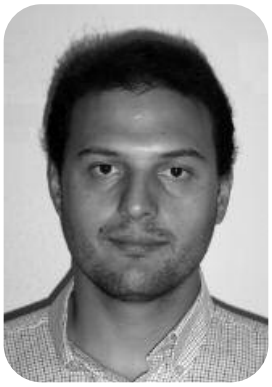
Clasificadores inductivos para el posicionamiento web

Por **Francisco José Soltero Domingo** y **Diego José Bodas Sagi**

Francisco José Soltero Domingo, licenciado en ciencias físicas por la UNED (1996). Máster en multimedia y vídeo interactivo y doctorando en la UC3M.
En la actualidad es profesor de ingeniería técnica informática de sistemas en el CES Felipe II (Aranjuez, Madrid).

Resumen: *En este trabajo se muestra cómo el estudio individual de los distintos atributos básicos de un recurso web no es suficiente para inferir las distintas estrategias de posicionamiento de un motor de búsqueda. El problema fundamental que se plantea es cuál es la relación entre los distintos elementos que componen la página y el peso que cada uno de ellos aporta al posicionamiento final. Como alternativa a este problema se propone la utilización de técnicas de aprendizaje inductivo, más concretamente, clasificadores arbóreos. Los resultados se ven reflejados en dos experimentos, fruto de la aplicación de dos algoritmos de aprendizaje distintos. Como resultado final se observa que la aplicación de esta técnica puede ser un punto de partida muy interesante para la optimización del posicionamiento web.*

Palabras clave: *Ranking web, Clasificadores inductivos, Optimización de recursos web, Posicionamiento web.*



Diego José Bodas Sagi, licenciado en ciencias matemáticas (UCM) en 1998. Máster en administración y dirección de empresas (UNED), y experto universitario en sistemas de gestión de bases de datos (UNED). Es profesor de ingeniería técnica informática de sistemas en el CES Felipe II (Aranjuez, Madrid).

Title: Inductive sort keys for web positioning

Abstract: *This paper shows how the individual study of different basic attributes from a web resource is not enough to infer different positioning strategies for search engines. The fundamental question is about the relation between the different elements that compose the page and the importance that each of them contributes to the final positioning. As an alternative to this problem, we propose the use of inductive learning techniques, more concretely, sort trees. Results are reflected in two experiments, the outcome of two different learning algorithms. As a final result, we conclude that the application of this technique can be a very promising starting point for optimising web positioning.*

Keywords: *Web ranking, Optimisation of web resources, Positioning with decision trees*

***Soltero Domingo, Francisco José; Bodas Sagi, Diego José**. "Clasificadores inductivos para el posicionamiento web". En: El profesional de la información, 2005, enero-febrero, v. 14, n. 1, pp. 4-13.*

Introducción

En la actualidad, la Red se ha convertido en la biblioteca global del conocimiento humano. Los motores de búsqueda o buscadores son una de las herramientas básicas que posibilitan la localización de información relevante dentro del contenido disponible[1]. El más popular es sin duda *Google*, al que le siguen otros como *Yahoo search*, *MSN*, *AskJeeves*, etc.

En los albores de estos buscadores la principal prioridad se centraba en guardar los datos de la forma más eficiente posible. Por ello las primeras investigaciones se concentraron en el estudio de la estructura física de los ficheros y en sus distintos tipos de indización. Con el paso del tiempo los datos almacenados han pasado de cientos de miles a miles de millones, si bien la mejora de los elementos de hardware y la estructura de indización de palabras y búsqueda de datos a través de listas invertidas han permitido que las consultas respondan de manera resuelta. En estos momentos, para una consulta típica, se pueden localizar treinta mil recursos, lo cual conlleva que un usuario podría necesitar toda una vida para poder leer las referencias documentales de todos los documentos enviados por el buscador en una única consulta. Por tanto, es necesario ordenar y filtrar los datos por algún orden de rele-

Artículo recibido el 25-09-04
Aceptación definitiva: 04-12-04

vancia. Y en este punto comienza unos de los grandes 'misterios' de estos motores: el criterio seguido para establecer la relevancia de los datos y, por tanto, de los recursos.

Existen varios factores que no debemos olvidar: las organizaciones encargadas del desarrollo y mantenimiento de estos buscadores son empresas privadas que cotizan en bolsa, por lo que su fin más importante es la rentabilidad económica. Además, la mayor parte de los recursos localizados en estos motores de búsqueda son páginas corporativas de entidades que ofrecen sus servicios a través de internet, y cuyo negocio se ve afectado por su posicionamiento entre los primeros resultados, sobre las consultas realizadas por sus potenciales clientes. Como podemos intuir, la determinación de los criterios de relevancia a la hora de establecer el posicionamiento de un motor de búsqueda se engloba más en el ámbito de los secretos comerciales que en la investigación científica, a pesar de los miles de artículos que se escriben sobre el tema.

«El objetivo de esta investigación es obtener un método que sea capaz de deducir los criterios que utilizan los motores de búsqueda»

El objetivo de este trabajo ha sido llevar a cabo un método que sea capaz de deducir los criterios que utilizan esas herramientas para establecer si un recurso se encontrará entre los veinte primeros lugares de todos los resultados devueltos. Como casos de estudio se emplean las consultas en el motor de búsqueda de *Google*. El conjunto de métodos y técnicas empleadas para el posicionamiento web podría constituir el embrión de una futura metodología de posicionamiento que dote al usuario de mejores criterios para evaluar las búsquedas efectuadas.

El resto del artículo esta dividido en varios apartados. El primero trata de la obtención del modelo de datos, para posteriormente realizar un análisis clásico de los resultados. Después veremos la utilización de clasificadores inductivos arbóreos y finalizaremos con las conclusiones y la bibliografía.

