restante.

En la tabla 2 se pueden ver los resultados de la clasificación si se usa el pesado binario muy común en las tareas de clasificación. En la tabla 3 se muestran los resultados del pesado de incidencia total. En general las precisiones y recuerdos con el pesado de incidencia son menores; en el caso del clasificador *Net Bayes* el pesado propuesto con el clasificador *Bayes Naive* obtiene mejor recuerdo que cuando se usa el pesado binario.

| Clasificador | Atributos | Matriz de confusion | Precision | Recall | Clase |
|---|---|---|---|---|---|
| BayesNet | 1314 | a b <-- class as<br>1259 564 \| a = 1<br>1461 9388 \| b = 0 | 0.463 | 0.691 | 1 |
| | | | 0.943 | 0.865 | 0 |
| Bayes Multinomial | 1314 | a b <-- class as<br>1446 377 \| a = 1<br>1753 9096 \| b = 0 | 0.452 | 0.793 | 1 |
| | | | 0.96 | 0.895 | 0 |
| Bayes Net | 1330 | a b <-- class as<br>1283 540 \| a = 1<br>1489 9360 \| b = 0 | 0.463 | 0.704 | 1 |
| | | | 0.945 | 0.863 | 0 |
| Bayes Multinomial | 1330 | a b <-- class as<br>1435 388 \| a = 1<br>1418 9431 \| b = 0 | 0.503 | 0.787 | 1 |
| | | | 0.96 | 0.869 | 0 |

Tabla 2. Resultados con pesado binario Tradicional

| Clasificador | Atributos | Matriz de confusion | Precision | Recall | Clase |
|---|---|---|---|---|---|
| Bayes Naive | 1201 | a b <-- class as<br>456 155 \| a = 1<br>801 2831 \| b = 0 | 0.363 | 0.746 | 1 |
| | | | 0.948 | 0.779 | 0 |
| Bayes Multinomial | 1201 | a b <-- class as<br>452 159 \| a = 1<br>761 2871 \| b = 0 | 0.373 | 0.74 | 1 |
| | | | 0.948 | 0.79 | 0 |
| Bayes Net | 1330 | a b <-- class as<br>409 202 \| a = 1<br>518 3114 \| b = 0 | 0.441 | 0.669 | 1 |
| | | | 0.939 | 0.857 | 0 |
| Bayes Naïve Complement | 1330 | a b <-- class as<br>456 155 \| a = 1<br>801 2831 \| b = 0 | 0.363 | 0.746 | 1 |
| | | | 0.948 | 0.779 | 0 |

Tabla 3. Resultados con el pesado propuesto de incidencia Total

## 5 Conclusiones y trabajo futuro

Se puede concluir que el margen de recuerdo y precisión se no incrementados en con la mayoría de los clasificadores, sin embargo es interesante que el en el caso de la clasificación con *Bayes Naive* se obtienen mejores recuerdo que el *Net Bayes* esto de puede deber a que el tipo de pesado ya tiene una representación que contempla las relaciones entre las palabras; En la figura 1 se puede ver que si una red bayesiana la raíz de la red y sus hijos son los atributos, todo esto seria el clasificador bayesiano (para el caso de esta figura es *Naive Bayes*) y el 3 nivel, las hojas, son las palabras. Es así que los atributos están intrínsecamente ligados a la probabilidad de las palabras.
Pero este incremento en el recuerdo de la clase positiva bien puede hacer que en la recuperación de artículos de nuestro interés perdamos menos información relevante no compensa el tiempo que tarda en ejecutarse la instanciación.

Como se pudo ver la selección de diferentes atributos pueden variar los resultados por eso se propone para obtener mejores resultados seguir probando nuevos atributos y hacer *clusters* de clasificadores con diferentes atributos

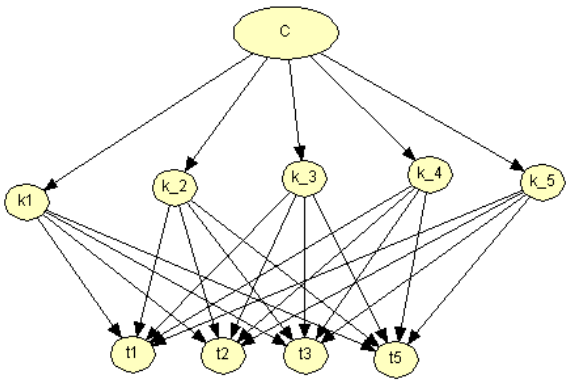


Figura 1. Clasificador Bayes simple con atributos k1, k_2,k_3, k_4… que dependen de que las palabras t1, t2, t3… aparezcan en el documento. Aquí se ve que si se el método completo asemeja una *Net Bayes*.

## Referencias

1. Proyecto UNAM: Analisis de Palabras Claves de de RegulonDB, Centro de Investigacion, sobre fijacón de Nitrogeno Laboratorio de Genomica Computacional. http://www.ccg.unam.mx/~proadmin/classification/index.html (sitio no público)
2. http://medlineplus.gov/

3. Shah P. et al. **Information extraction from full text scientific articles**: **where are the keywords**? *BMC Bioinformatics* (2003)
4. Escobar Acevedo, Adelina, Clasificador automático de textos para la base de datos Regulón DB.Curso Reconocimiento de patrones, INAOE, 2008.
5. ftp://ftp.cs.cornell.edu/pub/smart/english.stop.
6. A.B. Goldberg, D. Andrzejewski, J. Van Gael, B. Settles, X. Zhu, M. Craven, Ranking Biomedical Passages for Relevance and Diversity: University of Wisconsin, Madison at TREC Genomics 2006, University of Wisconsin, Madison