UNIVERSIDAD SAN PABLO - CEU

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA SUPERIOR DE TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA

CLASIFICACIÓN NO SUPERVISADA DE DOCUMENTOS

Autor: David Bravo Alcobendas.

Director: Carlos Oscar Sánchez Sorzano.

Septiembre 2008

UNIVERSIDAD SAN PABLO-CEU

ESCUELA POLITÉCNICA SUPERIOR

División de Ingeniería Informática y de Telecomunicación

Calificación del Proyecto Fin de Carrera

Datos personales del alumno

D.N.I.	
APELLIDOS	NOMBRE

Directores

Director 1 (tantos como sean los directores)

D/D*

Tribunal calificador

Presidente

D/D*	FIRMA

Secretario

D/D*	FIRMA

Vocal

D/D*	FIRMA

Fecha de calificación

Calificación

Resumen

La recuperación de información, es la ciencia que se encarga de la búsqueda de información en documentos, también dentro de conjuntos de documentos y en las fuentes donde se almacenan datos de tipo texto, imágenes o sonido.

Una de las formas de recuperación de información a partir del aprendizaje, consiste en realizarlo mediante una clasificación no supervisada que se basa en muestras que previamente no están clasificadas.

En este proyecto se estudian las técnicas de clasificación existentes para desarrollar un programa que clasifique documentos en el ámbito de la clasificación no supervisada.

Se trabaja sobre un modelo de espacio vectorial de términos que se consigue tras analizar los documentos en formato texto, aplicando técnicas de representación de datos de alta dimensionalidad sobre espacios de menor dimensionalidad que conserven sus propiedades.

Se descubren los atributos que más diferencian al conjunto de los documentos pertenecientes al corpus, detectando y mostrando las características que comparten ciertos documentos, como pueden ser los temas más característicos y descriptivos que se están compuestos a su vez por listados de palabras que los identifican.

Se evalúa el rendimiento ofrecido por cada una de estas técnicas de representación de textos, se analizan para diversos corpus de documentos, optimizando el coste computacional de los algoritmos y los resultados de la clasificación.

Abstract

The information retrieval is the science responsible of finding information in documents, also within sets of documents and sources where data is stored in text, images or sound.

This project explores the existing clustering techniques to develop a program to classify documents in the field of unsupervised classification. It works on a model of space vector of terms that is achieved after analyzing documents in text format, using techniques of representation of high dimensional data in a smaller dimensionality space that retains their properties.

We discover the most distinguish attributes to all documents belonging to the corpus, detecting and displaying the characteristics they share certain documents, as maybe the most characteristic themes and narrative that are constructed in turn by word lists that identify them.

It assesses the performance offered by each of these techniques representation of texts, analyses for different body of documents, optimizing the cost of computational algorithms and results of the classification.

Agradecimientos

A mis padres y abuelo por su apoyo, comprensión y consejos que me han dado en toda la carrera y durante el proyecto y que tanto se han preocupado por mí, que han sabido darme ánimos siempre que lo he necesitado.

A mis amigos en quienes siempre nos apoyamos cada vez que estamos atascados.

A mi director de proyecto que tanto tiempo me ha dedicado y que sin su ayuda esto no habría salido a flote, que me ha ayudado a superar metas que veía muy difíciles.

Índice de contenidos

Índice de Tablas

Índice de Ilustraciones

1 Introducción

1.1 Descripción General

Debido a un incremento del almacenamiento de la información de manera digital en formato de texto promovido por el avance de las nuevas tecnologías, encontramos cada día numerosas fuentes de información, muy útiles tomadas en conjunto pero poco prácticas debido a que previamente tenemos que hacer un análisis del conjunto de la información.

Para ser capaces de asimilar todos los documentos que tenemos ante nosotros, es necesario realizar una tarea de filtrado, para encontrar la información que realmente necesitamos, se trata de desechar todo aquello que no nos interesa quedándonos con lo que nos puede resultar útil.

En ocasiones es interesante informarnos sobre un tema en concreto, pero podemos perder mucho tiempo mirando documentos que no hablan de lo que exactamente nos interesa, por eso sería muy útil poder hacer búsquedas en un corpus de documentos ya existente, dentro un tema en concreto mirando para ello sólo en los documentos que nos interesan.

Puede llevar mucho tiempo hacer una preclasificación, es decir saber de qué hablan exactamente los documentos y cual es la relación entre ellos, tener una primera visión del conjunto de los documentos.

Por lo tanto es interesante tener una herramienta que gestione automáticamente documentos de texto basándose en su contenido, independientemente del número de archivos, de la similaridad entre ellos mismos, de los diversos temas que puedan tratar, proporcionando además compatibilidades con varios tipos de archivos existentes, los más comunes en el almacenamiento de los archivos.

1.2 Objetivos

Por todo esto se pensó en desarrollar una herramienta que facilitase un primer acercamiento a un conjunto de documentos que puede resultarnos conocido, o totalmente nuevo.

Se ha desarrollado una herramienta que se encargue de todos los aspectos necesarios para un tratamiento de la información, desde sus formatos originales en archivos almacenados en el disco duro, pasando por un preprocesado de los datos, consistente en preprocesar y filtrar la gran cantidad de información de la entrada, gestionando el conjunto de los documentos para que resultase transparente para el usuario, basándose para ello en técnicas de clasificación de documentos y algoritmos de descubrimiento de patrones ocultos, y por supuesto una interfaz con el usuario para mostrar los datos finales e interactuar con los requisitos del usuario.

Una de las ventajas de esta idea radica en la posibilidad de trabajar con conjuntos de documentos no etiquetados, es decir sin conocer la correspondencia entre los documentos, y las categorías a las que pertenecen, lo que lo hace válido para cualquier situación.

1.3 Organización del contenido

- Este primer capítulo contiene una introducción al tema del proyecto, mostrando lo que se desea conseguir y justificando el tema abordado. Se incluye también la explicación de los objetivos del proyecto y la descripción de los contenidos de cada uno de los capítulos.
- En el segundo capítulo se muestran de manera ilustrativa, las técnicas de funcionamiento para los modelos de recuperación de la información.
- El tercer capítulo contiene una descripción de manera teórica del algoritmo de factorización de matrices no negativas, utilizado para reducir la dimensionalidad de los datos, además de comentar las aplicaciones más habituales de esta técnica.
- En el cuarto capítulo se describe el algoritmo de clustering K-means utilizado para realizar agrupaciones de los datos que contienen relaciones entre los documentos y los temas de los que tratan. También se describe el funcionamiento teórico del agrupamiento jerárquico utilizado para generar una estructura dentro de los datos obtenidos anteriormente.
- En el quinto capítulo se establecen las bases teóricas del escalamiento multidimensional, utilizado para obtener una representación de las distancias entre cada uno de los documentos.
- El sexto capítulo describe la metodología utilizada para resolver los problemas propuestos, así como un análisis exhaustivo de todos datos que se han obtenido en

cada una de las partes mencionadas anteriormente y una comparación entre los métodos utilizados.

- En el capítulo séptimo se exponen las conclusiones generales del Proyecto, vistas desde el punto de vista técnico. Se incluye un apartado en el que se comentan las futuras líneas de investigación y posibles mejoras para complementar el presente proyecto.

- El último capítulo es la Bibliografía, en el cual se hace referencia a todas las fuentes de información utilizadas para la realización de este Proyecto.

1.4 Estado del arte

Una idea similar es el buscador *Clusty* utiliza inteligencia artificial para agrupar las páginas y organizarlas como un árbol jerárquico, a su vez, las ramas son desplegables, lo cual nos permite ir profundizando en los resultados seleccionados de una forma bastante natural, lo que facilita que se pueda encontrar la información con mayor facilidad y rapidez.

El funcionamiento sería algo similar al nuestro en el sentido de que cuando se introduce una palabra en el buscador, se devuelven un número determinado de documentos, y esos documentos se clasifican en grupos, la diferencia sería que los documentos en nuestro caso se hallan en el disco duro de nuestra máquina, y en caso del buscador al pinchar en un link, nos llevaría a la dirección del documento de origen.

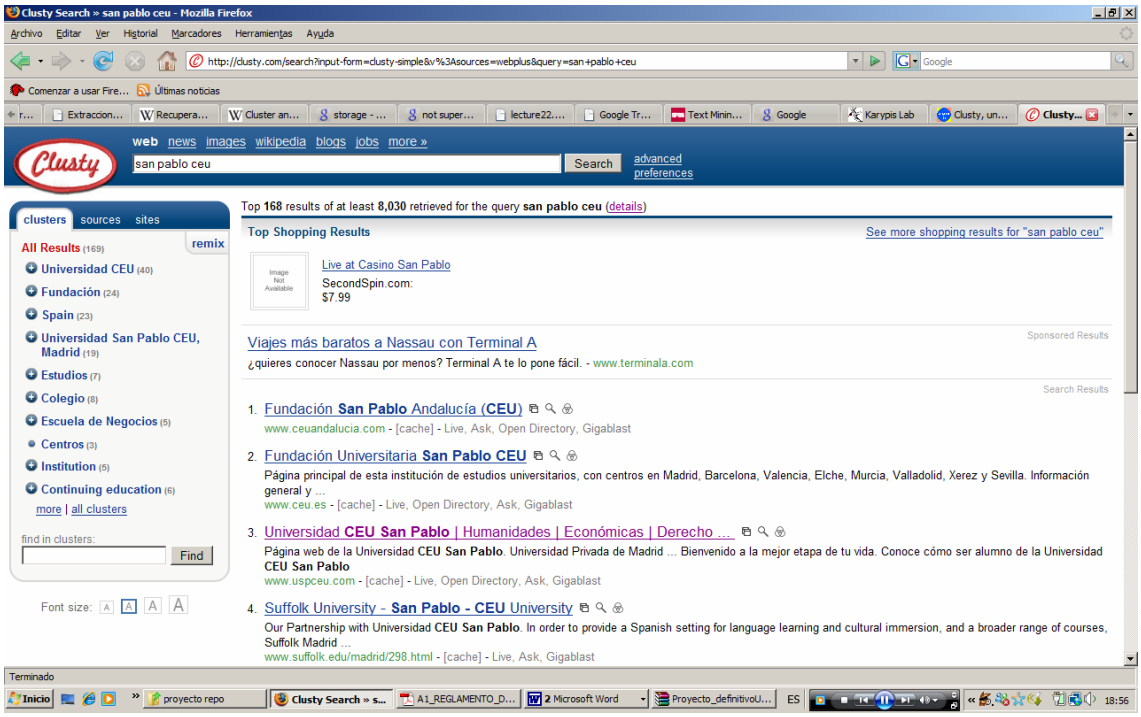

Ilustración 1.1 Buscador Clusty

El software más parecido a nuestros objetivos pertenece a gCluto desarrollado por George Karypis, miembro del DTC *Digital Technology Center*, este software se encarga de hacer grupos a partir de matrices de datos, es utilizado principalmente para investigaciones relacionadas con los genes, donde se pueden ver los elementos asociados a unas montañas que representan los grupos.

La diferencia con nuestro programa, sería que este software no puede trabajar con palabras de documentos porque no esta diseñado para procesar este tipo de datos.

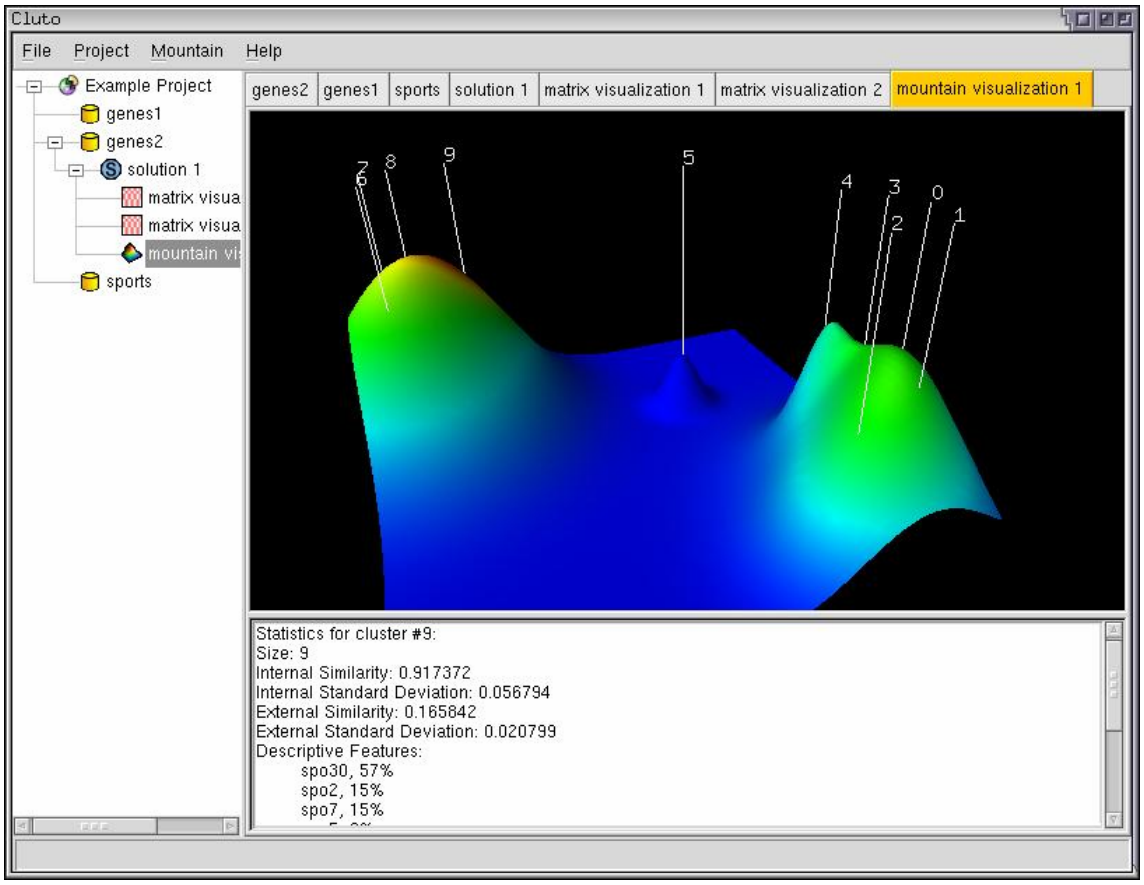

Ilustración 1.2 Programa que realiza agrupamientos de datos de entrada

Otro proyecto similar es WEBSOM, que es un método para organizar de manera automática las colecciones de documentos de texto mediante la generación de mapas visuales a partir de los documentos con el objetivo de facilitar la minería y la recuperación de la información.