2. Obtención del modelo de datos

Para la realización del estudio se va a establecer una serie de criterios iniciales. El primero de ellos es que las consultas se realizarán en lengua inglesa. La indización de los buscadores se realiza por palabras de un lenguaje concreto, y dado que la mayor parte de los buscadores y, por tanto, de las bases de datos se encuentran en inglés, se ha establecido que éste es el idioma más adecuado para el estudio. De cualquier forma los resultados obtenidos en este trabajo no están influenciados por el lenguaje de la consulta. En cuanto al sitio de *Google* seleccionado será el del Reino Unido, como consecuencia directa del idioma escogido.

http://www.google.co.uk

Los temas de las consultas se establecerán en un entorno comercial. La selección de las claves de búsqueda es una de las labores más importantes, ya que en su mayor parte las páginas están construidas a partir de las mismas. Para ayudarnos en este propósito utilizaremos una serie de programas específicos denominados "ad word analyzer"[9]. Concretamente vamos a emplear dos programas gratuitos: *Espotting keyword generator*[2] y el generador de claves del propio *Google*. Para obtener este segundo necesitamos abrir previamente una cuenta online para a*dword*; una vez hecho, nos aparecerá un generador aplicable a la búsqueda de palabras clave para múltiples países y en unos treinta idiomas.

Si emprendemos la labor de optimizar una página desde cero es necesario tener la visión que el propio motor tiene de ésta, ya que si editamos su código fuente tendríamos la visión que el usuario tiene de la misma, que no coincide en absoluto con la captada por el *spider* (también denominado araña o robot de búsqueda de recursos) del buscador. Existen herramientas que simulan a estos robots (s*earch engine spider simulator*)[3] que nos pueden ayudar en esta tarea. En nuestro estudio hemos supuesto que las páginas ya han sido optimizadas previamente por el usuario y parte de esta optimización ha sido la que le ha servido al buscador para llevarlas a los primeros lugares del ranking. Por tanto, con la edición y posterior análisis del código fuente es suficiente.

Otro elemento relevante es la selección de los atributos adecuados; hemos optado por 30. Su selección está basada en las recomendaciones hechas por la mayor parte de los optimizadores de páginas web.

Los componentes seleccionados son los siguientes:

—Elementos de calidad de la página: *PageRank*, tamaño html (KB).

—Etiqueta *Meta description*: frecuencia, prominencia y peso de la clave.

—Etiqueta *Meta keywords*: frecuencia, prominencia y peso de la clave.

—Título: frecuencia, prominencia y peso de la clave.

—Cuerpo del documento (*Body*): número de palabras, palabras clave en negrita y subrayadas. Frecuencia, prominencia y peso de la clave. Clave al principio y al final del documento.

—Cabeceras (*H1-H6 Head*): frecuencia y prominencia y peso de la clave en la primera cabecera. Frecuencia y peso de la clave en todas las cabeceras.

—Enlaces (*Links*): frecuencia y peso de la clave (texto y etiqueta *Alt*).

—Atributos *Alt* de la etiqueta <*img*>: frecuencia y peso de la clave.

—Comentarios: frecuencia y peso de la clave.

Definiciones: La frecuencia es el número de veces que la clave es utilizada en un área concreta; el peso es el cálculo del número de palabras en un área multiplicado por la frecuencia y dividido por el número total de palabras; la prominencia viene dada por el orden de aparición de la clave de búsqueda en el área analizada.

Una vez seleccionados los distintos parámetros que compondrán nuestro modelo, iniciamos su simulación con la obtención de los datos correspondientes a las dos primeras páginas devueltas. En este caso, la selección se enmarca en el hecho de que las estadísticas demuestran que los usuarios sólo leen las primeras páginas de resultados. Por tanto la muestra final estará compuesta de 3 consultas con los 20 primeros resultados devueltos por el motor de búsqueda *Google*.

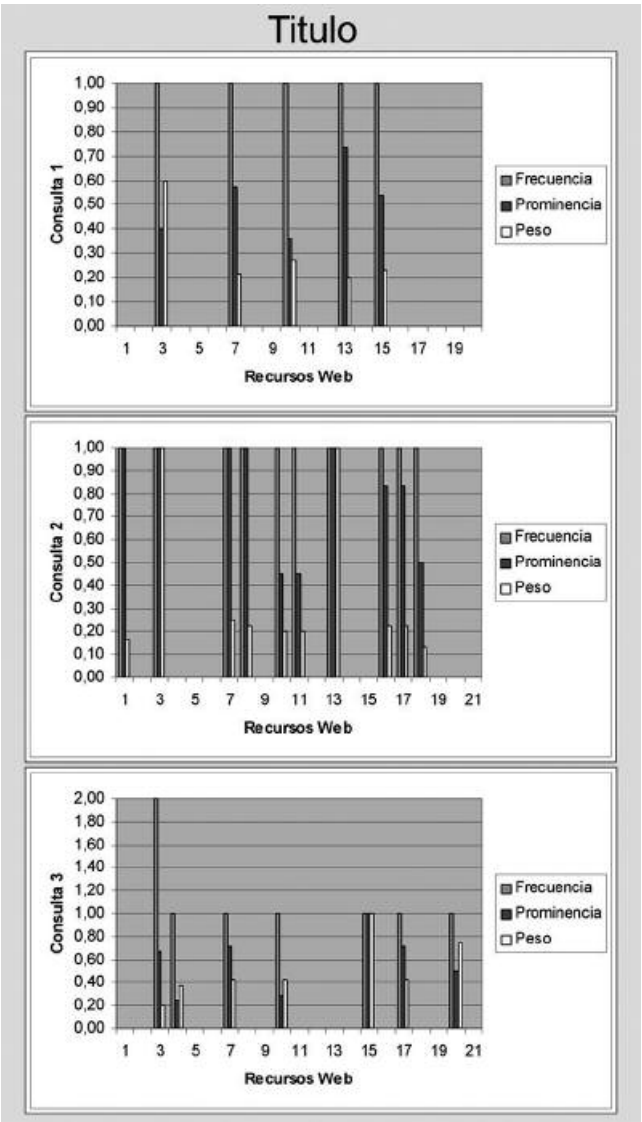


Figura 1. Palabra clave en el título

3. Análisis clásico de resultados

Se encuentra dividido principalmente bajo dos cuestiones: el primero es el factor *on the page* o elementos sobre la página, que se encargaría del análisis de los elementos que la componen: textos, etiquetas, estructura del web, navegación, enlaces, partes con contenido dinámico, texto de los enlaces, etc. En cuanto al segundo, denominado *off the page*, o factor externo a la página, se englobaría en el estudio de la cantidad y calidad de los enlaces desde otras páginas, así como al número de visitas recibidas, entre otros.

