Dyna
ISSN: 0012-7353
dyna@unalmed.edu.co
Universidad Nacional de Colombia
Colombia

Marín Lopera, Andrés; Ortega Lobo, Oscar; Branch, John William
Agrupamiento de resultados obtenidos de búsquedas hechas sobre la Web para un catálogo de acceso público en línea
Dyna, vol. 71, núm. 142, julio, 2004, pp. 57-67
Universidad Nacional de Colombia
Medellín, Colombia

AGRUPAMIENTO DE RESULTADOS OBTENIDOS DE BÚSQUEDAS HECHAS SOBRE LA WEB PARA UN CATÁLOGO DE ACCESO PÚBLICO EN LÍNEA

ANDRÉS MARÍN LOPERA
Profesor del Departamento de Ingeniería de Sistemas. Universidad de Antioquia
amarin@udea.edu.co

OSCAR ORTEGA LOBO
Profesor del Departamento de Ingeniería de Sistemas. Universidad de Antioquia
oortega@udea.edu.co

JOHN WILLIAM BRANCH
Profesor de la Escuela de Sistemas. Facultad de Minas. Universidad Nacional de Colombia
jwbranch@unalmed.edu.co

Recibido para revisar 14 de Mayo de 2003, aceptado 7 de Junio de 2003, versión final 7 de Octubre de 2003

RESUMEN: Los Catálogos de Acceso Público en Línea (*OPAC*) permiten consultar colecciones disponibles en bibliotecas. Cuando un usuario de un *OPAC* formula una consulta demasiado general, recibe una lista extensa de fichas bibliográficas. El usuario debe leer toda la lista resultante y seleccionar las fichas interesantes, lo cual puede tomar tiempo y conducir a ignorar fichas pertinentes.

El ser humano es capaz de explorar ágilmente información organizada en estructuras jerárquicas. Una lista de fichas resultante de una consulta a un *OPAC* puede ser organizada, con la ayuda del computador, en una estructura jerárquica. En este estudio se emplea un algoritmo de agrupamiento jerárquico no supervisado, denominado *Principal Direction Divisive Partitioning*, el cual es aplicado en un prototipo de interfaz Web para un *OPAC* denominado *Cataweb*. Un análisis del primer nivel de una jerarquía producida a partir de los resultados de la consulta: *"sistem% geogra%"*, mostró la validez de las agrupaciones obtenidas.

PALABRAS CLAVES: Opac, agrupamiento, minería de datos , descubrimiento de patrones en los datos de la Web, CDS/ISIS

ABSTRACT: Online Public Access Catalogs (OPAC) offer query services to library users. Queries that are imprecisely formulated produce long lists of bibliography entries which must be read by the users in order to select the most relevant ones. Reading time and effort can take the user to miss very interesting bibliography entries.

Humans are skilled to quickly obtain information goals by exploring hirarchical structured information. A bibliography list issued as a query result can be hierarchically organized by a computer algorithm. In the present study, a non-supervised hierarchical clustering algorithm, the *Principal Direction Divisive Partitioning (PDDP)*, is used for rendering the bibliography lists resulting from queries on an OPAC called CATAWEB. Careful analysis of the first level of the rendered hierarchy for the query *"system% geogra%"* showed the validity of the clusters obtained.

KEYWORDS: Opac, clustering, data mining, web mining, CDS/ISIS

1 INTRODUCCIÓN

Wold Wide Web es un amplio recurso de información y servicios que continúa con un rápido crecimiento. Se han desarrollado poderosos motores de búsqueda para la localización de documentos de acuerdo a sus contenidos. Estos buscadores contienen enormes índices a los documentos disponibles en la Web y mediante las consultas que hacen los usuarios, los motores entregan las direcciones Universal Resource Location (URL) de aquellos documentos que satisfacen la consulta. Frecuentemente las consultas recuperan resultados que aunque satisfacen los criterios de búsqueda no son de interés para el usuario.

La caracterización y clasificación de documentos en la Web es un tema de estudio y se ha venido trabajando mediante algoritmos *inteligentes* que tratan de extraer la estructura semántica de los documentos basado en las palabras del documento o en la estructura de las etiquetas *HTML*. Las técnicas de *agrupamiento*, en inglés: *(Clustering)*, ofrecen la ventaja de permitir hacer procesos de categorización no supervisada por humanos y además no requiere un conocimiento *a priori* de categorías.

Los *OPAC* de hoy en día son aplicativos de software que han sido migrados a la *Web* ya sea en forma nativa a través del software en que está hecho, a través de interfaces a otros sistemas existentes o mediante interfaces de tipo *Web* a sistemas de consulta estándar como lo es el protocolo Z39.50 [1].