Los documentos se encuentran situados en los puntos del mapa, y es posible realizar búsquedas a partir de su contenido seleccionando con el ratón los puntos visibles en el nivel más bajo de la pantalla del mapa.

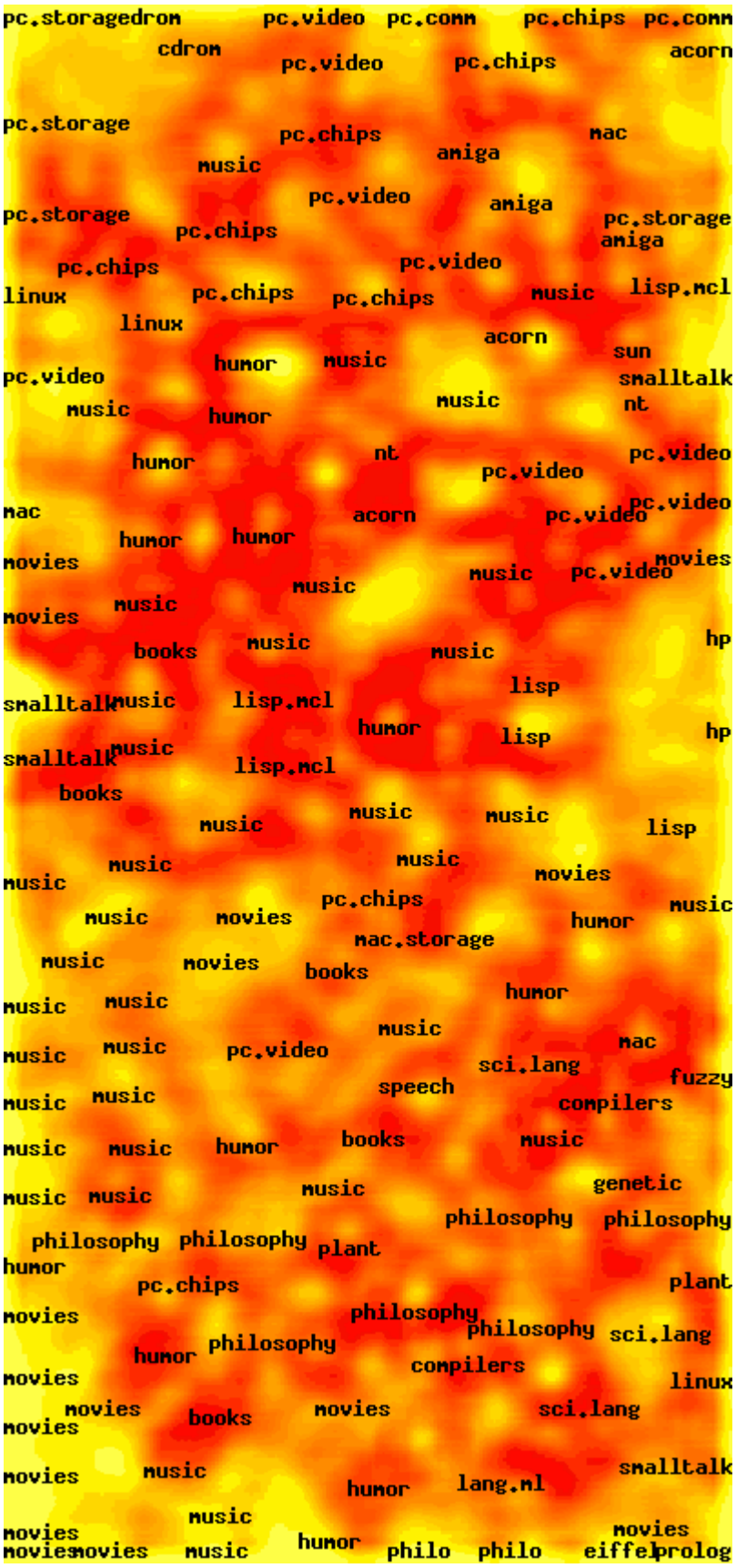

Ilustración 1.3 Ejemplo de mapa de contenidos

La interfaz con el usuario es activa, ya que se permite pinchar en cualquier área en el mapa para obtener una vista ampliada. El color hace referencia a la densidad o la tendencia de agrupación de los documentos, las zonas de color amarillo son las agrupaciones de documentos, y las zonas oscuras son los espacios vacíos entre los grupos.

Un proyecto muy relacionado con el nuestro es *KartOO*, que es un meta-buscador de información web que presenta sus resultados en forma de mapas. Los sitios encontrados después de realizar la búsqueda son representados por imágenes de documentos de diferente tamaño según su importancia. Es posible realizar una nueva búsqueda a partir de los temas y expresiones propuestos.

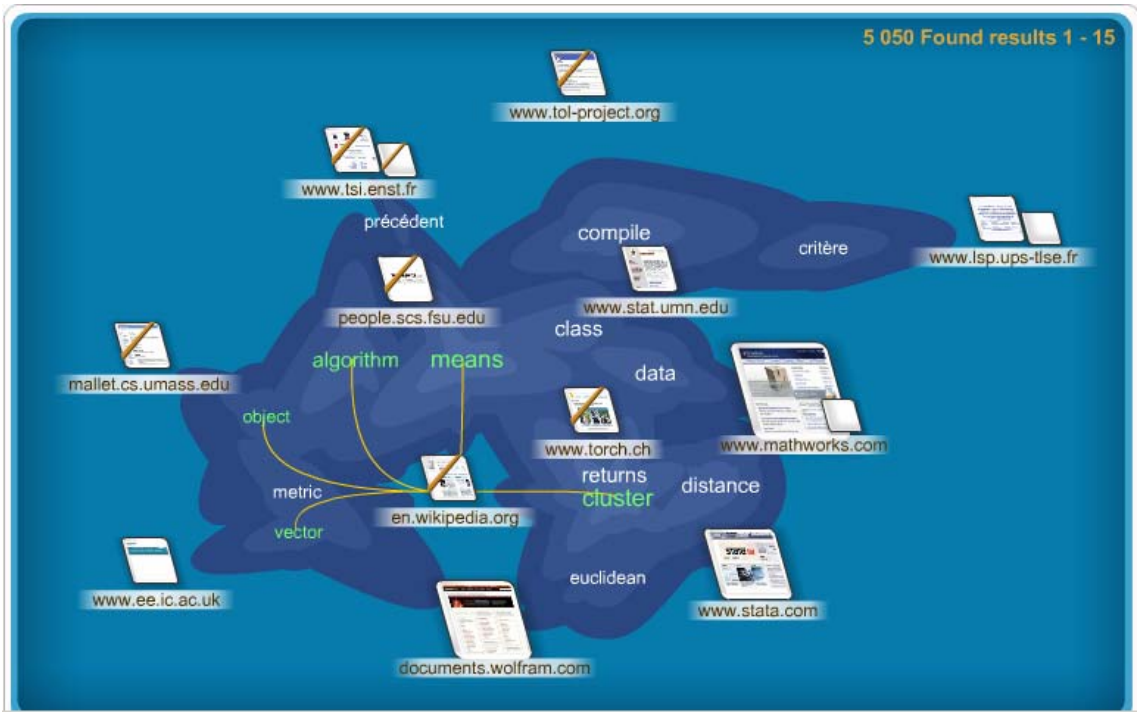

Ilustración 1.4 Resultado de buscador Kartoo

Cuando se realiza una búsqueda y se pasa el ratón por encima de un documento, se crean unas flechas desde el documento seleccionado, hacia las palabras importantes de ese documento.

Se agrupan los temas relacionados con la búsqueda que se ha efectuado, en este caso *Kmeans* como se puede ver en la siguiente imagen.

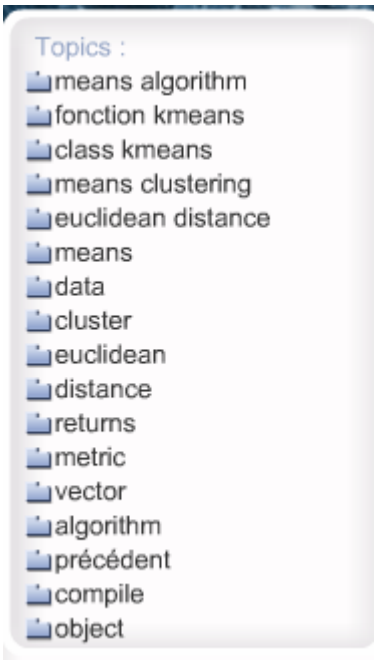

Ilustración 1.5 Temas relacionados

Un proyecto que contiene muchas funcionalidades es el buscador *Grokker*, que se encarga de recuperar los resultados de determinadas fuentes como pueden ser *Wikipedia*, *Yahoo*, *Amazon Books*, que pueden ser seleccionadas de manera independiente, o simultáneamente para encontrar más cantidad de resultados, los resultados se organizan por temas, y se muestran en un mapa.

El mapa muestra las categorías generadas representadas de forma abstracta en una imagen donde los círculos corresponden a las categorías y lo que hay dentro son enlaces y documentos. Es posible explorar el contenido de manera visual dentro de una categoría haciendo clic en un vínculo como se puede ver en la ilustración 1.7.