3.1. Resultados sobre el factor (*on the page*)

El tamaño de los recursos obtenidos en la consulta en ningún caso supera los 30 KB. Por otra parte, en cuanto a los atributos de frecuencia, prominencia y peso de la clave en el título, podemos observar (figura 1) que es bastante desigual: sólo en el 50% de las páginas se encuentra un título que coincide con la clave de búsqueda.

«Las organizaciones encargadas de la creación y mantenimiento de los buscadores son empresas privadas que cotizan en bolsa»

Otra característica observada en los documentos retornados, aunque no forma parte de nuestros atributos, es la falta de uso de marcos (frames) que es inexistente en todos los resultados, posiblemente debido a que la mayoría de los buscadores encuentran dificultades con ellos (aunque en teoría *Google* no tiene problemas a la hora de procesarlos). En cuanto al uso de lenguajes de programación, aparece *Javascript* en algunas páginas, mientras que *Visual basic script* no se encuentra en ninguna.

Las etiquetas *Meta description* (figura 2) y *Meta keyword* (figura 3) también son utilizadas por algunas páginas, aunque la mayoría de los recursos no las utilizan. Esto puede ser debido a que entre los webmasters existe la creencia de que algunos buscadores ya no las utilizan para posicionar, y por tanto prefieren no introducirlas. Tampoco existen redirecciones a otras páginas debido a las penalizaciones en el ranking de posicionamiento.

Las variables referentes al cuerpo del documento, frecuencia, prominencia y peso (figura 4) de la clave