En este trabajo se pretende mostrar como se puede aplicar una técnica de *agrupamiento* al sistema OPAC *(On-line Public Access Catalog)* CataWeb [2] desarrollado en la Universidad de Antioquia, para ayudar a los usuarios finales a tener consultas mas efectivas, mediante el agrupamiento automático de los resultados obtenidos en consultas; de esta forma, el usuario puede descartar lo que no requiere para que se pueda concentrar en lo que está buscando realmente.

2 PROBLEMA

2.1 CATAWEB EL PROYECTO "CATAWEB

Generación automática de páginas Web con soporte de búsquedas para un catálogo bibliotecario" desarrollado en la Universidad de Antioquia [2] permite la integración de catálogos bibliotecarios que provengan de diferentes sistemas de catalogación en un mismo motor de búsqueda.

Cataweb convierte cada entrada del catálogo original en una página Web, la cual queda indexada en un motor de búsqueda de Internet, ver la Figura 1. Las búsquedas dentro de *Cataweb* son similares a las de cualquier buscador de Internet, esto es, sin ninguna estructura preestablecida; es decir, todas las palabras de la entrada del catálogo son clave, se puede formular una búsqueda con el apellido de un autor y, por ejemplo, una palabra que esté en el título, lo cual sería válido. *Cataweb* define un formato de entrada fijo para el proceso de incorporación de catálogos bibliotecarios, el cual se puede generar fácilmente con los reportes que permite efectuar el software *CDS/ISIS*; en otros sistemas se debe desarrollar una interfaz que lo genere. Una vez que el catálogo es dado en el formato preesTablacido, se pasa a través de un reconocedor, el cual distingue una entrada del catálogo de las demás y con ella genera un archivo con etiquetas *HTML* que a su vez se constituye en una de las páginas *Web* del sitio que ofrece el servicio. Una captura de pantalla típica se puede observar en la Figura 2. Se genera una página *Web* por cada entrada dentro del catálogo original. En la red Internet hay disponibles programas de uso libre que sirven como mecanismo de indexación de páginas *Web* para luego poder hacer búsquedas sobre ellas, tales como, el *Mnogosearch* [3] y el *Htdig* [4]. Durante el desarrollo del proyecto *Cataweb* se evaluaron ambos motores pero finalmente se seleccionó *Mnogosearch*.

Todas las palabras que componen la entrada del catálogo se convierten en palabra clave, una vez son indexados por el motor de

búsqueda. Tener todas las palabras como índices es ventajoso porque se flexibilizan las búsquedas; sin embargo, en ciertos casos, se puede saturar al usuario debido a que el sistema puede arrojar demasiada información.

