C.D.M Cuadernos de Documentación Multimedia
Vol. 15
Año 2004
ISSN: 1575-9733

Algunas Técnicas de Clasificación Automática de Documentos

Carlos G. Figuerola,

José L. Alonso Berrocal,

Ángel F. Zazo Rodríguez,

Emilio Rodríguez

INTRODUCCIÓN

La idea de clasificación es bien conocida por quienes se dedican a la documentación. Sin entrar en disquisiciones formales, se trata de organizar los documentos en alguna forma que permita después su mejor recuperación. En torno a ello se han elaborado diversas técnicas, que se han aplicado con mejor o peor fortuna. Con la creciente disponibilidad de documentos en formato electrónico, susceptibles, por consiguiente, de ser procesados de manera automática, surge la posibilidad de abordar la clasificación de documentos de manera automática. Este trabajo describe algunas de las técnicas y algoritmos aplicables en clasificación automática, los conceptos básicos en que se basan tales algoritmos, así como los instrumentos necesarios para aplicarlos. Del mismo modo, en la medida en que tales técnicas y algoritmos hayan sido aplicados, se ofrece una estimación del alcance y posibilidades de cada uno de ellos.

Clasificación supervisada y no supervisada

Por lo general, cuando se habla de clasificación automática se distingue entre dos escenarios diferentes, que obviamente, requieren soluciones distintas. Estos escenarios reciben diversos nombres, pero básicamente consisten en lo siguiente: de un lado, una situación en la que se parte de una serie de clases o categorías conceptuales prediseñadas a priori, y en la que labor del clasificador (manual o automático) es asignar cada documento a la clase o categoría que le corresponda.

En el segundo escenario posible, no hay categorías previas ni esquemas o cuadros de clasificación establecidos a priori. Los documentos se agrupan en función de ellos mismos, de su contenido; de alguna manera, podemos decir que se autorganizan. Es lo que se conoce como clasificación (automática) no supervisada o *clustering*; no supervisada porque se efectúa de forma totalmente automática, sin supervisión o sistencia manual [1].

El primer tipo o escenario se conoce como clasificación supervisada, no sólo porque requiere la elaboración manual o intelectual del cuadro o esquema de categorías, sino también, como se verá, porque requiere un proceso de aprendizaje o entrenamiento por parte del clasificador, que debe ser supervisado manualmente en mayor o menor medida.

CLASIFICACIÓN SUPERVISADA O CATEGORIZACIÓN

Como se ha dicho antes, en esta modalidad se parte de una serie de clases o categorías prediseñadas a priori, en las cuales hay que colocar a cada uno de los documentos. Conocida también como categorización, facilita la Recuperación ad-hoc, limitando las búsquedas a las

Vol. 15
Año 2004
ISSN: 1575-9733

clases o categorías que el usuario elige, basándose en el conicimiento intelectual que previamente tiene del tema; pero también se aplica en cosas como el filtrado automático de documentos, el clásico problema del *routing* de documentos o la navegación autónoma en el web.

Figure 1: Fig. 1 Clasificación Supervisada

Aunque hay gran cantidad de algoritmos capaces de hacer clasificación supervisada, la idea o enfoque básico es muy parecido: conseguir, de alguna manera, construir un patrón representativo de cada una de las clases o categorías; y aplicar alguna función que permita estimar el parecido o similitud entre el documento a clasificar y cada uno de los patrones de las categorías [6]. El patrón más parecido al documento es el que nos indica a qué clase debemos asignar ese documento.

Para la construcción de los patrones de las categorías se utilizan documentos clasificados manualmente de antemano, que sirven como ejemplo. El proceso de formar esos patrones de cada clase a partir de esos documentos preclasificados se conoce como *entrenamiento* o *aprendizaje*; y la colección de documentos preclasificados utilizada se conoce como colección de entrenamiento.