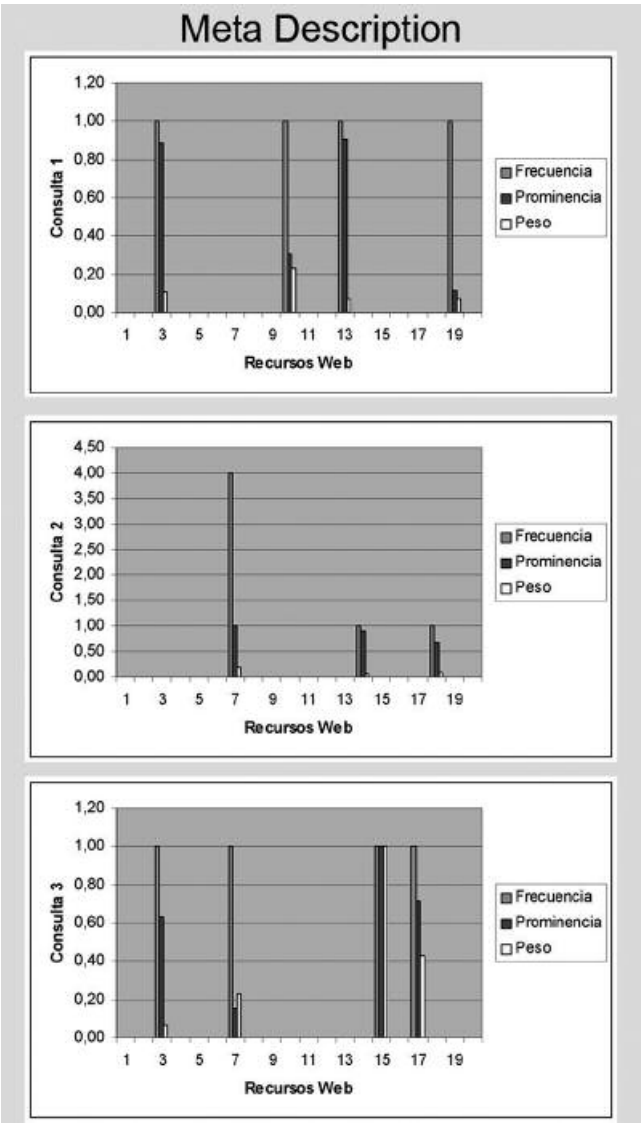


Figura 2. Atributos de la etiqueta Meta description

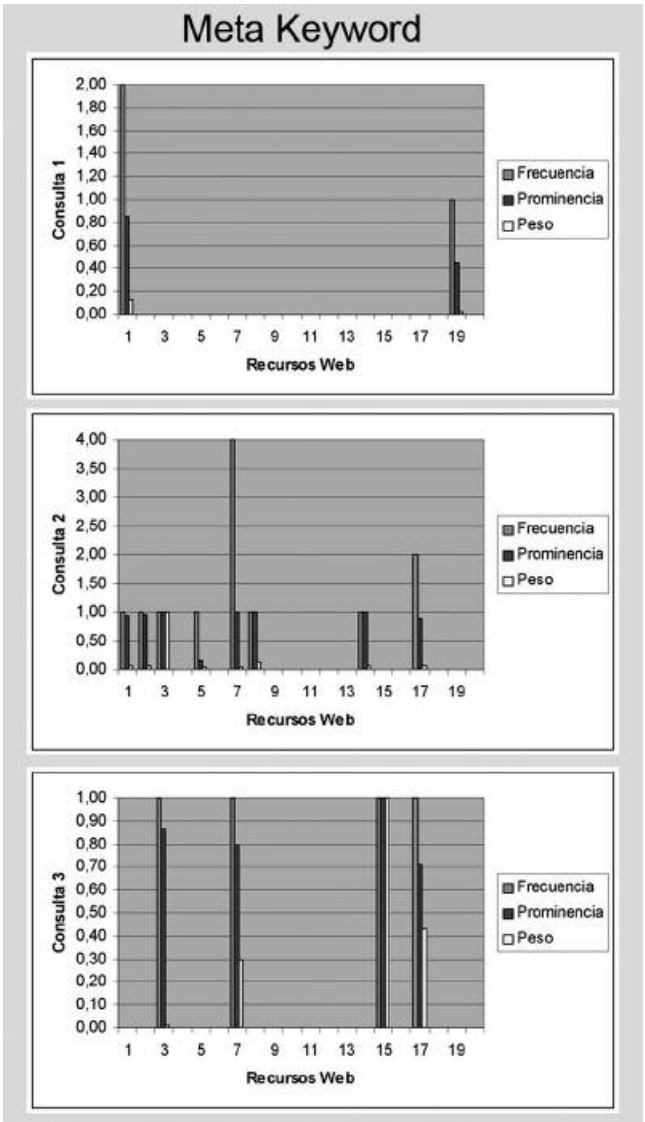


Figura 3. Atributos de la etiqueta Meta keyword

principal o clave de búsqueda en el cuerpo del documento son bastante desiguales; existen páginas donde es elevado, pero también se detectan otras en las que es pequeño o no aparece.

En lo que respecta al texto enlazado, etiquetas *Alt* y comentarios, no parecen ser especialmente relevantes, puesto que en los resultados apenas son influyentes. De hecho, la mayor parte de las páginas recuperadas no poseen la clave de búsqueda en ninguno de estos elementos.

3.2. Factores externos a la página (*off the page*)

Este tipo de optimización ya no corresponde a la página propiamente dicha, sino más bien a los conocimientos estratégicos sobre cómo conseguir los enlaces, cuáles interesan más, cómo se deben editar, etc.

Para este estudio vamos a tomar el *PageRank* que es un valor numérico que asigna *Google* a todas sus páginas y que representa su importancia en internet[10]. Cuando una página enlaza con otra, ésta le asigna su voto ponderado, que se establece en función de la importancia de la página que lo emite. El *PageRank* es un dato valioso ya que determina la calidad de una página para *Google*, basado en el número de enlaces de calidad que posee.

Para poder mostrar este parámetro con mayor profundidad hemos tomado una muestra más grande, compuesta por los veinte primeros recursos y otros veinte más localizados entre los cien primeros para cada una de las consultas.

Como podemos observar (figura 6) existen muchos recursos que tienen valores de *PageRank* elevados y que no se muestran entre los veinte primeros resultados.

3.3. Resumen del análisis clásico

Como conclusiones iniciales podemos observar el hecho de que existen elementos o atributos que determinan el posicionamiento de las páginas. Analizados estos datos de manera parcial, como hemos realizado

en este apartado, no podemos concluir ningún resultado de especial relevancia. De hecho la mayor parte de los *SEO* (abreviatura en inglés de «optimizador de motor de búsqueda") nos indican que optimicemos todos estos elementos, divididos en los factores expuestos anteriormente, pero en realidad desconocen la influencia real de los factores e incluso de los atributos. Todo esto contribuye a la introducción de 'ruido' o penalizaciones en las páginas.

Conocer la relación entre los atributos, el peso de los mismos, y su variabilidad en el tiempo son los factores que debemos determinar para conseguir los criterios de selección del motor de búsqueda en la obtención del posicionamiento final de los recursos web. Un análisis clásico ha resultado infructuoso en la consecución de resultados determinantes.

Por tanto, vamos a ampliar el estudio con métodos de aprendizaje automático. El objetivo consiste en comprobar si a partir de los atributos elegidos podemos determinar correctamente qué páginas se encuentran entre las veinte primeras y cuáles no, para las consultas seleccionadas.

Para tratar de profundizar más en los resultados se va a aumentar la muestra con una nueva consulta. Ahora en cada búsqueda, al igual que se hizo con el estudio del *PageRank*, se tendrán en cuenta cuarenta recursos (con un valor final de la muestra de ciento sesenta). Para cada consulta se han tomado veinte muestras correspondientes a las veinte primeras páginas. El resto de muestras, otras veinte, se encuentran entre las posiciones veinte y cien obtenidas por cada consulta en el buscador.

4. Utilización de clasificadores inductivos arbóreos

El empleo de técnicas de aprendizaje automático para la obtención de patrones sobre grandes bases de datos se denomina minería de datos (*data mining*). Dentro de las distintas formas de aprendizaje, el tipo de realimentación disponible es el elemento más importante a la hora de determinar la naturaleza del pro-