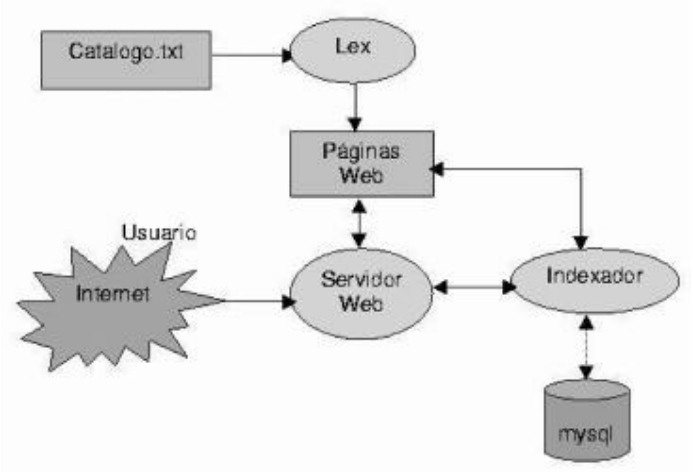

Figura 1. Esquema de operación del sistema *Cataweb*

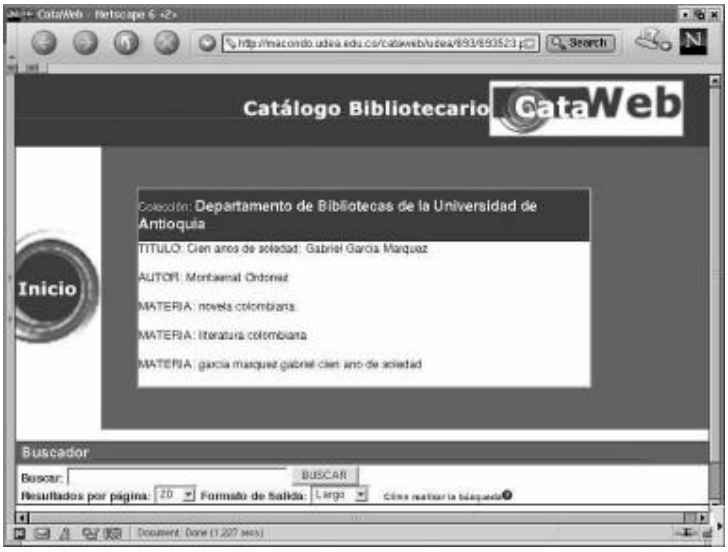

Figura 2. Captura de una pantalla típica en el sistema *Cataweb*

2.2 DIFICULTADES

Cuando las consultas que se formulan a los sistemas *OPAC's* son demasiado generales, los resultados pueden ser muy abundantes, incluso una lista que se extiende por varias páginas, en la cual seguramente los resutados estarán en un orden que no necesariamente es el más conveniente para la necesidad del usuario. Aunque el usuario tiene opciones de refinar su búsqueda restringiendo el dominio de búsqueda mediante el uso de operadores de tipo lógico o booleanos, hay estudios que demuestran que el uso de operadores lógicos o booleanos no es bien comprendido por el público general, se tienden a interpretar de forma incorrecta [5]. Existen diferencias entre la forma cómo se formulan las búsquedas en

los sistemas de catálogo y la forma cómo los usuarios formulan en su mente las búsquedas.

Primero, los OPAC exigen el uso de los mismos términos con los cuales se crearon los índices. Segundo, los OPAC hacen transformaciones de las consultas del usuario, o exigen que el usuario realice las transformaciones en forma de expresiones lógicas, las cuales tienen un significado diferente para el usuario y para el sistema [6]. Bates [7] encontró que existe solamente una probabilidad del 10% al 20% de que dos personas usen el mismo término para un concepto. Peters [8] afirma que el 58% de las sesiones de búsqueda comienzan con términos no usados en los sistemas de vocabulario controlados. Fenómenos como los dos mencionados introducen ineficacias en el proceso de búsqueda asociado a las consultas realizadas a los *OPAC's*.

2.3 OBJETIVO

Se propone evaluar un sistema que permita agrupar resultados obtenidos de sistemas de catálogos públicos en línea de forma automática, para que así el usuario vaya descartando grupos indeseados y analice los más pertinentes para su búsqueda. Se desea probar un método específico que emplee técnicas de minería de textos para agrupar resultados obtenidos de una consulta, para presentar al usuario final los resultados de su búsqueda agrupados automáticamente de acuerdo a un número de grupos prefijado.

3 ANTECEDENTES

Se ha venido trabajando en dos tipos de técnicas de agrupamiento. El primer tipo son los algoritmos de agrupamiento de particionamiento jerárquico, los cuales se pueden usar para calcular una solución de agrupamiento jerarquizado empleando una metodología de biseccionamiento de grupos. En esta metodología, todos los documentos son inicialmente particionados en dos grupos, entonces, uno de estos grupos conteniendo más de un documento se selecciona para ser biseccionado de la misma forma. Este proceso puede continuar hasta alcanzar un número de grupos prefijado o incluso hasta obtener un árbol en cuyas hojas sólo quede un único documento. Esta metodología construye su árbol jerarquizado desde arriba hacia abajo, lo cual se denomina en inglés *top down*.

El segundo tipo de técnicas de agrupamiento son los algoritmos aglomerativos, en las cuales se construye un árbol de agrupamiento de la siguiente manera. Inicialmente se construye un grupo por cada documento, los cuales constituirán las hojas del árbol; luego, repetidamente, se crean nodos intermedios conformando pares de grupos, hasta obtener el nodo raíz del árbol. Los algoritmos aglomerativos construyen así el árbol de abajo hacia arriba, lo cual se denomina en inglés *bottom up*.

En [9] se evalúan diferentes algoritmos de agrupamiento con el ánimo de poder comparar tanto las metodologías aglomerativas como las particionales. En dicha evaluación se concluye que los algoritmos particionales dan mejores resultados para grandes conjuntos de documentos no sólo debido a los bajos requerimientos computaciones sino también a sus resultados mejores o comparables en cuanto a la calidad del agrupamiento.
Los algoritmos de agrupamiento pueden jugar un papel importante en la caracterización y clasificación no supervisada de documentos, al proveer un mecanismo para poder organizar y visualizar grandes cantidades de información en pequeños grupos que contengan algunas similaridades en sus significados. En particular, las soluciones de agrupamiento jerárquico dan una vista de los datos a diferentes niveles de granularidad, haciéndoles ideales para que las personas puedan visualizar e interactivamente explorar grandes colecciones de documentos.

3.1 SELECCIÓN DE LA TÉCNICA EMPLEADA

En la tabla 1 se presentan los criterios que se consideraron al comparar las metodologías de algoritmos de agrupamiento con las de tipo particionamiento jerárquico y las de tipo aglomerativo jerarquizado para ser aplicadas

en el *Agrupamiento de resultados obtenidos de búsquedas hechas sobre la Web para un catálogo de acceso público en línea*.

Con base en los criterios de la Tabla 1 se escogió una metodología de particionamiento jerárquico.