REPRESENTACIÓN DE DOCUMENTOS Y FUNCIONES DE SIMILITUD

Para poder llevar a cabo la clasificación automática es preciso contar con una serie de elementos previos. En nuestro caso, lo más importante es contar con una forma consistente de representar cada documento (su contenido), de manera que esa representación pueda generarse de forma automática. En este sentido, el formalismo más utilizado es el del bien conocido modelo vectorial, formulado por G. Salton ya en los años 70 [25] y ampliamente utilizado por sistemas de recuperación en la actualidad.

Básicamente, cada documento puede ser representado mediante un vector de términos. Cada término lleva asociado un coeficiente o *peso* que trata de expresar la importancia o grado de representatividad de ese término en ese vector o documento. Este peso puede calcularse de forma automática a partir de diversos elementos, basándose en las frecuencias de los términos, tanto en toda la colección de documentos con que se trabaje, como dentro de cada documento en particular.

Vol. 15
Año 2004
ISSN: 1575-9733

Sin entrar aquí en detalles técnicos, es obvio que un mismo término puede ser más o menos significativo en un contexto que en otro, de manera que tendrá diferente peso en un documento que en otro. Adicionalmente, dependiendo del ámbito de conocimiento en que se inscriba la colección documental, unos términos cobran más importancia que en otros; así, términos que aparecen en casi todos los documentos parecen poco aprovechables para recuperar documentos a partir de ellos. De otra parte, el tamaño o número de términos de cada documento también juega un papel importante. No parece lo mismo que un mismo término aparezca dos veces en un documento largo, de muchas páginas; a que aparezca también dos veces en un documento corto, de un par de párrafos.

Estos elementos son los que forman parte en el cálculo de los pesos; aunque hay multitud de ecuaciones propuestas para estimar dichos pesos, todas se basan, de una u otra forma en estos elementos, y es obvio que pueden ser obtenidos de manera automática; al igual que pueden extraerse automáticamente los términos que conforman un documento.

Independientemente del algoritmo de clasificación que se utilice, en algún momento será preciso estimar la cercanía, parecido o similitud entre dos documentos. Si hemos representado cada documento mediante un vector, podemos aplicar alguna de las muchas funciones de similitud disponibles para estimar la cercanía o parecido entre dos vectores. Existen multitud de funciones de similitud, propuestas ya incluso desde los años 50 [3, 4]. En realidad, trabajos experimentales han mostrado que los resultados obtenibles son muy parecidos con cualquiera de dichas funciones; lo cual se explica porque en realidad todas usan los mismos elementos de información [23]. En ocasiones, la diferencia hay que buscarla en la mayor o menor facilidad de implementación.

La más conocida y utilizada de éstas es la llamada función del coseno, en referencia al coseno de ángulo que forman dos vectores que se representan en un espacio multivectorial. Básicamente se basa en el producto entre ambos vectores, aplicando un factor de normalización para obviar los efectos de la disparidad en los tamaños (número de términos) de los vectores o documentos.

Otras, entre las más conocidas, son los coeficientes de Dice y de Jaccard [13] (pp. 152 y ss), así como el coeficiente de solapamiento [12] (pp. 125 y ss.)

Algoritmos más importantes

Hay una buena cantidad de algoritmos propuestos para este tipo de clasificación. La mayor parte de ellos no son, en realidad, especificos para clasificar documentos, sino que se han propuesto para clasificar todo tipo de cosas. Sucede que algunos de éstos han sido utilizados (con más o menos adaptaciones) para la clasificación de documentos.

Entre los más utilizados, tenemos:

- algoritmos probabilísticos
- algoritmo de Rocchio

Vol. 15
Año 2004
ISSN: 1575-9733

- algoritmo del vecino más próximo y variantes
- algoritmos basados en redes neuronales

Algoritmos probabilísticos

Se basan en la teoría probabilística, en especial en el teorema de Bayes. Éste permite estimar la probabilidad de un suceso a partir de la probabilidad de que ocurra otro suceso, del cual depende el primero.