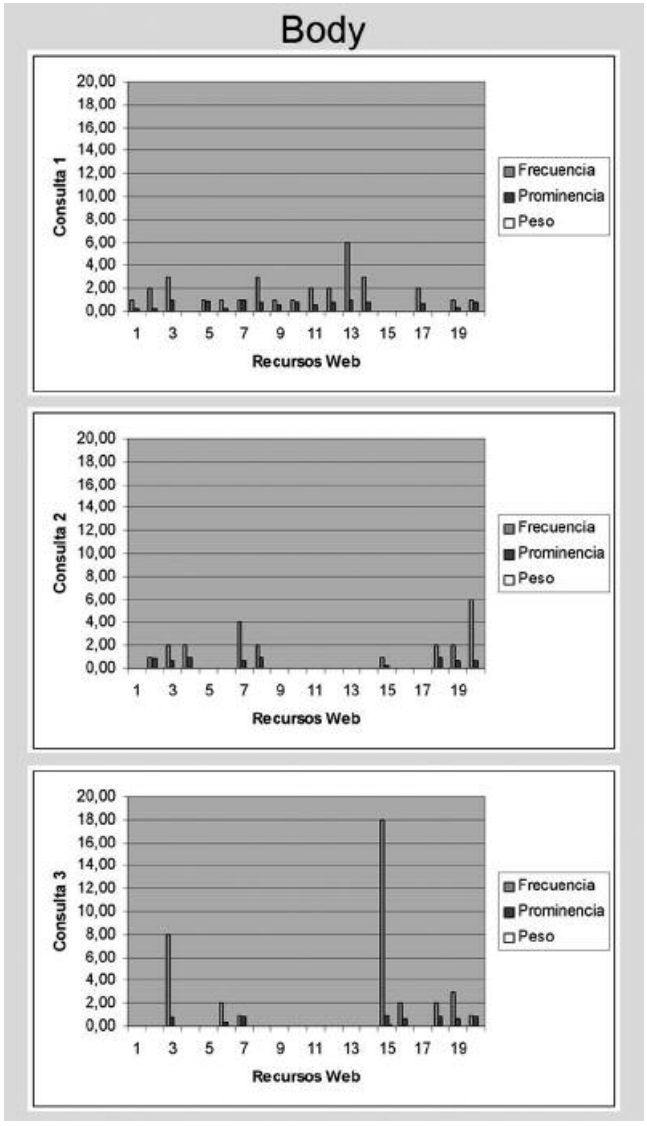


Figura 4. Atributos referentes al cuerpo del documento

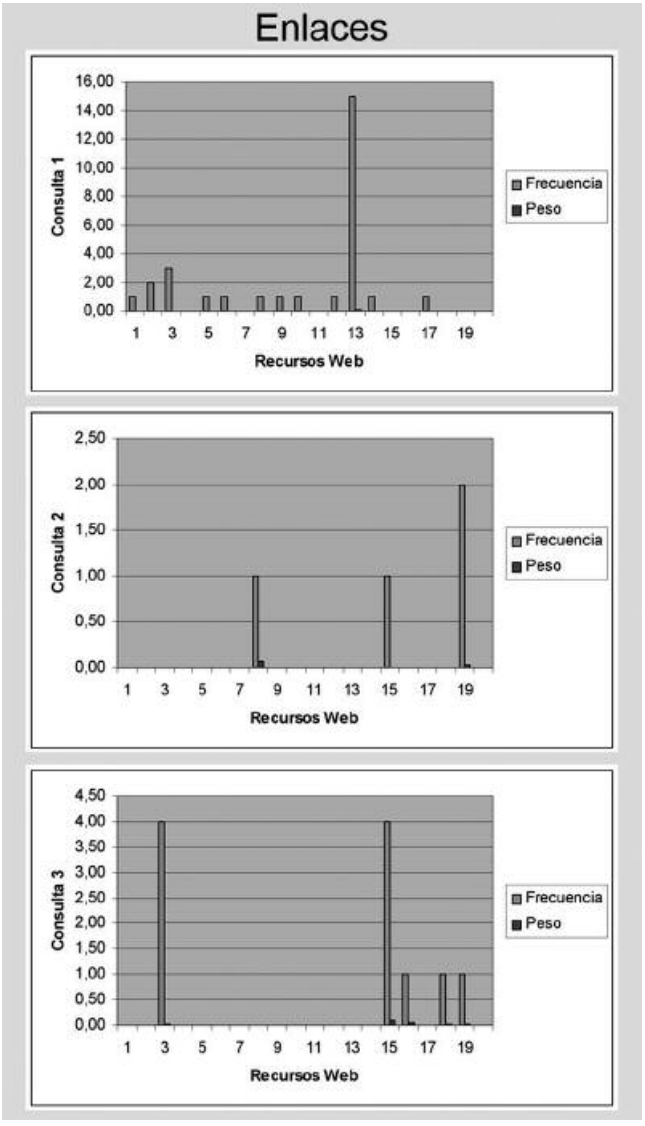


Figura 5. Frecuencia y peso de la clave de búsqueda en los enlaces

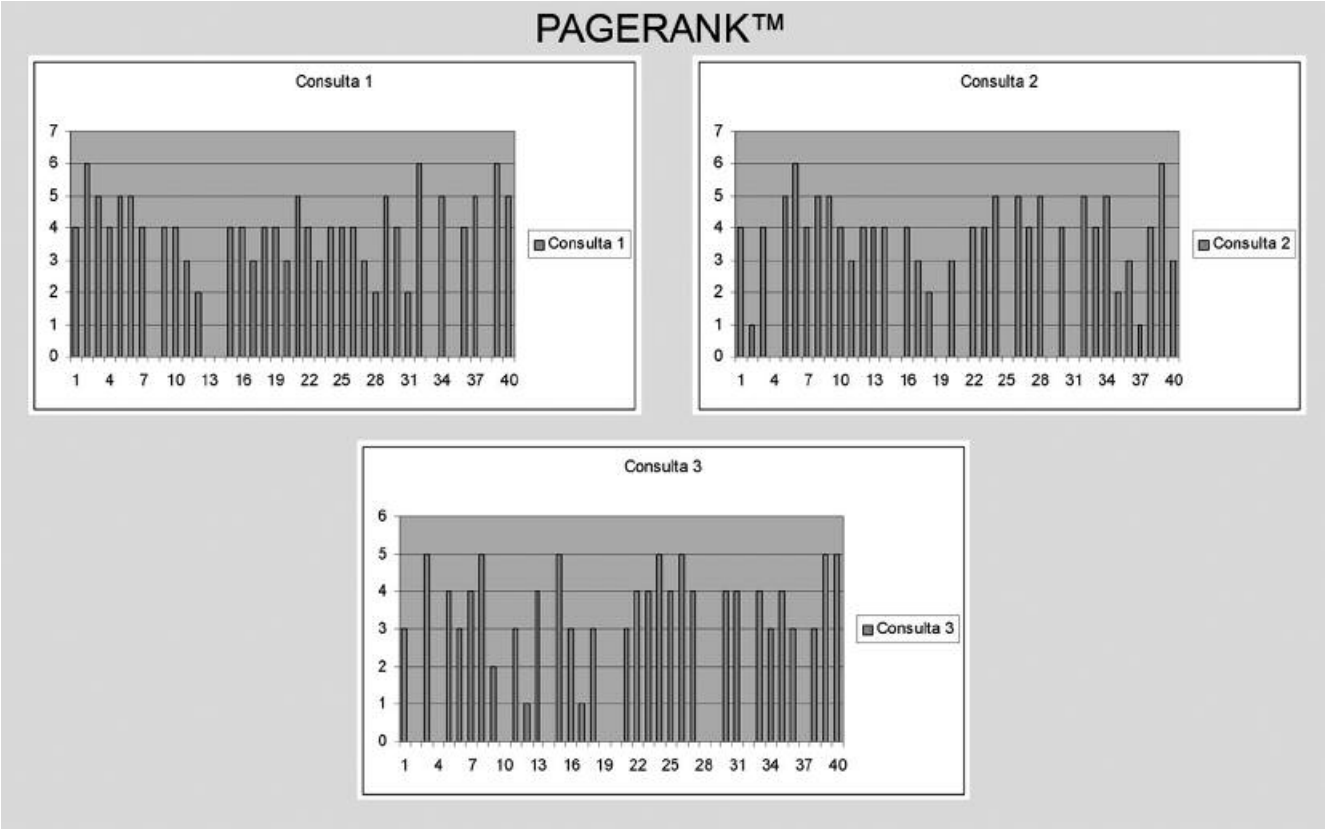


Figura 6. PageRank

blema de aprendizaje que se debe afrontar[5]. Se distinguen tres tipos de aprendizaje: supervisado, no supervisado y por refuerzo. En particular, el primero (también denominado inductivo) consiste en aprender una función a partir de ejemplos de sus entradas y salidas; es decir, recibe como entrada el valor correcto para determinados valores de una función desconocida y debe averiguar cuál es la función o aproximarla.

«En un futuro, y basándonos en los experimentos realizados, podríamos construir una aplicación cuyo parámetro de entrada sea un recurso web que genere como salida los valores de aquellos elementos que deban ser mejorados»

La inducción con árboles de decisión es una de las técnicas más empleadas debido a su sencillez de implementación. Un árbol de decisión toma como entrada un objeto o una situación descrita a través de un conjunto de atributos y devuelve una decisión: el valor previsto para una salida con respecto a una entrada dada. Los atributos de entrada pueden ser discretos o continuos. En el caso de funciones con valores discretos nos referimos a clasificación, mientras que la denominamos regresión cuando son valores continuos. En nuestro estudio nos centraremos en clasificaciones del tipo booleano, las cuales se clasifican como verdadero (páginas que se encuentran entre las 20 primeras) o falso (no se encuentran entre las 20 primeras). Un árbol desarrolla una serie de pruebas antes de alcanzar la solución. En cada uno de sus nodos internos se encuentra el valor de una de las propiedades y las ramas que salen del nodo están etiquetadas con los posibles valores de dicha propiedad. En los nodos hojas se encuentra el valor resultante si dicha hoja es alcanzada.

En el apartado anterior indicamos como factores fundamentales la necesidad de conocer la relación entre los atributos, al igual que el peso de los mismos. Como podemos observar, el aprendizaje inductivo (y más concretamente los árboles de clasificación) nos puede ayudar en esta tarea. En el caso del primero, los distintos nodos se posicionan en ramas, las cuales poseen atributos que están relacionados entre sí. Esta relación nos puede llevar incluso a descartar algunos atributos que, a pesar de encontrarse en la misma rama, no son necesarios ya que los ejemplos pueden estar bien separados en positivos y negativos, sin necesidad de agotar la totalidad de atributos para esa rama. En este caso estamos relacionando atributos al igual que eliminamos otros que no aportan nada y que por tanto el motor de búsqueda descarta a la hora de realizar la clasificación. En cuanto al peso, nos lo da la posición de los distintos atributos a lo largo del árbol: uno que se encuentre en un nivel más alto del árbol tendrá un peso mayor que otro situado en los últimos