Tabla 1. Metodologías particionales vs. Metodologías aglomerativas

Requerimientos	**Metodologías particionales** ***(Top Down)***	**Metodologías aglometarivas** ***(Botom Up)***
Baja carga computacional	Sí	No
Rápida obtención del prototipo	Sí	No
Mejores resultados en agrupamientos	Si	No

4 MÉTODO

Para este trabajo se seleccionó el algoritmo PDDP que es del tipo particionamiento jerárquico.[10]

4.1 PRELIMINARES

Definición: Si $\boldsymbol{A}$ es una matriz n n sobre un espacio vectorial $\boldsymbol{F}$, un **valor propio** de $\boldsymbol{A}$ en $\boldsymbol{F}$ es un escalar c de $\boldsymbol{F}$ tal que la matriz $\boldsymbol{A}$-$c\boldsymbol{I}$ es singular (no inversible); esto es, c es un **valor propio** de $\boldsymbol{A}determinante(c\boldsymbol{I}\text{-}\boldsymbol{A})=0$. $\boldsymbol{X}$ es un **vector propio** de $\boldsymbol{FF}(\boldsymbol{X})=c\boldsymbol{X}$, donde c es un **valor propio** de $\boldsymbol{A}$.

El algoritmo *Principal Direction Divisive Partitioning* se aplica a vectores de documentos. Un vector de documento $\boldsymbol{d}=(d_1,d_2,...,d_n)^T$ es un vector columna cuya entrada i-ésima, d_i, es la frecuencia relativa de la palabra j-ésima, ver la Tabla 2.

Tabla 2. Ejemplo de una matriz de frecuencias de palabras en documentos

En la primera columna aparecen todas las palabras ocurrentes en los documentos. Las columnas siguientes representan documentos y cada campo contiene la frecuencia de la palabra en el documento

	Nombre de documento				
Palabras	**Fútbol**	**Turismo**	**Fiestas**	**Vías**	**Empleo**
Cali	2	2	1	2	1
Medellín	3	4	0	1	3
Pereira	1	2	0	0	0
Ibagué	2	5	0	3	3
Tunja	0	9	0	1	0
Pasto	2	1	1	0	1

Los vectores de documentos se normalizan, esto es, cada d_i se transforma, como lo ilustra la ecuación (1), de tal manera que la norma del vector es igual a 1, así que cada entrada es un valor numérico dado por:

$$d_i=\frac{TF_i}{\sqrt{\sum_j (TF_j)^2}} \qquad (1)$$

donde TF_i es el número de ocurrencias de la palabra i en el documento específico $\boldsymbol{d}$. Se denota a (1) como “norma de escalamiento”.

Dada una colección de documentos $\boldsymbol{d}_1,\ldots,\boldsymbol{d}_m$, la media o centroide del conjunto de documentos es:

$$\boldsymbol{w}=\frac{\boldsymbol{d}_1+...+\boldsymbol{d}_n}{m}=\boldsymbol{M}\cdot\boldsymbol{e}\cdot\frac{1}{m}, \qquad (2)$$

donde $\mathbf{M}=(d_1,..,d_m)$ es una matriz $n \times m$ de vectores de documentos, $\mathbf{e}=(1,1,\ldots,1)^{\mathrm{T}}$ es el vector cuyos elementos son todos *unos*. Si $\boldsymbol{w}=\mathbf{0}$, entonces la matriz de covarianza debe ser $\boldsymbol{M}\cdot\boldsymbol{M}^T$, de aquí que cada elemento es un vector columna, pero en el caso general la matriz de covarianza es:

$$\boldsymbol{C}=(\boldsymbol{M}-\boldsymbol{we}^T)\cdot(\boldsymbol{M}-\boldsymbol{we}^T)^T=\boldsymbol{A}\cdot\boldsymbol{A}^T, \qquad (3)$$

donde $\boldsymbol{A}=(\boldsymbol{M}-\boldsymbol{we}^T)$. Esta matriz es simétrica y definida positiva, así que todos los *valores propios* son reales y no negativos. Los *vectores propios* correspondientes a los k mayores *valores propios* son llamados *principales componentes* o *principales direcciones*. Para el algoritmo *PDDP*, interesan sólo los *vectores propios* de $\boldsymbol{C}$, no los *valores propios*, así que el escalonamiento de la matriz $\boldsymbol{C}$ no es importante para el algoritmo mencionado.

La descomposición singular de valores de una matriz $\boldsymbol{A}_{n\times m}$ (SVD) se define por la expresión

$\boldsymbol{A}=\boldsymbol{U}\mathbf{S}\,\boldsymbol{V}^T$ donde $\boldsymbol{U}$, $\mathbf{S}$, $\boldsymbol{V}$ son respectivamente, $n\times n, n\times m, m\times m$, $\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}, \boldsymbol{V}^T\boldsymbol{V}=\boldsymbol{I}$, y $\mathbf{S}=diag\{\sigma_1,\sigma_2,\ldots,\sigma_{min\{m,n\}}\}$.