De hecho, el algoritmo de este tipo más conocido, y también el más simple, es el denominado *naive Bayes*, propuesto desde hace bastante tiempo [17, 23] Básicamente, se trata de estimar la probabilidad de que un documento pertenezca a una categoría. Dicha pertenencia depende de la posesión de una serie de características, de cada una de las cuales conocemos la probabilidad de que aparezcan en los documentos que pertenecen a la categoría en cuestión.

Naturalmente, dichas características son los términos que conforman los documentos, y tanto su probabilidad de aparición en general, como la probabilidad de que aparezcan en los documentos de una determinada categoría, pueden obtenerse a partir de los documentos de entrenamiento; para ello se utilizan las frecuencias de aparición en la colección de entrenamiento.

Cuando las colecciones de aprendizaje son pequeñas, pueden producirse errores al estimar dichas probabilidades. Por ejemplo, cuando un determinado término no aparece nunca en esa colección de aprendizaje, pero aparece en los documentos a categorizar. Esto implica la necesidad de aplicar técnicas de *suavizado*, a fin de evitar distorsiones en la obtención de las probabilidades.

Con dichas probabilidades obtenidas de la colección de entrenamiento, podemos estimar la probabilidad de que un nuevo documento, dado que contiene un conjunto determinado de términos, pertenezca a cada una de las categorías. La más probable, obviamente, es a la que será asignado.

Como se ha apuntado más arriba, la implementación del *naive Bayes* es sencilla y rápida, y sus resultados son bastante buenos, como atestiguan numerosos trabajos experimentales: [19, 15, 28, 1].

Algoritmo de Rocchio

El llamado *algoritmo de Rocchio* [24] es bien conocido y aplicado en la realimentación de consultas. En este ámbito, la idea es simple: formulada y ejecutada una primera consulta, el usuario examina los documentos devueltos y determina cuáles le resultan relevantes y cuáles no. Con estos datos, el sistema genera automáticamente una nueva consulta, basándose en los documentos que el usuario señaló como relevantes o no relevantes. En este contexto, el algoritmo de Rocchio proporciona un sistema para construir el vector de la nueva consulta, recalculando los pesos de los términos de ésta y aplicando un coeficiente a los pesos de los la

Vol. 15
Año 2004
ISSN: 1575-9733

consulta inicial, otro a los de los documentos relevantes y otro distinto a los de los no relevantes.

En el ámbito de la categorización, el mismo algoritmo de Rocchio proporciona un sistema para construir los patrones de cada una de las clases o categorías de documentos. Así, partiendo de una colección de entrenamiento, categorizada manualmente de antemano, y aplicando el modelo vectorial, podemos construir vectores patrón para cada una de las clases, considerando como ejemplos positivos los documentos de entrenamiento de esa categoría, y como ejemplos negativos los de las demás categorías.

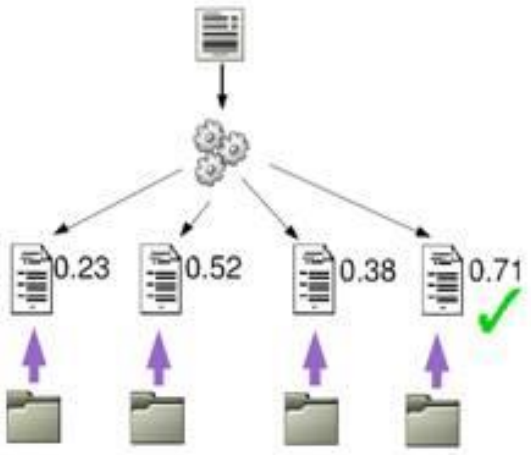

Figure 2: Fig. 2 El algoritmo de Rocchio es una forma de construir patrones de cada clase

Una vez que se tienen los patrones de cada una de las clases, el proceso de entrenamiento o aprendizaje está concluído. Para categorizar nuevos documentos, simplemente se estima la similitud entre el nuevo documento y cada uno de los patrones. El que arroja un índice mayor nos indica la categoría a la que se debe asignar ese documento.