niveles. Otro elemento que se mencionaba como importante es el de la variabilidad, la construcción de árboles de decisión es relativamente sencilla y por tanto se podrían implementar con cierta facilidad.

«El segundo experimento, con una tasa de acierto del 97,5%, indica que la hipótesis propuesta es válida»

Dentro de los distintos árboles de clasificación se ha realizado un estudio inicial de cuáles se adaptan mejor a nuestro estudio y, hemos seleccionado dos: el clasificador *LMT* (*Logistic Model Tree*) y el *Random Forest*.

4.1. Clasificador *LMT*[6]

Dos populares métodos de clasificación son la regresión logística lineal y los árboles de inducción, para los cuales existe una gran complementariedad. La combinación de ambos permite una estructura de árbol con modelos de regresión logística en las hojas. La gran ventaja de este enfoque es que las estimaciones explícitas de probabilidad de la clase son mejores que las de una simple clasificación.

4.2. Clasificador *Random Forest*

Se basan en el desarrollo de muchos árboles de clasificación[7]. Para clasificar un nuevo objeto desde un vector de entrada, ponemos dicho vector bajo cada uno de los árboles del bosque. Cada árbol genera una clasificación, en términos coloquiales diríamos que cada árbol vota por una clase. El bosque escoge la clasificación teniendo en cuenta el árbol más votado sobre todos los del bosque.

Cada árbol se desarrolla como sigue:

—Si el número de casos en el conjunto de entrenamiento es *N*, prueba *N* casos aleatoriamente, pero con sustitución, de los datos originales. Este será el conjunto de entrenamiento para el desarrollo del árbol.

—Si hay *M* variables de entrada, un número $m<<M$ es especificado para cada nodo, *m* variables son seleccionadas aleatoriamente del conjunto *M* y la mejor partición de este *m* es usada para dividir el nodo. El valor de *m* se mantiene constante durante el crecimiento del bosque.

—Cada árbol crece de la forma más extensa posible, sin ningún tipo de poda.

Una vez realizada esta pequeña introducción teórica pasamos a la realización de los experimentos con ambos árboles. Todos los experimentos están realizados con la herramienta *Weka*[8] y presentamos de manera original los informes generados por la herramienta, pero añadiendo un comentario al principio de cada apartado que explique el mismo.