Las entradas diagonales de $\mathbf{S}$ son llamadas *valores singulares* y satisfacen $\sigma_1\geq\sigma_2\geq\ldots\geq\sigma_{min\{m,n\}}\geq 0$

Las columnas de $\boldsymbol{UV}$ son los vectores *singular izquierdo* y *singular derecho* respectivamente. Se sabe que la descomposición en vectores y valores propios de $\boldsymbol{C}=\boldsymbol{AA}^T$ se relaciona al SVD de $\boldsymbol{A}$ por:

$$\boldsymbol{C}=\boldsymbol{AA}^T=(\boldsymbol{U}\mathbf{S}\,\boldsymbol{V}^T)\cdot(\boldsymbol{V}\mathbf{S}^T\boldsymbol{U}^T)=\boldsymbol{U}\mathbf{S}^2\boldsymbol{U}^T, \qquad (4)$$

donde $\mathbf{S}^2$ denota a la matriz diagonal $n\times n$:

$$diag\left\{\boldsymbol{s}_1^2,\boldsymbol{s}_2^2,...,\boldsymbol{s}_{\min\{\min\}}^2,0,...,0\right\} \qquad (5)$$

(se extiende con ceros si $m<n$).

4.2 ESQUEMA GENERAL DEL ALGORITMO PDP

El algoritmo PDDP opera en un espacio muestral de m muestras en las que cada una es un vector que contiene valores numéricos. Cada documento se representa por un vector columna (1) de valores de atributos, los cuales son contadores de palabras normalizadas de tal forma que cada vector documento tiene una longitud de 1. (Ver la Tabla 2).

- Se comienza separando el conjunto entero de documentos en dos particiones usando *la dirección principal*.
- Cada una de las dos particiones se separa en dos subparticiones usando el mismo proceso recursivamente.

Como resultado se genera una estructura jerárquica de particiones organizadas en un árbol binario, en el cual cada partición es un nodo hoja (no ha sido dividido) o ha sido separado en dos subparticiones que conforman dos hijos del árbol, ver Figura 3.

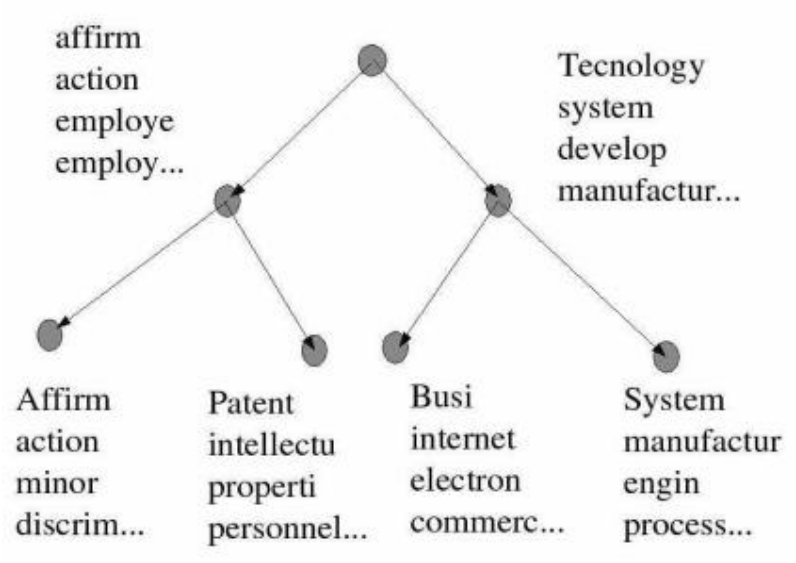

Figura 3. Ejemplo de un particionamiento jerarquizado [11]

4.3 DIVIDIENDO UNA PARTICIÓN

Una partición de p documentos se representa por una matriz de $n \times p$ $\boldsymbol{M_p}=(\boldsymbol{d_1} \cdots \boldsymbol{d_p})$ donde cada $\boldsymbol{d_i}$ es un vector-n que representa a un documento. La matriz $\boldsymbol{M_p}$ es una submatriz de la original $\boldsymbol{M}$ que consiste en una selección de p columnas de $\boldsymbol{M}$. La *principal dirección* de $\boldsymbol{M_p}$ son los vectores propios de su matriz de covarianza $\boldsymbol{C}$. Sea $\boldsymbol{w}=\frac{\boldsymbol{M_p e}}{p}$ la media o centroide de los documentos $d_1,\cdots,d_p$ y la matriz de covarianza es $\boldsymbol{C}=(\boldsymbol{M_p}-\boldsymbol{we}^T)(\boldsymbol{M_p}-\boldsymbol{we}^T)^T$.