El algoritmo de Rocchio ha sido utilizado en tareas de categorización con buenos resultados [14]. En [9] se realizaron experimentos con varios grupos de noticias de Internet, así como con noticias de la agencia de prensa Reuters, con excelentes resultados. En [2] se obtuvieron resultados con noticias de prensa en español comparables a los obtenidos por clasificadores humanos.

Algoritmo del vecino más próximo y variantes

El algoritmo del vecino más próximo (*Nearest Neighbour, NN*) es uno de los más sencillos de implementar. La idea básica es como sigue: si calculamos la similitud entre el documento a clasificar y cada uno de los documentos de entrenamiento, aqué de éstos mas parecido nos estará indicando a qué clase o categoría debemos asignar el documeto que deseamos clasificar.

Vol. 15
Año 2004
ISSN: 1575-9733

Desde un punto de vista más práctico, el *vecino más próximo* puede aplicarse con cualquier programa de recuperación de tipo *best macth*. Lo más frecuente es utilizar alguno basado en el modelo vectorial, pero, en puridad, esto no es imprescindible. Lo necesario es que sea *best macth* y no de comparación exacta, como pueda ser el caso de los booleanos; el algoritmo de basa en localizar el documento *más* similar o parecido al que se desea clasificar. Para esto no hay más que utilizar ese documento como si fuera una consulta sobre la colección de entrenamiento.

Una vez localizado el documento de entrenamiento más similar, dado que éstos han sido previamente categorizados manualmente, sabemos a qué categoría pertenece y, por ende, a qué categoría debemos asignar el documento que estamos clasficando.

Una de las variantes más conocidas de este algoritmo es la del *k-nearest neighbour* o *KNN* que consiste en tomar los *k* documentos más parecidos, en lugar de sólo el primero. Como en esos *k* documentos os habrá, presumiblente, de varias categorías, se suman los coeficientes de los de cada una de ellas. La que más puntos acumule, será la candidata idónea.

El *KNN* une a su sencillez una eficacia notable. Obsérvese que el proceso de entrenamiento no es más uqe la indización o descripción automática de los documentos, y que tanto dicho entrenamiento como la propia categorización pueden llevarse a cabo con instrumentos bien conocidos y disponibles para cualquiera. De otra parte, numerosas pruebas experimentales han mostrado su eficacia; éste es el caso de los experimentos publicados en [29] sobre diversas colecciones documentales. Parte de ellas estaban constituídas por noticias de prensa (un tipo de documento habitual en este tipo de experimentos, por otra parte), así como de referencias y resúmenes de literatura médica procedentes de MedLine [7].

KNN parece especialmente eficaz cuando el número de categorías posibles es alto, y cuando los documentos son heterogéneos y difusos. Esto parece confirmarse también en los trabajos de categorización de páginas *web* expuestos es [5].

Redes neuronales

Las redes neuronales en general han sido propuestas en numerosas ocasiones como intrumentos útiles para la Recuperación de Información y también para la clasificación automática. Coincidiendo con el auge general, en todos los campos, de las redes neuronales, especialmente en los primeros años 90, encontramos varias aplicaciones en el campo que nos interesa [21].

De una manera genérica, una de las principales aplicaciones de las redes neuronales es el reconocimiento de patrones. No es de extrañar, por tanto, que se hayan aplicado a problemas de categorización de documentos. Básicamente, una red neuronal consta de varias capas de unidades de procesamiento o neuronas interconectadas; en el ámbito que nos ocupa la capa de entrada recibe términos, mientras que las unidades o neuronas de la capa de salida mapea clases o categorías. Las interconexiones tienen pesos, es decir, un coeficiente que expresa la mayor o menor fuerza de la conexión. Es posible entrenar una red para que, dada una entrada

Vol. 15
Año 2004
ISSN: 1575-9733

determinada (los términos de un documento), produzca la salida deseada (la clase que corresponde a ese documento). El proceso de entrenamiento consta de un ajuste de los pesos de las interconexiones, a fin de que la salida sea la deseada.