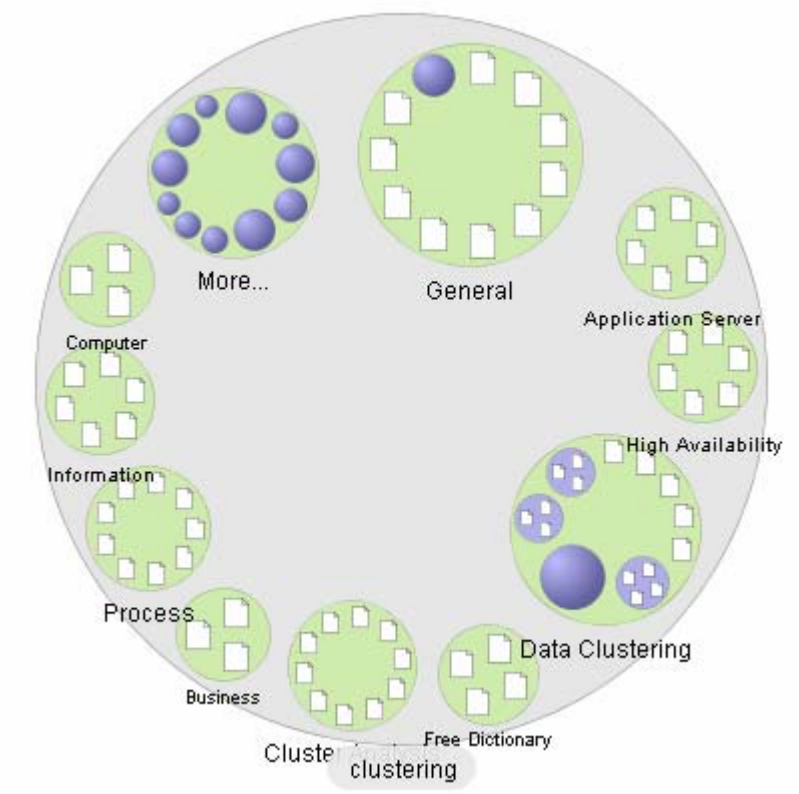

Ilustración 1.6 Mapa de relaciones

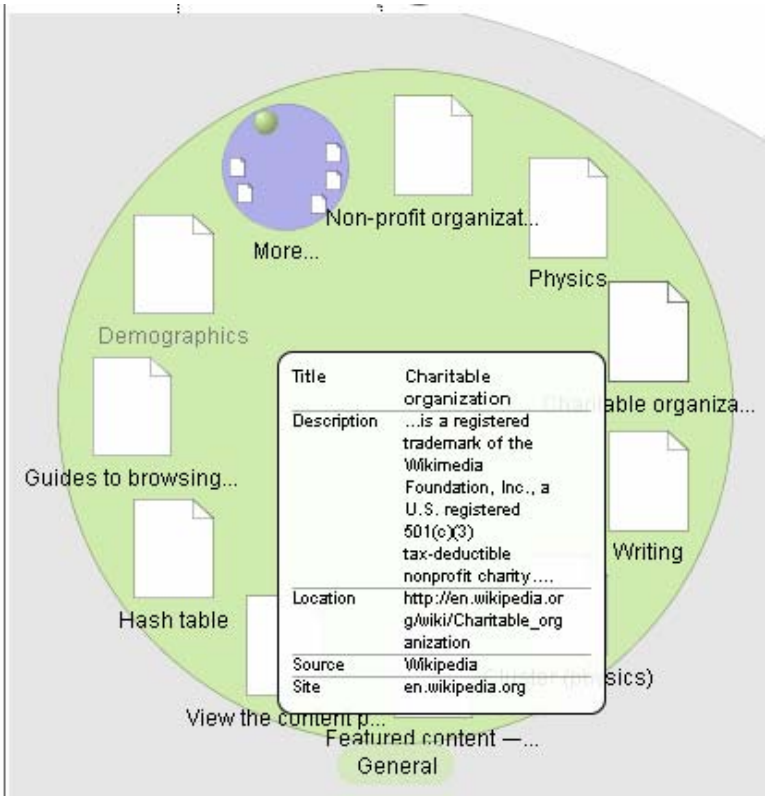

Ilustración 1.7 Ampliación sobre el mapa

A partir de los resultados obtenidos tras realizar la primera búsqueda, se muestran las categorías y subcategorías, en el caso de que existan, en un panel central sobre el cual se puede ver su contenido pinchando sobre la categoría seleccionada sobre el panel derecho.

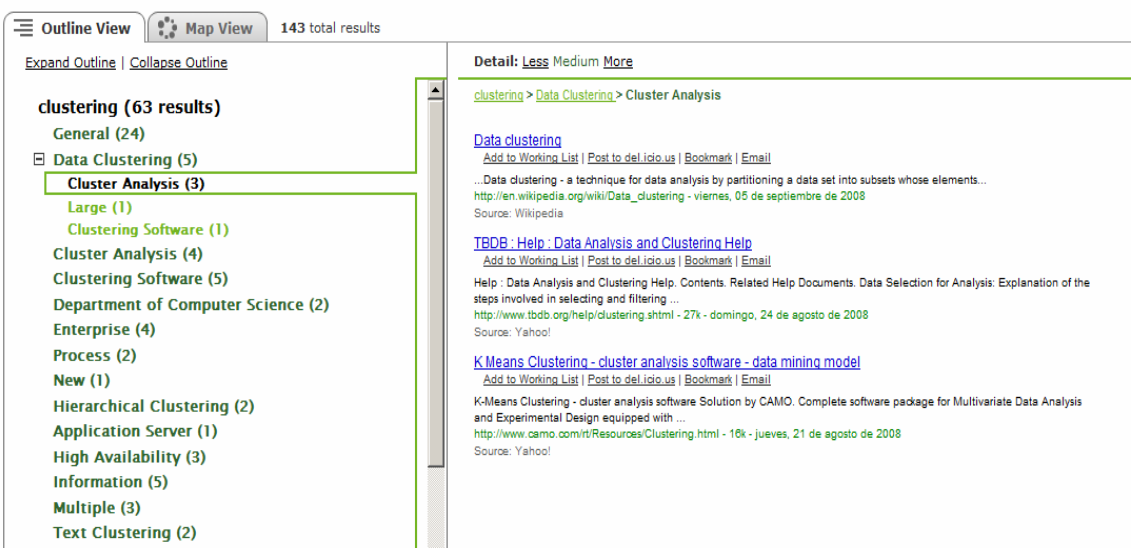

Ilustración 1.8 Panel de categorías y sus contenidos

Existe una opción para seleccionar los documentos de más reciente creación dentro de una categoría, lo que reduce la cantidad de documentos encontrados en función de la exigencia de la fecha de inicio.

2 ANTECEDENTES Y ESTADO DE LA CUESTIÓN

2.1 Introducción

La información se puede almacenar en forma de datos estructurados, es decir aquellos atributos o variables fuertemente tipificados, por ejemplo los tipos int, float o string. Cada atributo en una relación está definido para todos los registros, existiendo un tipo de orden dentro de los datos. Tenemos como ejemplos de registros las bases de datos relacionales:

Cliente (char)	Fecha Transacción (date)	Cantidad (int)
Temp.S.A.	05/02/1998	10.000
Servi2.S.L.	22/07/2005	50.000

Tabla 2.1 Ejemplo registro de bases de datos relacionales

Pero existe una gran diferencia con otro tipo de datos que no tienen este tipo de estructura pero que pueden ser mucho resultar mucho más valiosos en diversos contextos.

Tenemos por ejemplo una carta que muestra la baja de un cliente, pero que no está almacenada en una base de datos de manera estructurada, y que posee una información que puede resultar de una utilidad muy grande para una empresa.

Madrid a 23 de Marzo de 2007

Con la presente se notifica la baja del cliente A_Consulting debido a un incumplimiento del contrato por su parte.

Nuestros abogados se pondrán en contacto con ustedes para solucionar nuestras diferencias.

Un saludo.

Como se puede observar no existe una manera automática de poder analizar este dato para hacer posteriormente consultas automatizadas.

2.2 Acceso a la información

Es necesario entender la diferencia entre los datos, la información, el conocimiento y la sabiduría.

Los datos son una representación física de la realidad, estamos acostumbrados a tratarlos en diversos formatos como pueden ser numéricos, palabras, letras, incluso sonidos o videos, habitualmente es lo que tratan los ordenadores por su simplicidad y velocidad de procesamiento.

Subiendo un escalón por encima y trabajando sobre esos datos tenemos la información, que no son más que datos a los que se ha asociado un significado y que son más fácilmente comprensibles por las personas. Pueden estar almacenados en ordenadores, incluso ser generados por ellos, pero habitualmente son creados por personas, como ejemplos tenemos presentaciones gráficas, documentos, cartas.

Cuando esa información se organiza y se aplica a situaciones y problemas específicos estamos ante el conocimiento, como puede ser el presente documento que pretende dar solución a un problema concreto.

Por encima encontraríamos la sabiduría que es un conjunto de conocimientos aplicables a situaciones y problemas diversos, de mayor envergadura. Existen empresas que venden soluciones concretas basándose en el conocimiento, como pueden ser las empresas consultoras, es el exponente máximo del saber, del tratamiento de la información.

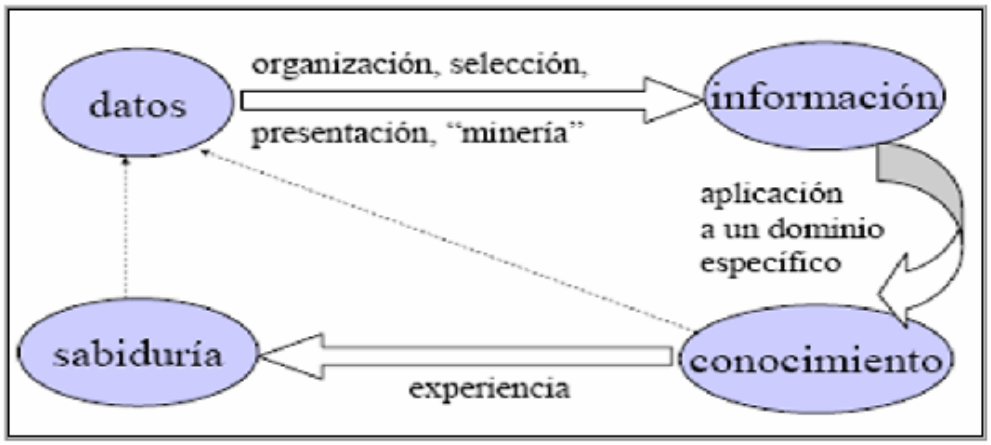

Ilustración 2.1 Relaciones de la información

Cada vez que subimos un nivel accedemos a una información cada vez más elaborada, pero que es más difícil su tratamiento de manera informática debido a una limitación de los ordenadores, que es debido a su falta de inteligencia que poco a poco se intenta solventar.

En ocasiones es necesario afrontar problemas reales para lo que se necesita tener un elevado conocimiento y si es posible sabiduría, el problema es la gran cantidad de tiempo que es necesario invertir para poseerlo.

2.3 Recuperación de la información

Relacionando los conceptos entre los tipos de información y de lo que se componen podemos generar la siguiente tabla:

Tipo de datos	Concepto
Estructurados	Datos
Semi-estructurados	Datos e información
No estructurados	Información

Tabla 2.2 Descripción de los tipos de datos

Lo que se pretende es acceder al conocimiento a partir de datos no estructurados que se almacenan como información y que es difícil de tratar por los medios habituales como pueden ser las consultas a las BBDD.

La situación de partida es un conjunto de documentos que están lo suficientemente elaborados como para ser considerados información o incluso conocimiento. Pero tenemos tanta cantidad de información que nos resulta difícil hacer una primera clasificación.

Un conjunto de documentos posee mucha información, pero lo que necesitamos conocer es una pequeña parte que se puede representar como se ve en la siguiente ilustración como una parte de todo aquello que tenemos delante.

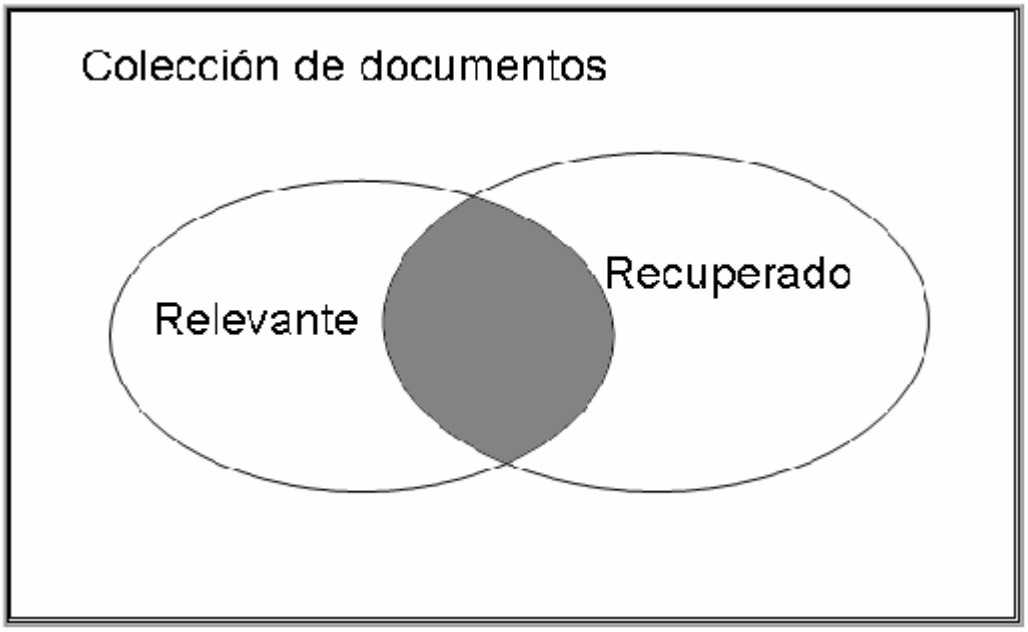

Ilustración 2.2 Recuperación de documentos

2.3.1 Definición

La IR es un estudio interdisciplinario que cubre muchas disciplinas generando normalmente un conocimiento parcial. Algunas de las disciplinas que se ocupan de estos estudios son la psicología cognitiva, la arquitectura de la información, diseño de la información, el comportamiento humano hacia la información, la lingüística, la semiótica, informática, biblioteconomía y documentación.

Encontramos ejemplos aplicados de esta disciplina como pueden ser los buscadores que todos conocemos como Google o Lycos, son algunas de las aplicaciones más populares de la recuperación de información. Su funcionamiento se trata en crear una lista de términos en lenguaje natural, además de un algoritmo que incluye las reglas lógicas de la búsqueda (tabla de verdad) y una valoración de los resultados o cantidad de información lograda o posible. Este motor de búsqueda es pues el que permite formular nuestra consulta.

2.4 Modelos de Recuperación

Los modelos de recuperación son una de las mejores herramientas para poder realizar una búsqueda de lo que se ha consultado dentro del conjunto de documentos del que queremos realizar una consulta. Para obtener los documentos más relevantes a la

consulta realizada, existen varios tipos de modelos de recuperación. Los modelos existentes se pueden clasificar en los siguientes:

Modelos de recuperación clásicos:

- Modelo vectorial
- Modelo booleano
- Modelo probabilístico

Otros modelos:

- Relevance feedback
- Basado en el lenguaje
- Redes de inferencia
- Lógica difusa

2.4.1 Modelo de vectorial

También conocido como modelo de espacio vectorial. Los documentos son representados utilizando un vector en el que se recogen las relaciones existentes entre el documento y sus características, en nuestro caso es un valor ponderado que refleja la importancia de cada palabra del documento con respecto al resto de los documentos.