La transformación de Karhuenen-Loeve consiste en la proyección de las columnas de $\boldsymbol{M_p}$ dentro del espacio señalado por los k principales **vectores propios** (aquellos que corresponden a los mayores k **valores propios**). El resultado es una representación de los datos originales en k grados de libertad en vez de n. Además de reducir la dimensionalidad, la transformación tiene otro efecto beneficioso: la remoción de ruido en los datos siempre y cuando se haya alcanzado un valor apropiado de k. En este caso el interés está en proyectar temporalmente cada documento dentro del **vector propio** principal denotado como $\boldsymbol{u}$, también denominado como componente principal o dirección principal. Esta proyección sólo se usa con el único propósito de determinar como quedan los documentos biseccionados en dos grupos y nada más.

La proyección del i-ésimo documento d_i está dada por la fórmula:

$$\sigma v_i=\boldsymbol{u}^T(\boldsymbol{d_i}-\boldsymbol{w}), \qquad (6)$$

donde σ es una constante positiva, es equivalente a trasladar todos los documentos de tal forma que su media es el origen y entonces proyectarlos hacia la principal dirección. Todos los documentos $\boldsymbol{d_i}$ para los cuales $v_i \leq 0$ son particionados dentro del hijo de la izquierda, de lo contrario, quedan dentro del hijo de la derecha del árbol.

4.4 CONSIDERACIONES COMPUTACIONALES

Dentro del cálculo del particionamiento la parte más costosa es el cálculo de los **valores propios** y los **vectores propios** de la matriz de covarianza $\boldsymbol{C}$, en especial si $\boldsymbol{C}$ (3) tiene una gran dimensionalidad. Es aquí donde la descomposición por valores singulares (SVD) es muy importante ver (5). De aquí se puede notar que los **vectores propios** de $\boldsymbol{C}$ son las raíces cuadradas de los vectores singulares de

$\boldsymbol{A}$. Los vectores singulares también satisfacen la siguiente relación. Sea $\boldsymbol{u_j}$,$\boldsymbol{v_j}$ denota la j-ésima columna de $\boldsymbol{U}$,$\boldsymbol{V}$, respectivamente, se tiene que:

$$\boldsymbol{u}_j^T\boldsymbol{A}=\boldsymbol{u}_j^T\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T=\sigma_j\boldsymbol{v}_j^T, \quad (7)$$

y de otro lado $\boldsymbol{A}\boldsymbol{v}_j=\sigma_j\boldsymbol{u}_j$. Usando la definición de $\boldsymbol{A}$, se puede ver que la proyección (6) corresponde exactamente a las entradas en el vector $\sigma_1\boldsymbol{v}_1^T$. Esto significa que una vez se tiene el SVD de la matriz $\boldsymbol{A}$, ya se tiene la proyección necesaria para dividir el grupo $\boldsymbol{M_p}$

Como no se requiere el SVD entero de $\boldsymbol{A}$, sólo se necesitan los vectores singulares principales $\boldsymbol{u}_1\boldsymbol{v}_1$. El SVD parcial de una matriz se puede obtener rápidamente usando el algoritmo Lanczos, el cual toma ventaja de la dispersidad presente en la matriz $\boldsymbol{M_p}$.

El proceso dentro del algoritmo Lanczos consiste en multiplicar repetidamente un vector $\boldsymbol{x}$ por la matriz $\boldsymbol{C}$ y entonces ortogonalizar el resultado contra todos los previos vectores de la secuencia. El algoritmo también construye una secuencia de matrices tridiagonales anidadas de dimensiones incrementales, calculando los valores propios de cada una hasta cumplir una condición de parada.

5 EVALUACIÓN

5.1 CONDICIONES EXPERIMENTALES

Se desea evaluar la calidad de los agrupamientos automáticos obtenidos por el algoritmo *PDDP* al aplicarlo a los resultados de una consulta específica hecha al sistema de catálogo bibliográfico *Cataweb*.

Para las pruebas realizadas se utilizaron los datos del catálogo del centro de documentación de la Facultad de Ingeniería de la Universidad de Antioquia. Dicho centro de documentación se especializa en temas de Ingeniería y en particular en aspectos del medio ambiente. El catálogo está cargado en el sistema *Cataweb* con 5000 registros aproximadamente.

Como se mencionó anteriormente, el sistema *Cataweb* indexa todas las entradas del catálogo, las cuales son páginas *Web*, mediante un motor de búsqueda. El motor de búsqueda utilizado, *Mnogosearch*, extrae las palabras que aparecen en las páginas *Web*, las que considera llaves de búsqueda que le permiten ubicar un documento ante una consulta de un usuario.