4.3. Experimento 1

4.3.1. Clasificador utilizado

En este primer experimento trabajamos con el árbol *LMT*.

Scheme: weka.classifiers.trees.LMT -B -P -I -1 -M 15

4.3.2. Relación de filtros utilizados

Los filtros nos proporcionan una manera de podar aquellos atributos menos relevantes o las instancias que pueden generar ruido. En este apartado también podemos visualizar el número de instancias y atributos que posee la muestra.

Relation: final-weka.filters.unsupervised.instance.Resample-S100-Z100.0

Instances: 160

Attributes: 31

4.3.3. Método de prueba

Para este estudio consideraremos el estimador por validación cruzada con *N* conjuntos. Los prototipos de *T* se distribuyen aleatoriamente en *V* conjuntos disjuntos *T1*, *T2*,...*TN* de un tamaño similar ($|Ti| \approx |T|/N$ i = 1, 2,..., *N*). El procedimiento de estimación puede plantearse como sigue:

1. Para todo *n*, n = 1, 2,..., *N*, construir un clasificador usando $T - Tn$, como conjunto de aprendizaje. Sea *dn* el clasificador construido así. Donde ninguno de los prototipos de *Tn* se ha usado para construir *dn*. Al finalizar este paso obtenemos *N* clasificadores, *dn*, con sus correspondientes estimaciones de error.

2. Usando el mismo procedimiento, construir el clasificador *d* usando todos los prototipos de *T*.

Para valores grandes de *N*, cada uno de los *N* clasificadores se construye usando un conjunto de prototipos de tamaño aproximado a $N(1 - 1/N)$, aproximadamente del tamaño de *T*. La suposición básica de la validación cruzada es que este procedimiento es "estable", esto es, que todos los clasificadores *dn*, n = 1, 2,..., *N* (construidos con casi todos los prototipos de *T*) tienen una tasa de error aproximadamente igual a la del clasificador *d* (construido con todos los prototipos de *T*).

Test mode:10-fold cross-validation

Correctly classified instances	130	81.25%
Incorrectly classified instances	30	18.75%
Kappa statistic	0.625	
Mean absolute error	0.213	
Root mean squared error	0.386	
Relative absolute error	42.6026%	
Root relative squared error	77.2043%	
Total number of instances	160	

Tabla 1

4.3.4. Modelo de clasificador (conjunto total de entrenamiento)

En este apartado se recoge el valor de los atributos seleccionados y los pesos de los mismos. También podemos ver el árbol del modelo logístico.

KeywordweightinTitle <= 0.6

| KeywordfrequencyinlinkstextandALT <= 0

| | KeywordfrequencyinTitle <= 0

| | | KeywordfrequencyinMETAKeywords <= 0

| | | | WordsinBody <= 260: LM_1:15/90 (10)

| | | | WordsinBody > 260

| | | | | PageRank <= 1: LM_2:15/105 (17)

| | | | | PageRank > 1: LM_3:15/105 (39)

| | | KeywordfrequencyinMETAKeywords > 0: LM_4:0/60 (4)

| | KeywordfrequencyinTitle > 0: LM_5:15/60 (33)

| KeywordfrequencyinlinkstextandALT > 0

| | KeywordfrequencyinALTattributes <= 1: LM_6:15/60 (37)

| | KeywordfrequencyinALTattributes > 1: LM_7:0/45 (4)

KeywordweightinTitle > 0.6: LM_8:15/30 (16)

Entre paréntesis podemos ver las instancias clasificadas en cada uno de los nodos del árbol.

Time taken to build model: 3.92 seconds

4.3.5. Resumen

En el resumen podemos observar las instancias correctamente clasificadas y las distintas tasas de error (tabla 1).

4.3.6. Detalle de precisión por clase

Ver tabla 2.

Correctly classified instances	156	97.5%
Incorrectly classified instances	4	2.5%
Kappa statistic	0.9499	
Mean absolute error	0.0819	
Root mean squared error	0.1989	
Relative absolute error	16.4313%	
Root relative squared error	39.8405%	
Total number of instances	160	

Tabla 3

4.3.7. Matriz de confusión

Aquí se recoge el número de clasificados: Para el apartado *true* nos indica el número de positivos bien clasificados, que para nuestro caso son 13 y el número de positivos más clasificados que son 34. Para el apartado *false* se recogen 11 mal clasificados con resultado negativo y 92 bien clasificados con resultado negativo.

a	b	<— classified as
63	17	a = true
13	67	b = false

4.4. Experimento 2

No se incluyen explicaciones de los apartados por ser iguales que en el experimento 1.

4.4.1. Clasificador utilizado

En este caso el clasificador es un bosque aleatorio (*Random Forest*).

Scheme: weka.classifiers.trees.RandomForest -I 10 -K 0 -S 1

4.4.2. Relación de filtros utilizados

Se ha realizado una selección de atributos con el fin de simplificar el modelo, intentando buscar los resultados más relevantes posibles.

Relation: final-weka.filters.unsupervised.instance.Resample-S200-Z100.0-weka.filters.unsupervised.attribute.Remove-R1-2,4,6-11,14-21,23-30-weka.filters.unsupervised.instance.Resample-S200-Z100.0-weka.filters.unsupervised.instance.Resample-S100-Z100.0
Instances: 160
Attributes: 6

4.4.3. Método de prueba

Al igual que en el anterior ejemplo, hemos utilizado el método de validación cruzada.

TP rate	FP rate	Precision	Recall	F-measure	Class
0.788	0.163	0.829	0.788	0.808	true
0.838	0.213	0.798	0.838	0.817	false

Tabla 2

Test mode: 10-fold cross-validation

4.4.4. Modelo de clasificador (conjunto total de entrenamiento)

Random forest of 10 trees, each constructed while considering 6 random features.

Time taken to build model: 0.32 seconds

4.4.5. Resumen

Ver tabla 3.

4.4.6. Detalle de precisión por clase

Ver tabla 4.