Con estos datos se realiza la representación vectorial, que será usada en las consultas para recuperar la información. La forma de recuperar la información consiste en comparar este vector con los vectores de los documentos, utilizando una función de similitud. El grado de similitud varía según la consulta que se realice. Cuanto mayor es el grado, se considera que más se ajusta a la petición.

Con este modelo se pueden obtener los documentos de forma ordenada y se puede limitar el número de resultados si se considera un grado de similitud mínimo.

2.4.2 Modelo booleano

Se trata de uno de los modelos de recuperación de información más simples que se conocen. Se fundamenta en el álgebra de Boole y en la teoría de conjuntos. Este modelo crea una expresión booleana para formalizar la consulta. Esta expresión utiliza los operadores booleanos AND, OR y NOT.

A la hora de recuperar la información, un documento tendrá más relevancia que otro teniendo en cuenta si una palabra está presente o no es decir:

- Si se encuentra la palabra: La contiene.
- Si se encuentran las dos palabras: palabra1 AND palabra2.
- Si se encuentra una sí y otra no: palabra1 AND NOT palabra2.
- Si se encuentra o una o la otra: palabra1 OR palabra2.

Dependiendo de los operadores booleanos que unan las palabras a buscar, se recuperarán unos documentos u otros, puesto que no es lo mismo buscar palabra1 AND palabra2 (tiene que aparecer ambas) que buscar palabra1 OR palabra2 (aparece o una o la otra).

El problema de este modelo es que si encuentra una serie de documentos, no sabe ordenarlos según la relevancia que tenga cada uno. Para solucionarlo se puede utilizar el modelo booleano extendido, que añade pesos a las palabras buscadas, lo que le lleva a aproximarse a un modelo vectorial.

2.4.3 Modelo Probabilístico

Este modelo se fundamenta en el cálculo de la probabilidad de que el documento sea relevante para la consulta realizada. Por tanto si cogemos un documento cualquiera entre un conjunto de n documentos, existe una cierta probabilidad de que dicho documento sea relevante para la pregunta realizada.

Se puede calcular la probabilidad de la siguiente manera:

Probabilidad(relevancia) = n / N, donde n es el conjunto de documentos relevantes y N es el conjunto de todos los documentos.

Para calcular la relevancia, se utilizan una serie de pesos dados a las características del documento. Para saber la relevancia, se usan índices de los términos que se conocen como descriptores con los pesos que se han establecido. Con esto se pretende recuperar

los documentos en los que existen los mejores descriptores de los que el usa en la consulta.

Al igual que en el caso del modelo vectorial se pueden ordenar los resultados obtenidos en función del grado de relevancia debido a que se usan pesos.

Este modelo necesita de una hipótesis inicial de independencia en la distribución de los términos en documentos relevantes y de independencia en la distribución de todos los documentos, para que se puedan establecer los documentos relevantes así como los pesos.

Se contabiliza el número de términos que aparecen y los supone independientes, esto hace que el cálculo de las probabilidades sea complejo.

2.4.4 Relevance feedback

Se realiza una búsqueda inicial donde se obtiene un conjunto de documentos que podemos denominar como relevantes, y el modelo vuelve a formular la consulta introducida por el usuario en este conjunto de documentos que son más cercanos a los intereses del usuario, además se vuelven a recalcular los pesos de los términos relevantes.

Se persigue conseguir que cada vez obtengamos resultados más relevantes, por lo que vamos descartando algunos resultados.

La modificación de la consulta se puede hacer de dos formas: manual o automática. En la manual es el usuario quien decide cuáles son los documentos más relevantes. En la automática se eligen asumiendo que los n primeros son los 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 la 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.

Mediante esta técnica se pretende llevar la consulta realizada hacia los documentos relevantes. Los resultados de este modelo son muy buenos ya que mejora en un alto grado la recuperación de documentos relevantes para la consulta realizada.

Sin embargo, si existe una palabra mal escogida en la consulta hará que los resultados sean peores.

2.4.5 Modelo basado en el lenguaje

Todos los modelos mencionados anteriormente (vectorial, booleano, probabilístico y relevance feedback) son muy usados, sin embargo hoy en día se está dando mucha importancia al procesamiento del lenguaje natural. Por tanto otro de los modelos a tener en cuenta es el que está basado en el lenguaje. Estos modelos se basan en una serie de conocimientos para conseguir descifrar e interpretar textos, así como obtener un listado de descriptores de forma automática.

El lenguaje natural es el más ambiguo de todos los lenguajes y cada palabra según el contexto en el que se encuentra, puede significar gran variedad de cosas. Por eso se ayuda de lenguajes documentales de representación del conocimiento, como los tesauros o las ontologías para tratar de descifrar el lenguaje natural. Si esto se relaciona con los documentos de la Web, se puede comprobar cómo existen otras técnicas (metadatos o lenguajes semánticos) para poder representar el conocimiento que contienen y poder recuperar información. Dentro de los lenguajes semánticos, sin duda el más conocido es XML (eXternal Markup Language).

Si todos los documentos de la Web estuviesen estructurados, el proceso de recuperación de la información sería rápido y sencillo, pero esto por desgracia no ocurre así y existe un gran porcentaje de documentos desestructurados.

2.4.6 Modelo basado en redes de inferencia

Dentro de una red de inferencia se pueden distinguir dos redes que la componen: red de consulta y red de documentos.

Cuando el usuario realiza su consulta se construye la red de consulta. Esta red tiene dos tipos de nodos: de consulta y de términos de los documentos. De cada nodo de término salen arcos orientados, que lo conectarán con los nodos de consulta correspondientes.

En cuanto a la red de documentos, se trata de una red fija. Se compone por dos tipos de nodos: de términos y de documentos. Estos nodos se corresponden a los términos de los documentos y a los documentos en sí respectivamente. Por cada nodo de tipo documento, salen arcos que los relacionan con los términos indexados.

Puesto que proviene del modelo probabilístico, el siguiente paso es calcular las probabilidades y una vez que se han estimado se realiza la inferencia, para lo cual se instancia cada documento de manera sucesiva, y se calcula la probabilidad de que la consulta sea satisfecha con ese documento instanciado.

Este modelo introduce una serie de variables aleatorias, que representan si la información requerida ha sido satisfecha, estas variables aleatorias son binarias. Se considera un documento como relevante por la cantidad de apoyo evidencial que una observación de a la consulta.

Que un documento cualquiera sea relevante, viene determinado por la relación que una determinada observación tiene con una consulta, se representa de la siguiente manera:

$$P(q \wedge d_j) = \sum_{\forall k} P(q \wedge d_j \mid k) P(k)$$

2.4.7 Modelo basado en lógica difusa

Los modelos que se basan en las probabilidades, deben realizar una estimación inicial de los pesos, y esto los hace bastante más complejos y más difíciles de implementar. El modelo de lógica difusa no asigna el peso (en este caso llamado grado de pertenencia) en un primer momento.

Es habitual hacer una comparación entre el modelo probabilístico y el modelo basado en lógica difusa. En éste modelo si el grado de pertenencia de los términos es elevado, aumenta la posibilidad de que ese término se encuentre en el documento con un mayor grado de relevancia, en cambio en el modelo probabilístico en el caso en el que unos términos se encuentren en el documento, estamos utilizando a la vez los documentos

relevantes y los no relevantes cuando calculamos las últimas probabilidades, lo que lo hace menos preciso.

Como ventaja adicional para este modelo, el uso de los modelos borrosos es muy recomendable para resolver problemas de imprecisión en el indexado de documentos.

3 REDUCCIÓN DE LA DIMENSIONALIDAD

3.1 Introducción

De manera genérica reducir la dimensionalidad de un conjunto de datos consiste en sintetizar una fuente de datos que habitualmente tiene unas dimensiones elevadas en un conjunto de datos más pequeño, esto es con menos dimensiones. Esto acarrea ventajas de almacenamiento, los datos reducidos ocuparan menos espacio en disco, además de ventajas de procesamiento, el tiempo invertido en cargar, modificar y guardar un conjunto de datos reducido, será mucho menor y podrá llevarse a cabo con una máquina de inferiores características a las que se requeriría con el conjunto inicial de los datos.

3.2 Sampling

Es una técnica muy simple, que se puede utilizar cuando tenemos una gran cantidad de información y puede mantener la estructura original facilitando el procesado consiste en muestrear.

Si bien esta técnica no reduce la dimensión de cada vector, sí que reduce la carga computacional de los métodos al eliminar vectores completos.

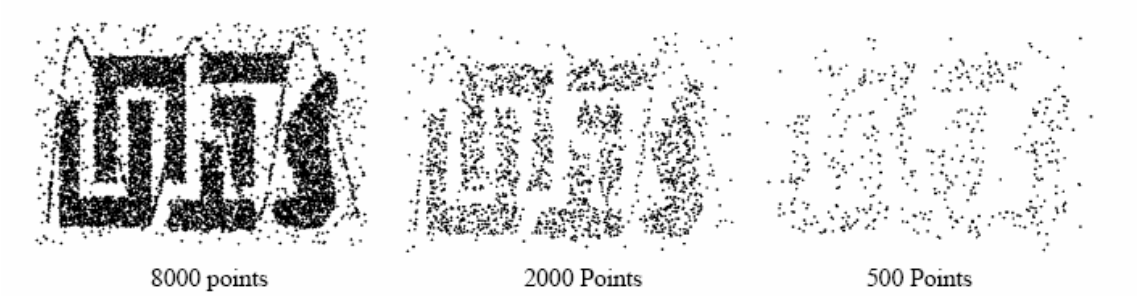

Ilustración 3.1 Muestreo de puntos

Este método consiste en coger muestras aleatorias del conjunto de los datos y suprimir estos valores con la finalidad de tener menos muestras totales, pero las suficientes para que se conserven la estructura y las propiedades del conjunto de datos original.

Cada muestra tiene la misma probabilidad de ser seleccionada para ser eliminada, también se suele partir el conjunto de las muestras total en grupos homogéneos y muestrear cada parte con la ventaja de que se mantienen las proporciones.

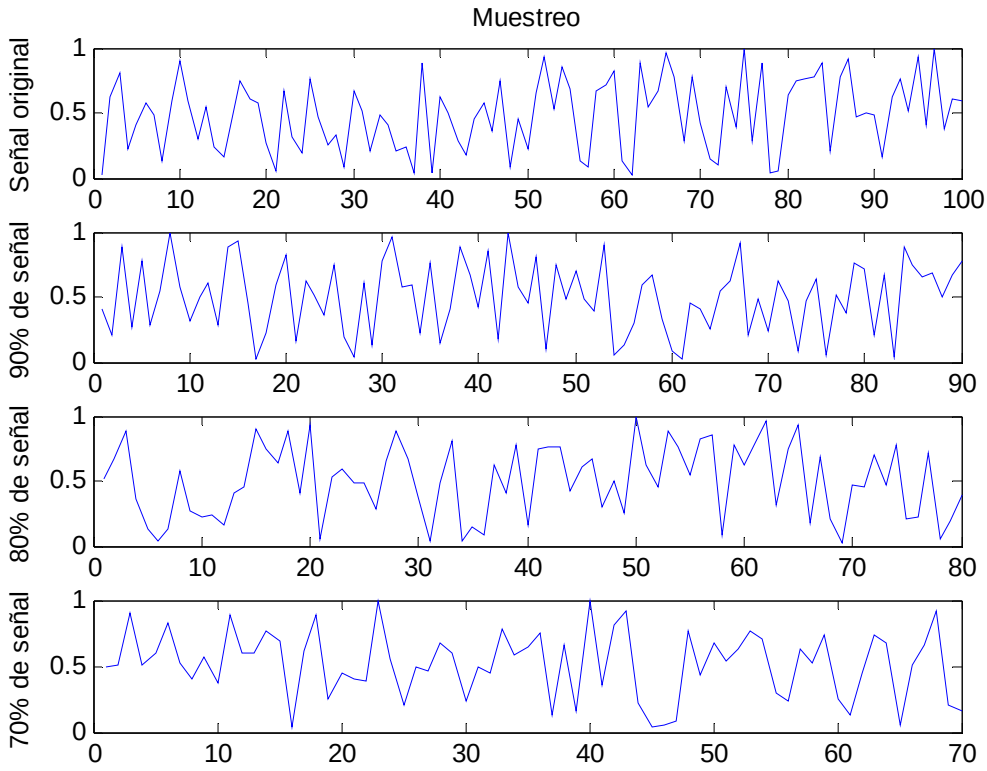

Ilustración 3.2 Resultado de tomar muestras en una señal

Se puede comprobar en la gráfica el deterioro que sufre una señal, cada vez que se quitan progresivamente los valores que la componen, llegando un punto en el que la señal pierde todo el parecido con la señal original convirtiéndose en inservible.