Con el fin de limitar el posible ruido que causan palabras muy repetidas y no relevantes que aparecen en todas o en varias entradas del catálogo, se usaron tres estrategias que tratan de evitar que dichas palabras sean ingresadas como índices dentro del motor de búsqueda. Primera, para los textos de ayuda, títulos, etiquetas y aspectos de diseño de la página *Web*, que usan palabras fijas, éstas se convirtieron en imágenes. Segunda, para las palabras de alta frecuencia del español, como son los artículos y los pronombres, se hace uso del mecanismo de *stop words* del motor de búsqueda, el cual consiste en ignorar estas palabras tanto en las consultas como en los índices asociados a la página *Web*. Tercera, los valores numéricos no se consideran como palabras válidas en los agrupamientos, por lo que son ignorados. La reducción de la cantidad de palabras por documento trae como beneficio adicional que se disminuye el trabajo del algoritmo *PDDP*.

Se asume que cada palabra existente en cada documento aparece sólo una vez, lo cual evita costos computacionales altos y por lo demás es de esperar que en las entradas de un catálogo bibliotecario no se repitan palabras debido a que en realidad no son textos fluidos, sino valores asociados a unos descriptores.

Al algoritmo *PDDP* se le definió como parámetro que agrupara los resultados en 5 grupos.

5.2 DESCRIPCIÓN DE RESULTADOS

Para las pruebas se buscó el patrón *sistem% geogra%*; es decir, se quiere encontrar todas las entradas del catálogo bibliotecario que tengan palabras que empiecen por *sistem* y que también tengan palabras que empiecen por *geogra*. La búsqueda sobre el sistema Cataweb arroja como resultado 39

documentos representados en páginas *Web*, que cumplen con el patrón de búsqueda. Dichos documentos son agrupados usando el algoritmo *PDDP*.

En la Tabla 3 se presentan los resultados obtenidos para cada grupo. Las palabras mostradas de cada grupo son aquellas que tienen mayor peso dentro del grupo y por tanto lo caracterizan.

Tabla 3. Grupos obtenidos después del agrupamiento

Cada columna representa las palabras principales que caracterizan al grupo.

Grupo 1 13 documentos	**Grupo 2** 6 documentos	**Grupo 3** 7 documentos	**Grupo 4** 6 documentos	**Grupo 5** 7 documentos
medellín	of	bc	geográfica	geografía
universidad	digitales	geografía	información	economía
antioquia	artificial	geográfica	sig	económica
grado	inteligencia	información	revista	bh
facultad	simulación	color	planificación	ambiente
ingeniería	and	cartografía	evaluación	medio
da	application	satelitales	acodal	desarrollo
electrónica	in	sistemas	herramienta	capitalismo
departamento	industria	sig	no	especio
área	recuperación	satélites	ps	global
especialista	algoritmos	remotos	gestión	globalización
electrónicas	bases	componentes	usos	innovación

El **grupo 1** es bastante interesante pues allí quedaron agrupados 13 documentos, 11 de ellos con características similares tales como trabajos de tesis, monografías, trabajos de grado, que hay en el centro de documentación y que contienen el patrón de búsqueda dado, los otros 2 documentos corresponden a un proyecto llamado Sigma. Las palabras de mayor peso encontradas por método *PDDP*: *Medellín, universidad, Antioquia, grado, Facultad e Ingeniería*. El **grupo 2** está compuesto por 6 documentos, no muy relacionados aparentemente; sin embargo, hay algo especial en los documentos de este grupo, 4 de ellos están en idioma inglés. En el **grupo 3** hay agrupados 7 documentos sobre mapas, cartografía, etc., tienen en común la parte inicial de su código de clasificación **bc**. El **grupo 4** tiene 6 documentos, las palabras *geográfica, información, sig* son las de mayor peso. En el **grupo 5** hay 7 documentos; las palabras de mayor peso son *geografía, economía, económica, bh* .

5.3 ANÁLISIS

El método *PDDP* de agrupamiento, como se dijo, es no supervisado; es decir, no requiere intervención humana, ni tampoco requiere entrenamiento previo, en el experimento sólo se le entregaron los 39 documentos o entradas del catálogo bibliotecario y se le fijó el parámetro de número de grupos a 5.

En el **grupo 1** la relación entre sus documentos es tan fuerte que la primera bisección que se hizo del universo de documentos dejó este grupo aparte y el resto de grupos se generó a partir de nuevas bisecciones del resto de documentos. La mayo parte de documentos pertenecen a tesis de grado de la universidad de Antioquia pero hay dos que son de otra cosa, esto se puede explicar porque la palabra **Medellín** es una de las de mayor peso en la caracterización del grupo lo cual es común a todos estos documentos.

El **grupo 2** está compuesto por varios documentos en idioma inglés; esto se explica debido a que por una parte no se cargaron las *stop words* del inglés y por otra, a pesar de que el *Monogosearch* sí trabaja con distintos idiomas, el *Cataweb* genera páginas *Web* marcadas para un solo idioma. Mirando la palabra de mayor peso: *of*, queda claro que esta preposición del inglés, bastante común, fue la que caracterizó a este grupo y debió quedar filtrada dentro de las *stop words*.