4.4.7. Matriz de confusión

a	b	<— classified as
63	17	a = true
3	82	b = false

4.5. Resumen de los experimentos

En el primero, el número de instancias correctamente clasificadas es de un 81,25%, mientras que para el segundo caso es de un 97,5%. Además para este segundo experimento el número de instancias bien clasificadas para ejemplos positivos es de 74, con tan solo 1 fallo, mientras que el número de ejemplos negativos bien clasificados es de 82, con 3.

Según podemos apreciar, la tasa de precisión es muy elevada con unos valores de 0,961 para los casos positivos y de 0,988 para los negativos. Igualmente sucede con la tasa de *recall*, siendo de 0,987 para los casos positivos y 0,965 para los negativos.

Otro dato importante es que el tiempo que toma el clasificador *Random Forest* en realizar la clasificación es de 0,32 segundos, lo cual es óptimo para la clasificación online de páginas consultadas.

5. Conclusiones

Cuando iniciamos este trabajo el planteamiento inicial era realizar un estudio clásico de los factores relevantes para el posicionamiento en el motor de búsqueda *Google*. Este estudio contemplaba tanto los elementos propios de la página como las distintas estrategias de posicionamiento, tráfico, etc. Ambos factores analizados uno por uno no aportaban a nuestra investigación ningún dato que esclareciera cómo los distintos buscadores posicionaban nuestras páginas, más bien era frustrante contemplar cómo de la desigualdad de los mismos no se podía inferir ningún tipo de resultado válido. Las razones fundamentales de este fracaso las podemos encontrar en la falta de relación entre los distintos atributos y los pesos que estos elementos aportaban en los criterios utilizados por los buscadores para su posterior posicionamiento.

Para encontrar estos factores que influencian de manera definitiva en el posicionamiento nos hemos apoyado en técnicas de aprendizaje inductivo, y más concretamente en árboles de decisión. Los resultados encontrados en los dos experimentos son buenos, pero fundamentalmente el segundo, con un 97,5% de acierto, nos indica que la hipótesis propuesta es ciertamente válida.

Por tanto, no es necesario para el posicionamiento colocar todos los atributos sino conocer la relación entre los mismos, cómo son las ramas del árbol de decisión utilizadas por el buscador y los pesos de los atributos; es decir, el nivel que ocupan los mismos en el árbol de decisión.

Las recomendaciones prácticas para la selección de atributos generadas por ambos experimentos serían las siguientes: el documento no debe contener demasiadas palabras. Además es muy importante que la palabra clave aparezca en negrita dentro del cuerpo del documento. La primera cabecera, y el peso de la clave en ésta, debe ser lo mayor posible. Por último, la frecuencia y el peso de la clave en el título son factores muy relevantes.

«La mayor parte de los SEO desconoce la influencia real de los factores e incluso de los atributos. Todo esto contribuye a la introducción de *ruido* o de penalizaciones en las páginas»

Además, las técnicas de aprendizaje automático permiten que con un mayor número de muestras y consultas los resultados puedan mejorar, con lo cual con un número adecuado de las mismas podríamos obtener una función muy ajustada al resultado final de un motor de búsqueda.

En un futuro, y basándonos en los experimentos realizados, podríamos construir una aplicación cuyo parámetro de entrada sea un recurso web que genere como salida los valores de aquellos elementos de éste que deban ser mejorados. Además, nos debería indicar cuáles de ellos sólo generan ruido o tienen una aportación nula. Esta herramienta también tendría que mostrarnos cuál sería la situación actual del ranking de una página sobre un rango determinado y para un motor de búsqueda en concreto.

A un nivel más teórico podemos plantearnos cómo ampliar este tipo de herramientas para su interacción con la web semántica. La cuestión radica en que ahora debemos atrapar el contenido semántico de la cadena

de búsqueda. En función de dicho contenido semántico, atributos situados en un nivel inferior del árbol de decisión podrían escalar puestos para ubicarse en niveles superiores y conseguir así que los resultados se ajusten con mayor exactitud a los deseos del usuario. Estos árboles de decisión "dinámicos" posibilitarían la captación del contenido semántico y su integración en la estrategia de búsqueda.

6. Bibliografía

1. **Sparck Jones, K.**; **Willett, P.** *Readings in information retrieval*. San Francisco: Morgan Kaufmann Publishers, Inc., 1997.

2. *Delivering business to businesses. Keyword Generator*. Consultado en: 22-10-04.
*http://www.espotting.com/popups/keywordgenbox.asp_*

3. *Search engine optimization tools. SEO tools*. Consultado en: 22-10-04.
http://www.webconfs.com/search-engine-spider-simulator.php

4. *Alexa. Alexa related links and traffic ranking*.
Consultado en: 22-10-04.
http://www.alexa.com

5. **Russell, Stuart**; **Norving, Meter.** *Inteligencia artificial. Un enfoque moderno*. Prentice-Hall, 2004. Isbn 84-205-4003-X.

6. **Landwehr, Niels**; **Hall, Mark**; **Frank, Eibe.** *Logistic model trees*. Consultado en: 22-10-04.
http://www.cs.waikato.ac.nz/~ml/publications/2003/landwehr-etal.pdf

7. **Leo, Breiman.** *Random forest*. Consultado en: 22-10-04.
http://www.stat.berkeley.edu/users/breiman/RandomForests/cc_home.htm#overview

8. **Witten, Ian H.**; **Frank, Eibe.** *Data mining: practical machine learning tools and techniques with Java implementations*. Consultado en: 22-10-04.
http://www.webconfs.com/search-engine-spider-simulato.php

9. **Alderson, Jeff.** *Xybercode, Inc*. Consultado en: 22-10-04.
http://www.adwordanalyzer.com/

10. *Google*. Consultado en: 23-11-04.
http://www.google.com/technology/

Francisco José Soltero Domingo, *Departamento de Informática, Universidad Carlos III de Madrid.*
fsoltero@inf.uc3m.es

Diego José Bodas Sagi*, Departamento de Informática, Universidad Carlos III de Madrid.*
dbodas@inf.uc3m.es