La desventaja de aplicar esta técnica a nuestro conjunto de datos, es que aunque tengamos matrices de más de 5 E6 datos, se pierde información muy valiosa y se rompería la estructura de la matriz de datos.

3.3 Cambio de escala de agregación de los datos

La técnica consiste en agregar los datos a una escala más grande ya sea temporal, espacial o jerárquica, para reducir la variabilidad de los datos, con lo que se consigue tener una reducción en los datos y se afecta a las distribuciones estadísticas.

Se puede comprobar en la siguiente gráfica un ejemplo de reducir unos datos de temperaturas medias registradas a lo largo de un año en la estación de Xunqueira, cambiando la escala temporal de una escala mensual a una escala trimestral. Se observa

como se conserva la estructura de los datos en la gráfica que contiene en el eje de abscisas trimestres en vez de meses.

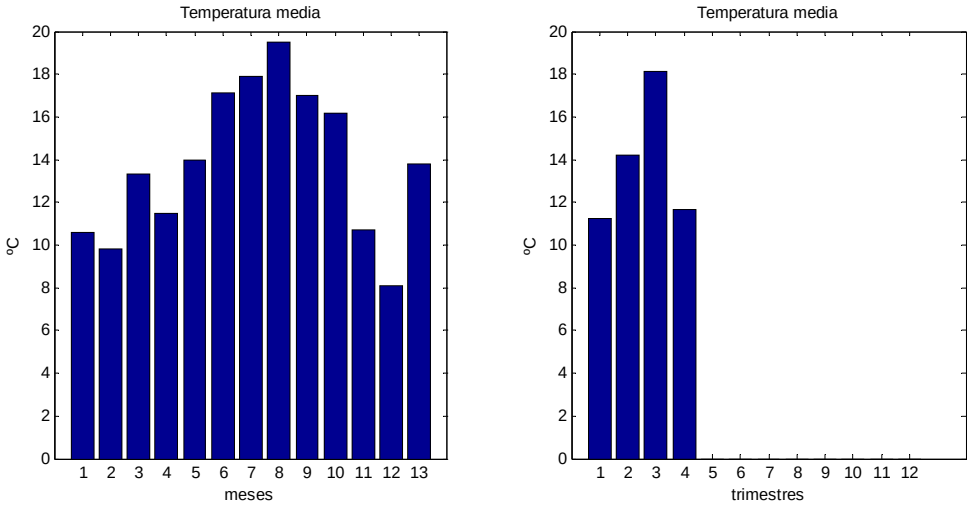

Ilustración 3.3 Resultado de cambio de escala de agregación de datos

Este método permite conseguir buenos resultados que suelen ser fácilmente interpretables al observarlos, el problema en nuestro caso es que al cambiar de escala se pierde demasiada información que es útil, como por ejemplo la relación entre los documentos y los temas.

3.4 Factorización de matrices no negativas, NMF

La factorización de matrices no negativas es una técnica de extracción de características de muy reciente desarrollo, que intenta obtener una representación local o *parts-based* de datos no negativos. Dada una matriz de datos no negativa, NMF intenta obtener su factorización en dos matrices no negativas W y H:

$$X \approx WH$$

3.4.1 El modelo NMF

NMF es un tipo de representación lineal y no negativa de un conjunto de datos. Si nuestros datos consisten en medidas o vectores de variables escalares no negativas, entonces los datos se pueden agrupar en la matriz $X \in \Re^{M,N}$ siendo cada columna de la

matriz una medida o vector x_j, donde $j = 1,...,N$. Una aproximación lineal de los datos viene dada por:

$$x_j \approx \sum_{i=1}^{K} w_i h_{ij} = W h_j$$

Siendo W es una matriz de tamaño MxK que contiene a los vectores base w_i en sus columnas. Los K vectores base w_i pueden considerarse como los bloques descriptores de los datos, y el vector de coeficientes h_j (de dimensión K) informa de la mayor o menor importancia de cada bloque descriptor en el vector de datos original x_j donde cada columna de H contiene al vector de coeficientes h_j correspondiente al vector de datos x_j. Se observa en la expresión que una representación lineal de los datos es simplemente una factorización de la matriz de datos. Las técnicas de PCA, ICA y NMF pueden considerarse todas ellas como factorizaciones de matrices con diferentes elecciones en la función de coste o en las restricciones que imponen en la representación.

NMF impone que todos los valores de ambas matrices sean no negativos, lo que significa que los datos son descritos utilizando sólo componentes aditivas. Esta restricción impuesta por NMF está motivada principalmente por la consideración de que en la mayoría de los sistemas físicos reales las cantidades implicadas no pueden ser negativas.

En la representación NMF, dada la matriz de datos X, las matrices W y H se definen como aquellas matrices no negativas que minimizan el error de reconstrucción entre X y WH. La función más utilizada es la de error cuadrático (distancia euclídea):

$$E(W,H) = \|X - WH\|^2 = \sum_{i,j} \left(x_{ij} - (WH)_{ij} \right)^2$$

La función objetivo que dirige el proceso de descomposición y que es necesario minimizar, está basada en la de verosimilitud de Poisson y se puede definir de la siguiente manera:

$$D(V,WH) = \sum_{i=1}^{N} \sum_{j=1}^{M} \left(V_{ij} \ln \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right)$$

De esta ecuación se deriva un algoritmo iterativo que empieza cuando se inicializan las matrices W y H con valores aleatorios positivos.

Para cada vector base $W_p \in \mathbb{R}^{nx1}$ se actualiza el correspondiente vector codificante $H_p \in \mathbb{R}^{1xm}$ y a continuación se actualiza y normaliza el vector base W_p, repitiendo este proceso hasta que se alcanza la convergencia.

3.4.2 Restricciones

Adicionalmente, se suelen imponer las siguientes restricciones:

- El número de bases o en nuestro caso el de temas que vamos a obtener tiene que ser mucho menor que la dimensión más grande que vamos a tener en la matriz de entrada de datos X, que en este caso será el número de vectores.
- V, W y H no tienen valores negativos.
- Las columnas de W están normalizadas (la suma de los elementos de cada columna vale 1).

3.4.3 Criterio de parada

En cada iteración se calcula la función objetivo, y se compara con el valor de esta en la iteración anterior, un criterio de parada puede ser que esta diferencia sea menor que un cierto porcentaje, o hacer un determinado número de iteraciones internas del algoritmo.

En el siguiente gráfico se muestra que para 100 iteraciones el algoritmo va convergiendo, empezando con valores muy elevados, y en poco tiempo logrando disminuir, es la variable obj descrita en la fórmula anterior.

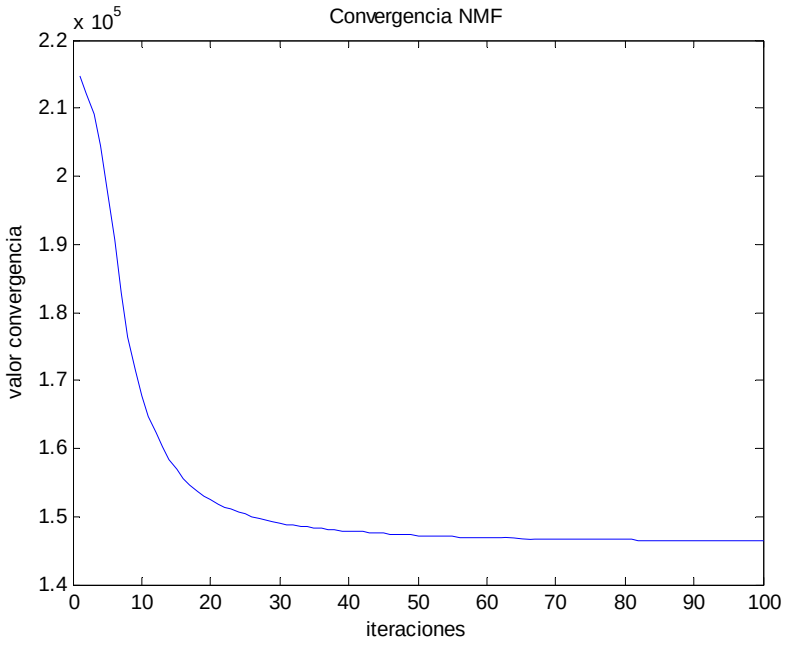

Ilustración 3.4 Convergencia del algoritmo NMF

Cuando restamos la variable obj de la iteración i, con la de la iteración i-1 se muestra que los cambios que se producen en los valores de obj, son cada vez más pequeños porque va convergiendo más lentamente en función del avance de las iteraciones del algoritmo NMF, esto se muestra en la siguiente gráfica.

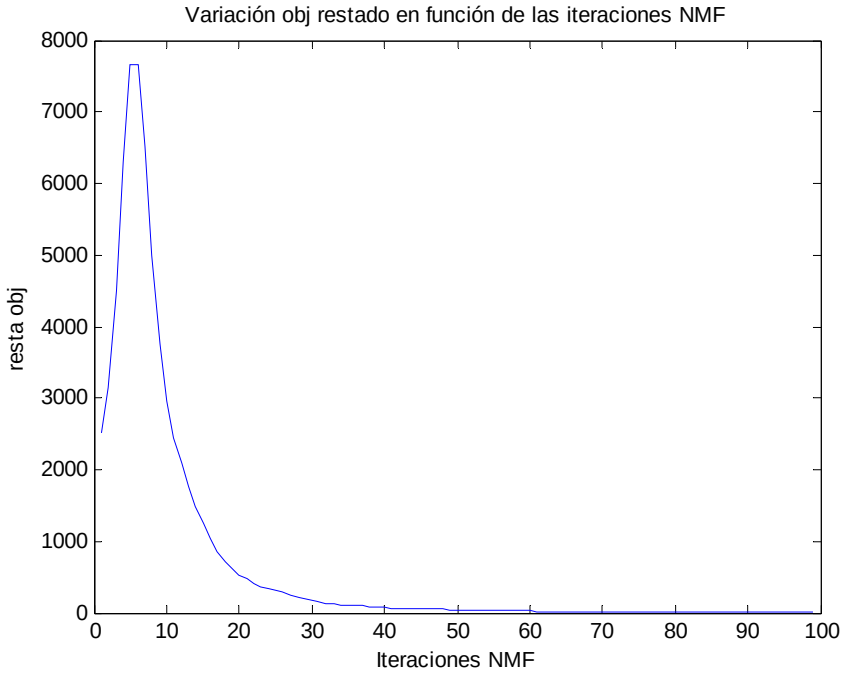

Ilustración 3.5 Tasa de trabajo del algoritmo NMF

También se observa que el algoritmo hace un mayor esfuerzo por comprimir los datos en las primeras iteraciones, cuando vamos en torno a las 10 iteraciones, y que es más difícil conseguir cambios a medida que el número de iteraciones se hace grande.

3.4.4 Aplicaciones

Esta técnica se puede aplicar al análisis exploratorio de datos como método de proyección para reducir la dimensionalidad de los datos o para descubrir patrones ocultos, aunque su aplicación más extendida es para facilitar la interpretación de los datos.

El éxito de la representación NMF radica en su capacidad de obtener características significativas de colecciones de datos reales.

Cuando X es una colección de imágenes de caras, la representación NMF genera vectores base que muestran al visualizarlos como imágenes las características intuitivas propias de las caras, esto es, ojos, boca, nariz, como se puede comprobar en la siguiente imagen.

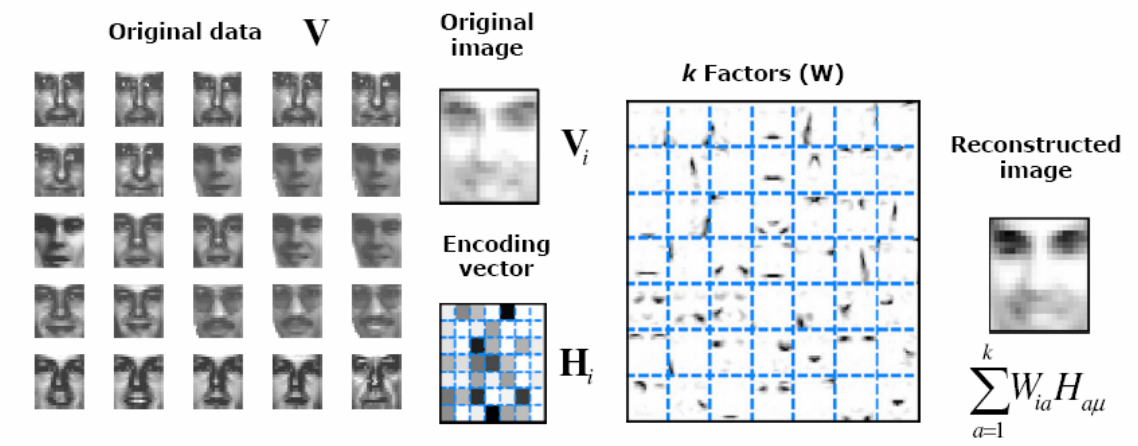

Ilustración 3.6 Descubrimiento de patrones con imágenes de caras

Es particularmente interesante para descomponer espectrogramas de señales de voz, donde se verifica la existencia de patrones espectrales que se repiten a lo largo de las ventanas de análisis.