En el **grupo 3** la palabra de mayor peso *bc*, debió haber sido filtrada dado que, como se ve, no es una palabra realmente, sino que corresponde a la parte alfabética del código de clasificación de las entradas del catálogo.
El **grupo 4** es una agrupación bastante buena; como se sabe, *SIG* es el acrónimo de Sistemas de Información Geográfica y sería de esperar que apareciera considerando el patrón de búsqueda empleado.
El **grupo 5** es una agrupación bastante buena, aunque como se ve, *bh* es también un código alfanumérico que no debió haber sido considerado y esta siendo una palabra de alto peso.
El prefijar el número de grupos puede afectar los resultados debido a que si es más grande de lo que debería ser, se hacen bisecciones innecesarias que pueden desorientar al usuario y si por el contrario son muy pocas pueden quedar grupos con documentos no muy relacionados entre sí. Como ejemplos de esto, son los dos documentos diferentes al patron general que se ven en el grupo 1. De otro lado, como el agrupamiento se hace sobre todos los elementos de cada ficha bibliográfica, esto ocasiona que se generen agrupaciones de cosas no homogéneas, por ejemplo no es igual que la palabra Medellín aparezca en la información de la publicación que en un título, pues en este último se sabe que el material trata sobre Medellín y en el primero el lugar de publicación es algo accidental.

6 CONCLUSIONES Y TRABAJO FUTURO

En el presente trabajo se exploró el uso de una técnica de agrupamiento jerárquico no supervisado para organizar una lista de fichas bibliográficas resultante de una consulta a un catálogo bibliotecario de acceso público en línea.
La técnica empleada es Principal Direction Divisive Partitioning [10], la cual, aunque está diseñada para agrupar documentos de texto completo, probó ser aplicable al conjunto de fichas obtenidas de búsquedas hechas sobre la *Web* para un catálogo de acceso público en línea llamado *CataWeb*. La evaluación incluida en el artículo abarca los resultados de una sola consulta y es considerada preliminar. Se requiere validar los resultados con un diseño experimental que abarque mas consultas y que incluya la evaluación de las jerarquías resultantes por parte de expertos en las temáticas asociadas a dichas consultas. Sería recomendable hacer nuevos experimentos pero agrupando solo las partes de la ficha autor, titulo y materia, pues el resto de elementos introducen ruido.
La interfaz desarrollada en el prototipo queda bastante completa; sin embargo, el algoritmo PDDP está especificado en lenguaje script de *MATLAB*. Sería recomendable reescribir el código del algoritmo PDDP en un lenguaje que sea más apto para la *Web*, considerando además los problemas de concurrencia por el acceso simultáneo de varios usuarios.

7. REFERENCIAS

[1] A. Z39.50, "Information retrieval (z39.50): application service definition and protocol specification," available: http://www.loc.gov/z3950/agency, [citado en Agosto del 2002].
[2] A. Marín, I. Ramírez, and D. Murillo, *Generación automática de paginas Web con soporte de búsquedas para un catalogo bibliotecario [CataWeb]*, [tesis de pregrado] ed., Universidad de antioquia, Ingenieria de Sistemas, 2001, available: http://bochica.udea.edu.co/~cendoi.
[3] L. Corp., "mnogosearchtm (former udmsearch) web search engine software," available: http://www.mnogosearch.org/, [citado en Agosto del 2002].
[4] T. ht://Dig Group, "Www search engine software," available: http://www.htdig.org/, [citado en Agosto del 2002].
[5] T. Bellardo, "An investigation of online searcher traits and their relationship to search outcome," *Journal of the American Society for Information Sciences*, vol. 36, pp. 241–250, 1985.
[6] C. L. Borgman, "Why are online catalogs still hard to use? " *Journal of the American Society for Information Sciences*, vol. 47, pp. 493–503, 1996.
[7] M. Bates, "Subject access in online catalogs: A design model." *Journal of the*

American Society for Information Sciences, vol. 37, pp. 357–376, 1986.

[8] T. A. Peters, "Controlled and uncontrolled vocabulary subject searching in and academic library online catalog," *Information Technology and Libraries*, vol. 27, pp. 201–211, 1991.

[9] Y. Zhao and G.Karypisi, "Evaluation of hieraquical clustering algorithms for document datasets," University of Minnesota, Tech. Rep., 2002.

[10] D. L. Boley, "Principal direction divisive partitioning," Department of Computer Science, University of Minnesota, Minneapolis, Tech. Rep., 1999.

[11] D. Boley, "Principal direction partitioning in data mining," 2000, slides of talk given at Stanford, February, 2000,Available: http://www-users.cs.umn.edu/ boley/PDDPslides00.pdf, [citado en Agosto del 2002].