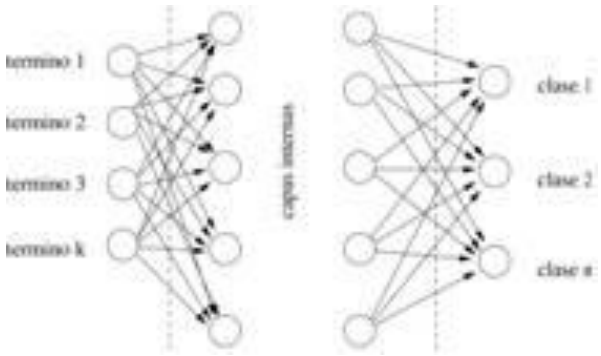

Figure 3: Fig.3 Red neuronal para clasificación automática

Diversos tipos de redes se han aplicado. Así, [26] utiliza una red de retropropagación aplicada al problema del *routing* o filtrado de documentos, algo muy parecido a la categorización; los resultados obtenidos son, según los autores, superiores a los obtenibles con el algoritmo de Rocchio. [22] presenta un experimento parecido, pro con páginas *web*.

Más habitual es el uso de los llamados Mapas Autorganizativos o SOMs (*Self Autoorganizing Maps*, de T. Kohonen [10]. Numerosos trabajos aplican este tipo de redes; probablemente uno de los primeros fue [16], en el que se clasificaron varios documentos técnicos basándose en términos de los títulos. Éste fue sguido algo más tarde por [18]. De mayor impacto ha sido el proyecto *WEBSOM* (`http://websom.hut.fi/websom/`), que ha generado una buena cantidad de literatura; por ejemplo [8, 11]. *WEBSOM* es básicamente una herramienta de organización automática de documentos y, sobre todo, de visualización de dicha organización.

CONCLUSIONES

La Clasificación Automática de documentos puede conseguirse mediante diversas técnicas. Se han revisado algunos de los algoritmos más frecuentes en clasificación supervisada de documentos; al margen de la mayor o menor complejidad de cada uno de ellos, parece que se trata de técnicas lo suficientemente maduras, cuyos resultados son equiparables a los conseguidos por clasificadores inteligentes, es decir, personas.

Algunas cuestiones, sin embargo, quedan por resolver; por ejemplo, las características idóneas de las colecciones de entrenamiento, tanto en tamaño como en distribución de las diferentes categorías. También, la mayor o menor capacidad de adaptación a cambios en los contenidos de las propias clases. Pensemos, por ejemplo, en un clasificador de noticias en distintas clases o secciones (deportes, economía, sucesos, etc.) entrenado en un momento determinado. Con el paso del tiempo las noticias, aún siendo de deportes, por ejemplo, contendrán términos en mayor o menor medida diferentes a los que aparecían en las noticias que se usaron para el

entrenamiento. Las necesidades, incluso la capacidad de captar los cambios en los documentos y clases, son cuestiones abiertas.

De otro lado, la aparición de tipos nuevos de documentos necesita otro tipo de respuestas. Así, los documentos multimedia difícilmente pueden representarse como vectores de términos, puesto que no consisten en texto. Las propias páginas *web*, por ejemplo, aunque siguen siendo documentos textuales en buena medida, incorporan otros elementos de información que es necesario recoger [20]; no sólo las imágenes o el sonido sino también los propios enlaces. Diversos trabajos se han centrado en esta cuestión, pero para que aún queda bastante por hacer.

REFERENCIAS

[1] Soumen Chakrabarti. *Mining the Web : discovering knowledge from hypertext data*. Morgan Kaufmann, San Francisco, CA, cop. 2003. ISBN: 1558607544.

[2] Carlos G. Figuerola, Ángel F. Zazo Rodríguez, and Jose Luis Alonso Berrocal. Automatic vs. manual categorization of documents in Spanish. *Journal of Documentation*, 57(6):763–773, 2001.

[3] L. Goodman and W. Kruskal. Measures of association for cross-classifications. *Journal of the American Statistical Association*, 49:732–764, 1954.