3.4.5 Ventajas NMF

Se pueden descubrir patrones ocultos y además medir las relaciones entre ellos aprovechando la magnitud de los valores de NMF.

Permite optimizar los algoritmos que se ejecuten utilizando los datos de salida de este porque reducen la dimensionalidad de los datos sin alterar el resultado para realizar una clasificación. Debido a sus múltiples ventajas lo utilizaremos como algoritmo de reducción de la dimensionalidad.

3.4.6 Reconstrucción de los datos

A partir de las matrices ya factorizadas, se puede volver a reconstruir la matriz original de una forma aproximada.

$$\hat{V} = WH$$

3.4.7 Pseudocódigo

Función NMF(Entrada matriz de datos $V \in \mathbb{R}^{nxm}$, dimensión de reducción r)

bases $W \in \mathbb{R}^{nxr}$, $H \in \mathbb{R}^{rxm}$

W, H se inicializan aleatoriamente con valores $w,h \in [0,1]$

preprocesado D;

```
while (no haya convergencia) {
     WH   se calcula el producto de matrices
     // Recalcular elementos de H
     for (a = 0; a < R; a++) {
           for (i = 0; i < M; i++) {
```

$$H_{ai} \leftarrow H_{ai} \frac{\left(W^T D\right)_{ai}}{\left(W^T WH\right)_{ai}}$$

```
           }
     }
```

```
      // Recalcular elementos de W
      for (i = 0; i < N; i++) {
            for (a = 0; a < R; a++) {
```

$$W_{ia} \leftarrow W_{ia} \frac{\left(DH^T\right)_{ia}}{\left(WHH^T\right)_{ia}}$$

```
            }
      }
}

}
```

4 CLUSTERING

4.1 K-means

4.1.1 Introducción

El método que McQueen propuso en el año 1967 es conocido como k-medias, y es el método de clustering particional más utilizado debido a su simplicidad su rapidez y su eficiencia.

Se parte de una selección inicial de k centroides, que serán considerados como los representantes de cada grupo de puntos, y se asigna cada uno de los elementos de la colección al grupo con el centroide más cercano. A continuación, se calcula el centroide de cada uno de los grupos resultantes, se observa que en los primeros pasos se obtienen las mayores diferencias entre los centroides originales y los calculados en las reasignaciones. Los puntos de la colección vuelven a asignarse al grupo del centroide más cercano, y estos pasos se repiten hasta que los k centroides no cambian de grupo en una iteración (esto es equivalente a decir que el valor de la función utilizada como criterio de optimización no varía).

El algoritmo K-Means es mucho más eficiente que los métodos jerárquicos porque los tiempos de cómputo requeridos son lineales con la cantidad documentos a agrupar, pero es dependiente de la selección inicial de centroides. Sus resultados suelen variar mucho si se aplica varias veces a la misma colección de documentos, ya que si la selección de centroides al azar es mala, la solución encontrada no será la más óptima, para remediar esta situación de aleatoriedad se han tomando las medidas pertinentes promediando los resultados hasta obtener menos variabilidad y una tendencia clara a un mismo resultado.

Habitualmente el algoritmo K-means converge a las pocas iteraciones, y se suele utilizar como condición de parada que pocos puntos cambian de cluster.

La distancia que se utiliza en el algoritmo para medir las distancias entre los elementos es la distancia euclídea.

4.1.2 Etapas

- Se consideran los k primeros elementos de la matriz de datos de entrada como k conglomerados o centroides con un único elemento, con la particularidad de que se van a permutar estos elementos para que el inicio de los primeros centroides sea diferente cada vez.

- Se asigna en el orden de la matriz de datos de entrada cada uno de los puntos al centroide más próximo.

- Después de cada asignación se recalculará el nuevo centroide.

- Cuando todos los objetos hayan sido asignados en el paso anterior, se calculan los centroides de los conglomerados obtenidos, y se reasigna cada objeto al centroide más cercano.

- Se Repiten los pasos 2 y 3 hasta que se alcance un determinado criterio de parada.

4.1.3 Limitaciones

Es importante tener en cuenta que el algoritmo de McQueen es sensible al orden con el que se encuentran los objetos en la matriz de datos de entrada, y fundamentalmente es sensible a los objetos que se encuentran en las K primeras posiciones.

También debemos observar que la noción de cluster es ambigua y depende de la naturaleza de los datos y de los resultados deseados, es decir cuantos grupos de salida tenemos.

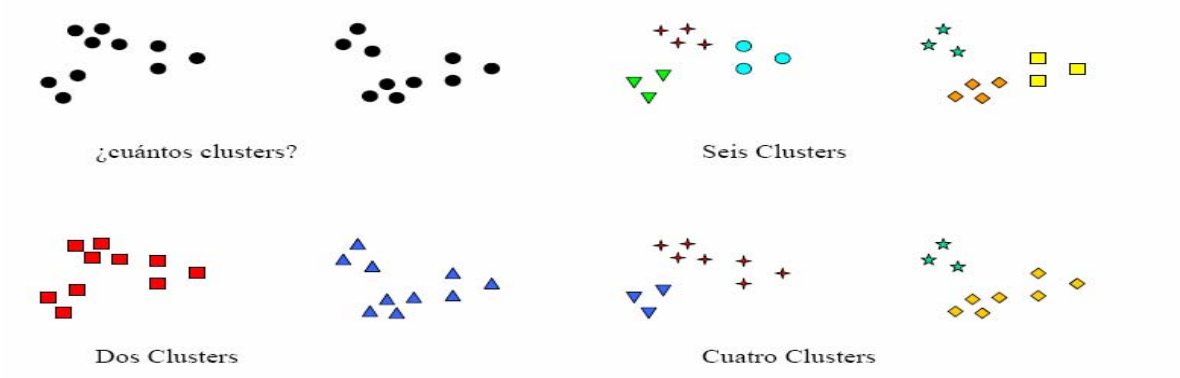

Ilustración 4.1 Concepto de cluster

El número de grupos se escoge inicialmente por el usuario, por tanto el algoritmo agrupará los puntos con respecto a los centroides escogidos, y se harán tantos grupos como haya definido el usuario.

La desventaja de este modelo es que el número de elementos que pertenecen a un grupo puede variar con respecto a una ejecución del algoritmo, y esto es debido a la inicial selección aleatoria de los centroides.

4.1.4 Resultados

Después de la ejecución de este algoritmo obtenemos una matriz de similaridad también se puede llamar matriz de coocurrencias que nos indica para cada repetición del algoritmo cuántas ocasiones un documento pertenece al mismo grupo que los demás documentos, por lo que un número elevado de coocurrencias entre dos documentos indica que tienen mucho en común.

Para pasar de una matriz de concurrencias entre los documentos a una matriz de distancias usamos la siguiente ecuación:

$$Dist(i, j) = 1 - coocurrencia(i, j) / \max(coocurrencia)$$

Se recorre cada una de las posiciones de la matriz de coocurrencia y se dividen sus valores entre el valor máximo de la matriz de concurrencia para normalizar sus componentes, y se resta de 1 para convertir los valores más repetidos en distancias más próximas, con este método se obtienen las distancias relativas que son compatibles con la entrada del siguiente algoritmo.

En la matriz de distancias la diagonal principal esta compuesta de ceros, porque corresponde a observar la relación entre un documento consigo mismo, con lo que la distancia que los separa será mínima, y esto se representa con un cero.

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \ldots & \ldots & 0 \end{bmatrix}$$

Ilustración 4.2 Matriz de distancias

Para trabajar más fácilmente con los datos de la matriz de distancias y debido a la simetría que existe a partir de la diagonal principal, convertiremos la matriz de distancias en un vector de distancias, para ello se suprimirá todo lo que queda debajo de la diagonal principal incluyendo la diagonal de ceros.

4.1.5 Pseudocódigo K-means

```
Inicializar m_i,  i = 1,…,k, en concreto, con algún dato x^t siendo t
aleatorio.
Repetir
For x^t  en X
b_i^t ← 1 if || x^t - m_i || = min_j || x^t - m_j ||
      b_i^t ← 0 sino
For m_i,  i = 1,…,k
      m_i ← sumar sobre t (b_i^t x^t) / sum over t (b_i^t )
Hasta que m_i converge
```

También se ha implementado la variante del algoritmo kmeans denominada bisecting kmeans cuyo pseudocódigo es el siguiente.

```
1: Inicializar en una lista de grupos un primer grupo que contenga
todos los puntos.
2: repetir
3: for i=1 hasta numero grupos
4: Sacar grupo i de lista de grupos
5: Particionar el grupo seleccionado utilizando K-means.
6: Añadir estos 2 grupos a la lista de grupos.
7: end
8: Hasta que en la lista de grupos haya K grupos.
```

4.2 Clustering jerárquico

4.2.1 Introducción

El clustering jerárquico consiste en la construcción de un árbol a través de un proceso de agregación de clusters. En el análisis de un conjunto de "n" muestras se asume como condición inicial la presencia de "n" clusters conteniendo cada uno de ellos una única muestra. Cada paso permite agregar los dos cluster más cercanos en un solo cluster. Repitiendo "n" veces esta operación se consigue obtener un único gran cluster formado por "n" muestras.

Posteriormente hay que analizar el árbol y elegir un nivel de agregación satisfactorio. Este resultado puede obtenerse generalmente de dos formas, seleccionando un número mínimo de cluster o eligiendo una distancia máxima aceptable. Se observa que por el mismo principio de agregación empleado, cada nuevo paso asocia dos cluster más distantes de cuanto lo hubieran sido los dos agregados inmediatamente antes.

4.2.2 Etapas

El agrupamiento jerárquico ascendente, comienza separando cada objeto en un cluster por sí mismo. En cada etapa del análisis, el criterio por el que los objetos son separados se relaja en orden a enlazar los dos conglomerados más similares hasta que todos los objetos sean agrupados en un árbol de clasificación completo.

El criterio básico para cualquier agrupación es la distancia. Los objetos que estén cerca uno del otro pertenecerían al mismo conglomerado o cluster, y los objetos que estén lejos uno del otro pertenecerán a distintos clusters. Para un conjunto de datos dado, los clusters que se construyen dependen de nuestra propia especificación de los siguientes parámetros:

- Cuando se calcula la distancia entre dos clusters, se puede usar el par de objetos más cercanos entre clusters o el par de objeto más alejados, o un compromiso entre estos métodos.

- La medida define la formula para el cálculo de la distancia. Por ejemplo, la medida de distancia euclídea calcula la distancia como una línea recta entre dos clusters. Las medidas de intervalo asumen que las variables están medidas en

escala, las medidas de conteo asumen que son números discretos, y las medidas binarias asumen que toman dos valores.

La estandarización permite igualar el efecto de las variables medidas sobre diferentes escalas.

El cálculo de la proximidad entre dos grupos es lo que diferencia entre sí a las técnicas de agrupamiento jerárquico. Existen diversas técnicas de agrupamiento aglomerativas como pueden ser *MIN*, *MAX* y *Group Average*. *MIN* define la proximidad del grupo como la proximidad entre los dos puntos más cercanos que están en diferentes grupos. *MAX* entiende la proximidad del grupo como la proximidad entre los puntos más lejanos pertenecientes a diferentes grupos. Si nuestras proximidades son distancias entonces los nombres *MIN*, y *MAX* se corresponden con el concepto de distancia intuitivo al que representan, es decir la distancia menor y la mayor entre los dos puntos que se encuentran en diferentes grupos, en cambio para similaridad entre grupos los conceptos están cambiados, porque existe mayor similaridad entre 2 grupos cuanto más próxima sea la distancia que los separa. Para que los nombres de las técnicas no confundan el significado del concepto al que se refieren, se utiliza el nombre de *single link* en el caso descrito para *MIN* y *complete link* para el caso *MAX*.

El tercer método con el que se puede medir la proximidad entre los grupos es *group average*, que define la proximidad entre los distintos grupos como la proximidad media entre pares de todos los pares de puntos de distintos grupos.

- Single Linkage $x_i \in S_i, x_j \in S_j; \min(d(x_i, x_j))$
- Complete Linkage $x_i \in S_i, x_j \in S_j; \max(d(x_i, x_j))$
- Average Linkage $\frac{1}{|S_i||S_j|} \sum_{x_i \in S_i} \sum_{x_j \in S_j} d(x_i, x_j)$

La siguiente ilustración muestra claramente la manera de medir la distancia para cada uno de los casos comentados anteriormente.

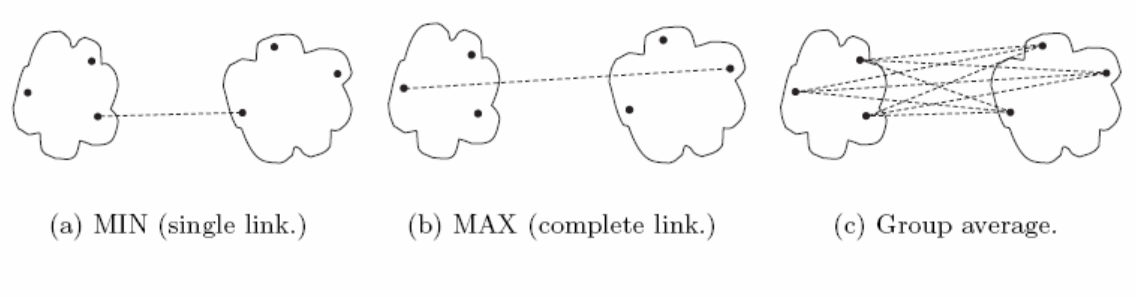

Ilustración 4.3 Diferentes métodos de medir distancias

4.2.3 Dendograma

Un dendograma es un tipo de representación gráfica o diagrama de datos en forma de árbol que organiza los datos en subcategorías que se van dividiendo en otros hasta llegar al nivel de detalle deseado, asemejándose a las ramas de un árbol que se van dividiendo en otras sucesivamente.

Este tipo de representación permite apreciar claramente las relaciones de agrupación entre los datos e incluso entre grupos de ellos aunque no las relaciones de similaridad o cercanía entre categorías. Observando las sucesivas subdivisiones podemos hacernos una idea sobre los criterios de agrupación de los mismos, la distancia entre los datos según las relaciones establecidas.

El dendograma es la representación gráfica que mejor ayuda a interpretar el resultado de un análisis cluster, se utilizó durante el desarrollo del proyecto para observar los resultados de los grupos.

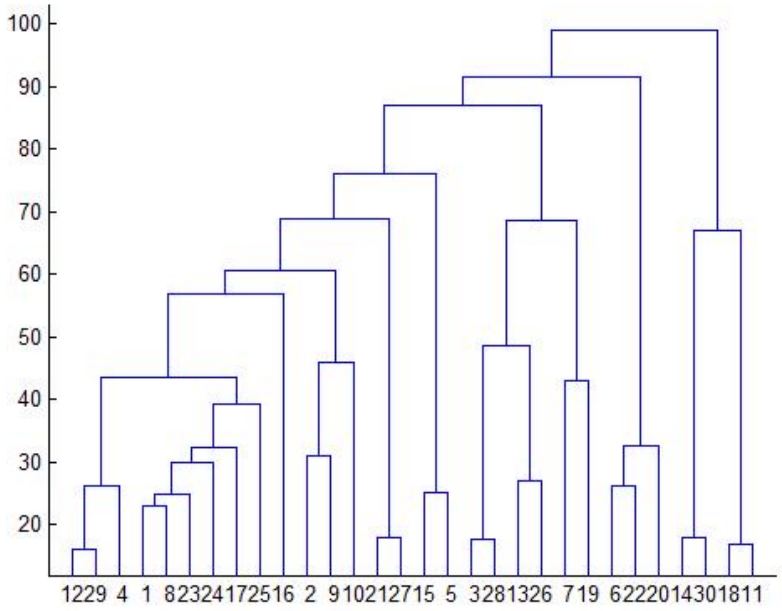

Ilustración 4.4 Representación dendograma

En la siguiente gráfica se puede ver el número de grupos que existe en cada momento si se recorre el árbol verticalmente y nos detenemos a una determinada distancia, contando el número de líneas verticales que hay a esa altura. En nuestro caso se ha optado por calcular las distancias entre grupos como *group average*, ya que de esta manera los puntos que se utilizan como representantes para medir la distancia no son ni el más cercano ni el más lejano, los cuales pueden no ir en consonancia con el resto de los puntos de un grupo concreto, y se consigue menos sensibilidad al ruido y a los *outliers.*

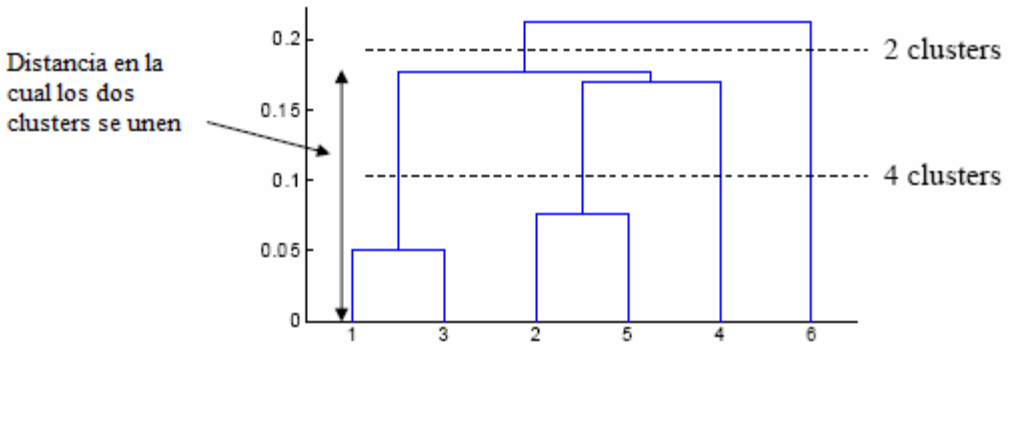

Ilustración 4.5 Relación entre distancia y grupos

4.2.4 Limitaciones

El dendrograma correspondiente a un conglomerado jerárquico no es único, puesto que por cada línea que une los clusters uno necesita especificar que sub-árbol va a la derecha y cuál a la izquierda.

Cada uno de los diferentes métodos de unión que hemos visto tiene sus ventajas y desventajas de manera independiente, y nunca vamos a saber cuál es la naturaleza de los datos con antelación, ya que en cada caso será diferente.

4.2.5 Pseudocódigo

Datos de entrada son S, una lista de elementos.

Datos de salida T, que es un árbol.

1 Se colocan los datos de S en grupos de manera independiente, creando así una lista de clusters *L*.

$L = S_1, S_2, S_3, \ldots, S_{n-1}, S_n$.

2 Se calcula la función de distancias entre cada par de elementos de L para encontrar los grupos más cercanos entre sí $\{S_i, S_j\}$.

$$\frac{1}{|S_i||S_j|}\sum_{x_i \in S_i}\sum_{x_j \in S_j} d(x_i, x_j)$$

3 Se quitan los elementos S_i y S_j que ya han sido asignados a un grupo de la lista de elementos *L*.

4 Los elementos más cercanos S_i y S_j se unen creando un nodo S_{ij} en el árbol *T*.

5 Volver al paso 2 hasta que sólamente quede un cluster.

5 ESCALAMIENTO MULTIDIMENSIONAL

5.1 Introducción

Las técnicas de MDS (*Multi Dimensional Scaling*) tratan sobre el siguiente problema: para un conjunto de similitudes o distancias observadas entre un par de objetos de un total de N, se trata de encontrar una representación gráfica de estos en pocas dimensiones, de modo que sus posiciones casi ajusten las similitudes o distancias originales.

Con N objetos, se buscan configuraciones de q < (N − 1) dimensiones, de modo que el ajuste entre las posiciones originales y las posiciones en las q dimensiones sea el más preciso posible, esto se mide mediante el concepto del stress.

Si se usan las magnitudes originales de las distancias o similitudes, se tiene el llamado escalamiento multidimensional métrico. Si se usan rangos (orden de las observaciones), en vez de distancias, se tiene el MDS no métrico.

5.2 Escalamiento multidimensional no métrico

Dados N objetos, existen $M = \frac{N(N-1)}{2}$ distancias o similitudes entre pares de diferentes objetos. Alternativamente, se pueden usar rangos ordenados. Las similitudes se pueden ordenar en orden creciente como:

$$s_{i_1 k_1} < s_{i_2 k_2} < \ldots < s_{i_m k_m}$$

Aquí $s_{i_1 k_1}$ es la menor de las M similitudes, donde i1, k1 es el par de observaciones que son menos similares y, del mismo modo, im, km, las más similares. Buscamos una configuración de dimensión q tal que las distancias entre los N objetos mantengan el orden expresado en la relación anterior. Es decir, tiene que cumplirse:

$$d^{(q)}_{i_1 \; k_1} > d^{(q)}_{i_2 \; k_2} > \ldots > d^{(q)}_{i_m \; k_m}$$

Lo importante es que se mantenga el orden, no las magnitudes en sí.

Para un número dado de dimensiones (q), puede que no se encuentre una configuración como la anterior que conserve las similitudes anteriores.

Kruskal dio una medida de la adecuación de la representación en q dimensiones a las similitudes originales basándose en las distancias d_{ij}; dicha medida se denomina stress y se calcula con la siguiente fórmula:

$$S = \sqrt{\frac{\sum (d_{ij} - \overset{*}{d}_{ij})^2}{\sum_{ij} d_{ij}^2}}$$

Se buscan representaciones geométricas en q dimensiones de modo que el stress sea mínimo. Empíricamente, se considera que si el stress es alrededor de 0,2, la bondad del ajuste es pobre; si es del 0,05, la bondad del ajuste es buena y a partir de 0,025 es excelente.

La idea es minimizar el stress para un número fijo q de dimensiones mediante un proceso iterativo con lo que se consigue que las coordenadas (x1, x2, ..., xt) de cada objeto se cambian ligeramente de tal manera que la medida de ajuste se reduzca.

El proceso iterativo consiste en calcular las distancias euclidianas entre los objetos de esa configuración, esto es, calcular las dij, que son las distancias entre el objeto i y el objeto j, posteriormente hay que realizar una regresión de dij sobre δij . Esta regresión puede ser lineal, polinomial o monótona. Por ejemplo, si se considera lineal se tiene el modelo:

$$d_{ij} = a + b\delta_{ij} + \varepsilon$$

y utilizando el método de los mínimos cuadrados se obtienen estimaciones de los coeficientes a y b, y de ahí puede obtenerse lo que genéricamente se conoce como una disparidad:

$$\hat{d}_{ij} = \hat{a} + \hat{b}\delta_{ij}$$

con estos datos se vuelve a calcular el stress hasta que la medida de ajuste entre las disparidades y las distancias de configuración no puedan seguir reduciéndose.

El resultado final del análisis son las coordenadas de los n objetos en las p dimensiones, se utilizarán estas coordenadas para elaborar un gráfico que muestre cómo están relacionados los objetos.

5.3 Escalamiento multidimensional métrico

En la técnica de escalamiento multidimensional métrico el conjunto de datos son relaciones percibidas entre los elementos de un conjunto de objetos o estímulos. Existe un isomorfismo entre los objetos y sus medidas de proximidad y entre el conjunto de puntos del espacio euclídeo y sus medidas de distancia por otro.

Dado un conjunto de n objetos se dispone de una matriz simétrica de distancias que contiene datos de la similaridad entre los objetos, el escalamiento multidimensional trabaja sobre este tipo de matrices para conseguir obtener distancias absolutas.

A partir de los n(n-1)/2 datos de la matriz de distancia, que son aquellos datos que quedan por encima de la diagonal principal como se puede ver en la siguiente imagen, se obtendrá una representación gráfica en un espacio de dimensión dos para obtener las relaciones que existirán entre los documentos debido a su proximidad.

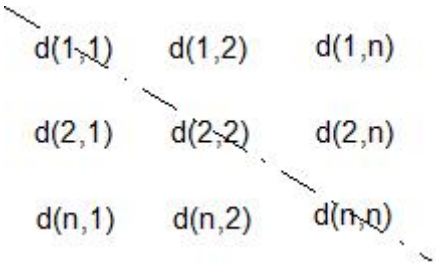

Ilustración 5.1 Matriz de distancias

A partir de la matriz de distancias D, se construye una matriz $A = (a_{ij})$ de coeficientes de asociación entre objetos, con sus elementos definidos como:

$$a_{ii} = -(\frac{1}{2})d_{ii}^{\ 2} = 0$$

$$a_{ij} = -(\frac{1}{2})d_{ij}^{\ 2}$$

La matriz B contiene una representación de los n objetos de un espacio euclídeo de dimensión h $(k \le h \le n)$, definida de la siguiente manera:

$$b_{ij} = a_{ij} - a_{i}. - a._{j} + a..$$

$$a_{i}. = \frac{1}{n}\sum_{j=1}^{n} a_{ij}$$

$$a._{j} = \frac{1}{n}\sum_{i=1}^{n} a_{ij}$$

$$a.. = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}$$

Con esta matriz se calculan sus autovalores, $\lambda_r (r = 1,2,...,h)$, cuyos autovectores asociados son las columnas de una matriz X cuya fila i-ésima contiene el punto P_i de coordenadas $x_{ij} (j = 1,...,h)$, y la r-ésima columna de X es el autovector correspondiente a λ_r.

Se define $\Gamma = X\Delta^{-1/2}$ como la matriz ortonormal que contiene los autovectores normalizados de B, sumando los cuadrados de un autovector x_j equivale al autovalor correspondiente λ_j, siendo $\Delta = diag(\lambda_1,...,\lambda_n)$. La matriz Γ contiene una relación entre los puntos basándose en la proximidad o lejanía de estos, tomaremos puntos que tengan dos dimensiones (x,y), para poder generar la gráfica buscada como se puede ver en el siguiente ejemplo.