[4] L. Goodman and W. Kruskal. Measures of association for cross-classifications ii: Further discussions and references. *Journal of the American Statistical Association*, 54:123–163, 1959.

[5] Norbert Gövert, Mounia Lalmas, and Norbert Fuhr. A probabilistic description-oriented approach for categorising Web documents. In Susan Gauch and Il-Yeol Soong, editors, *Proceedings of the Eighth International Conference on Information and Knowledge Management*, pages 475–482, New York, 1999. ACM.

[6] J. He, A. Tan, C. Tan, and S. Sung. *On Quantitative Evaluation of Clustering Systems*, pages 105–134. In Wu et al. [27], 2003.

[7] W. Hersh. Oshumed: An interactive retrieval evaluation and large test collection for research. In *ACM SIGIR Conference on R&D in Information Retrieval*, pages 192–201, 1994.

[8] Timo Honkela, Samuel Kaski, Krista Lagus, and Teuvo Kohonen. WEBSOM—self-organizing maps of document collections. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pages 310–315. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1997.

[9] Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 143–151, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[10] Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko Saloj¨arvi, Jukka Hankela, Vesa Pastero, and Antti Saarela. Self organization of a massive document collection. *Neural Networks*, 11(3):574–585, 2000.

[11] Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko Salojärvi, Vesa Paatero, and Antti Saarela. Organization of a massive document collection. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, 11(3):574–585, May 2000.

[12] Richard R. Korfhage. *Information Storage and Retrieval*. Wilet Computer Publishing, 1997.

[13] Gerald Kowalski. *Information Retrieval Systems. Theory and Implementation*. Kluwer Academic Publishers, 1998.

[14] David D. Lewis, Robert E. Schapire, James P. Callan, and Ron Papka. Training algorithms for linear text classifiers. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 18–22, 1996, Zurich, Switzerland (Special Issue of the SIGIR Forum)*, pages 298–306. ACM, 1996.

[15] Davis D. Lewis and Marc Ringuette. A comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, US, 1994.

[16] Xia Lin, Dagobert Soergel, and Gary Marchionini. A self-organizing semantic map for information retrieval. In Abraham Bookstein, Yves Chiaramella, Gerard Salton, and Vijay V. Raghavan, editors, *SIGIR*, pages 262–269. ACM, 1991.

[17] M. Maron. Automatic indexing: an experimental inquiry. *Journal of the ACM*, 8:404–417, 1961.

[18] Dieter Merkl. A connectionist view on document classification. In *Australasian Database Conference*, pages 0–, 1995.

[19] Marie-Francine Moens and Jos Dumortier. Automatic categorization of magazine articles. In P. de Bra and L. Hardman, editors, *Conferentie Informatiewetenschap 1999*, Amsterdam, 1999.

[20] S. Noel, V. Raghavan, and C. H. Chu. *Document Clustering, visualization and retrieval via link mining*, pages 161–194. In Wu et al. [27], 2003.

[21] Douglas W. Oard. Neural networks in information filtering and retrieval, 1994.

[22] C. Orwig, R. Chen, and H. Schuffels. Internet categorization and search:a machine learning approach. *Journal of Visual Communications and Image Representation*, 1(7):88–102, 1996b.

[23] C.J. van Rijsbergen. *Information retrieval*. 1979.

Vol. 15
Año 2004
ISSN: 1575-9733

[24] J. J. Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System. Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, Englewoods Cliffs, N. J., 1971.

[25] Gerard Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communication of the ACM*, 18:613–620, 1975.

[26] H. Schutze, David A. Hull, and Jan Q. Pedersen. A comparasion of classifers and document representations for the routing problem. *SIGIR 95*, 1995.

[27] Weili Wu, Hui Xiong, and Shashi Shekhar, editors. *Clustering and Information Retrieval*. Kluwer, 2003.

[28] Y. Yang and Xiu Liu. A re-examination of text categorization methods. In *22nd Annual International SIGIR*, pages 42–49, Berkley, August 1999.

[29] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90, 1999.