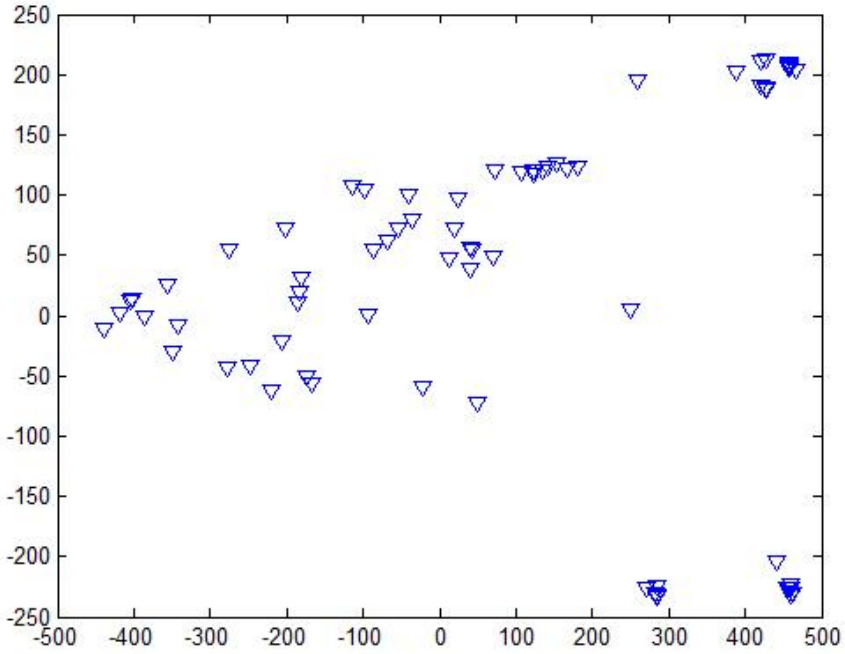

Ilustración 5.2 Representación 2D tras aplicar MDS

En la ilustración 6.1 se observa un gráfico sobre el que se han proyectado puntos con dimensionalidad inicial alta a una dimensión que es posible ver, esto es 2 dimensiones, pero guardando la relación entre las distancias de los puntos, hay que tener en cuenta que cuando se proyecta sobre dimensiones menores, puntos que originalmente estaban muy separados, pueden aparecer muy juntos en 2D, con lo cual este resultado esta sujeto a las restricciones de este concepto que debe asumirse.

5.4 Aplicaciones

Se consideran las distancias en relación a vuelos entre 10 ciudades norteamericanas:

	Atlanta	Chicago	Denver	Houston	L.Angeles	Miami	N.York	S.Francisco	Seattle	Washington
Atlanta	0.00	587.00	1212.00	701.00	1936.00	604.00	748.00	2139.00	218.00	543.00
Chicago	587.00	0.00	920.00	940.00	1745.00	1188.00	713.00	1858.00	1737.00	597.00
Denver	1212.00	920.00	0.00	879.00	831.00	1726.00	1631.00	949.00	1021.00	1494.00
Houston	701.00	940.00	879.00	0.00	1374.00	968.00	1420.00	1645.00	1891.00	1220.00
L.Angeles	1936.00	1745.00	831.00	1374.00	0.00	2339.00	2451.00	347.00	959.00	2300.00
Miami	604.00	1188.00	1726.00	968.00	2339.00	0.00	1092.00	2594.00	2734.00	923.00
N.York	748.00	713.00	1631.00	1420.00	2451.00	1092.00	0.00	2571.00	2408.00	205.00
S.Franciso	2139.00	1858.00	949.00	1645.00	347.00	2594.00	2571.00	0.00	678.00	2442.00
Seattle	218.00	1737.00	1021.00	1891.00	959.00	2734.00	2408.00	678.00	0.00	2329.00
Washington	543.00	597.00	1494.00	1220.00	2300.00	923.00	205.00	2442.00	2329.00	0.00

Tabla 5.1 Distancias entre ciudades de EEUU

Se desea representar en 2 dimensiones un mapa de las ciudades más relevantes de los Estados de EEUU conociendo una matriz que contiene las distancias entre una ciudad y todas las demás, teniendo en la diagonal principal una matriz de ceros porque sería el resultado de medir la distancia entre una ciudad y ella misma.

A continuación se muestra el mapa de EEUU con la localización de las ciudades que hemos escogido en la matriz de distancias para ver los resultados del algoritmo sobre un caso real.

Ilustración 5.3 Mapa de EEUU

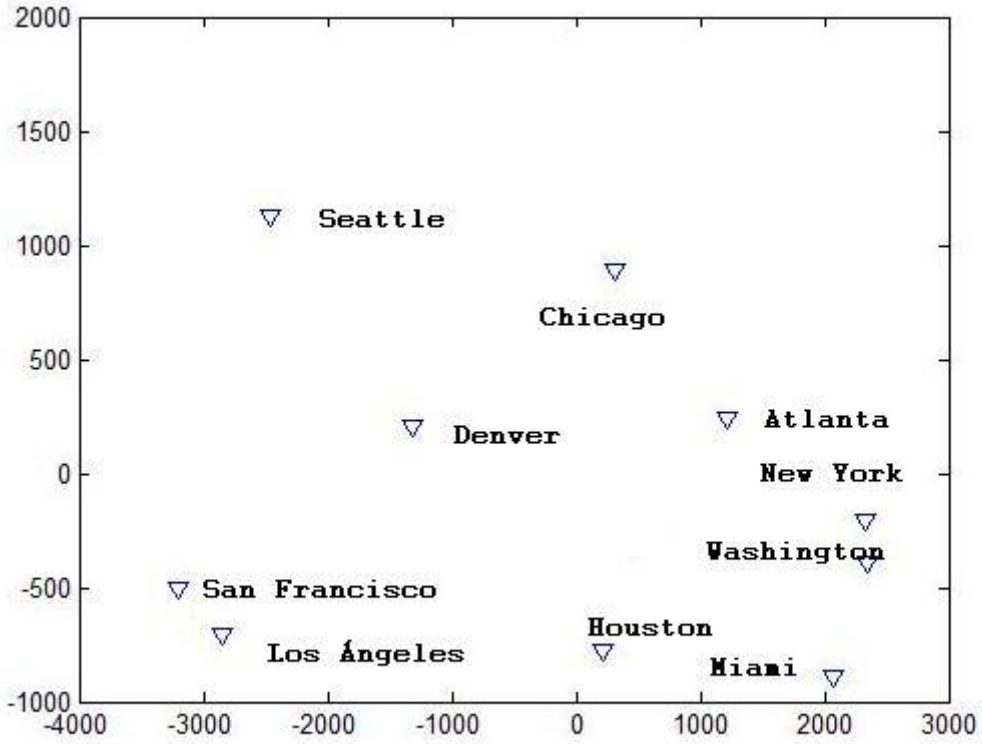

Ilustración 5.4 Resultado de posicionamiento de ciudades

Se observa que estos puntos corresponden a la localización de las ciudades en el mapa, y guardan la proporción entre las distancias, para la elaboración de la demostración del funcionamiento del algoritmo se ha utilizado la función classical multidimensional scaling de Matlab, con la matriz de datos de las distancias en la entrada.

5.5 Implementación

Se compararon los dos métodos expuestos utilizando el programa Matlab, y debido a un mejor ajuste de los resultados a la realidad, se ha implementado el escalamiento multidimensional métrico. Para poder calcular los autovalores y autovectores se ha utilizado la librería Jama en Java, el resto del código se ha realizado a partir del siguiente pseudocódigo que se basa en la teoría expuesta.

5.6 Pseudocódigo

1) Dada una matriz de similaridad D, obtener la matriz de distancias absolutas B.

$$B = -\frac{1}{2} HD^2 H$$

Siendo H la matriz de centrado:

$$H = I - R$$

$$R = \sum_{i=1}^{n} \frac{1}{n}$$

2) Diagonalizar B.

$$B = U \wedge U'$$

$\wedge$ es la matriz diagonal que contiene los autovalores de B.

U es una matriz n x p ortogonal cuyas columnas son los autovectores de B.

3) Las filas de $X = U \wedge^{1/2}$ son las coordenadas principales euclídeas de los elementos del conjunto sobre el que se calcularon las distancias.

6 DESARROLLO METODOLOGÍA Y RESULTADOS

6.1 Adquisición de datos

El primer paso en la adquisición de datos consiste en extraer la información que contiene el conjunto de los documentos que se van a agrupar, esto se hace con la finalidad de cambiar el formato del archivo original en un formato que sea más rápido y fácil de tratar por el ordenador.

El programa es capaz de trabajar principalmente con el tipo de archivo pdf que fue la idea original del proyecto, pero se amplió para poder leer archivos en texto plano y documentos xml.

6.1.1 Formato PDF

PDF (del inglés *Portable Document Format*, Formato de Documento Portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa *Adobe Systems*. Está especialmente ideado para documentos susceptibles de ser impresos, ya que especifica toda la información necesaria para la presentación final del documento, determinando todos los detalles de cómo va a quedar, no requiriéndose procesos anteriores de ajuste ni de maquetación. Cada vez se utiliza más como una especificación de visualización, gracias a la gran calidad de las fuentes utilizadas y a las facilidades que ofrece para el manejo del documento, como búsquedas, hiperenlaces, etc.

6.1.2 Características de PDF

- Es multiplataforma, lo que significa que puede ser visualizado por los principales sistemas operativos (Windows, Unix/Linux o Mac), sin que se modifiquen ni el aspecto ni la estructura del documento original.

- Puede integrar cualquier combinación de texto, gráficos, imágenes e incluso música.

- Es uno de los formatos más extendidos en Internet para el intercambio de documentos, por ello es muy utilizado por empresas, gobiernos e instituciones educativas.

- Es una especificación abierta, para la que se han generado herramientas de Software Libre que permiten crear, visualizar o modificar documentos en formato PDF, un ejemplo es la suite ofimática *OpenOffice.org*.

- Puede cifrarse para proteger su contenido e incluso firmarlo digitalmente.

- El archivo PDF puede crearse desde varias aplicaciones exportando el archivo, como es el caso de los programas de *OpenOffice.org* y también en el nuevo paquete ofimática de *Microsoft Office 2007*.

- Es el estándar ISO (ISO 19005-1:2005) para ficheros contendores de documentos electrónicos con vistas a su preservación de larga duración.

6.1.3 Incorporación al proyecto

Para extraer texto del formato pdf se utiliza una librería experta en el manejo de este tipo de datos llamada *pdfbox,* porque ofrece múltiples posibilidades en el tratamiento de este formato, como por ejemplo extracción de texto, unión de varios documentos en uno solo, encriptación y desencriptación de documentos, creación de documentos desde un formato de texto plano al formato pdf, crear documentos pdf a partir de imágenes e impresión de documentos. De todas estas funcionalidades, sólo se ha utilizado la función de extracción de texto desde formato pdf a texto plano, que será más fácil de tratar.

Para conseguir un mejor funcionamiento, hubo que instalar una librería denominada *FontBox* que contiene diversos tipos de fuentes, para hacer compatible la librería pdfbox con todos los tipos de formato de texto conocidos.

6.1.4 Formato TXT

Se proporciona soporte a documentos en formato txt que son conocidos también como archivos de texto plano, o texto simple, por carecer de información destinada a generar formatos (negritas, subrayado, cursivas, tamaño, etc.) y tipos de letra (por ejemplo, *Arial*, *Times*, *Courier*, etc.). El término texto plano proviene de una traducción literal del término inglés *plain text*, término que en lengua castellana significa texto simple o texto sencillo.

No están extendidos para la divulgación de la información por la simplicidad de sus posibilidades, que no abarcan la incorporación de imágenes, estructuración de la información, por eso han sido ampliamente desplazados por los nuevos formatos.

6.1.5 Formato XML

XML, sigla en inglés de *Extensible Markup Language* («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos, de la misma manera que HTML es a su vez un lenguaje definido por SGML. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

El lenguaje XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

6.1.6 Ventajas XML

Es extensible, lo que quiere decir que una vez diseñado un lenguaje y puesto en producción, igual es posible extenderlo con la adición de nuevas etiquetas de manera de que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato.

El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles. De esta manera se evitan bugs y se acelera el desarrollo de la aplicación.

Si una tercera persona decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.

6.1.7 Estructura del documento XML

La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes, entonces se tiene un árbol de pedazos de información. Un ejemplos puede ser el caso de un tema musical, que se compone de compases, y estos están formados a su vez por notas, estas partes se llaman elementos, y se las señala mediante etiquetas.

Una etiqueta consiste en una marca hecha en el documento, que señala una porción de éste como un elemento con un sentido claro y definido, tienen la forma **<nombre>**, donde *nombre* es el nombre del elemento que se está señalando.

A continuación se muestra un ejemplo para entender la estructura de un documento XML:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE Edit_Mensaje SYSTEM "Lista_datos_mensaje.dtd"
                              [<!ELEMENT Edit_Mensaje (Mensaje)*>]>
 <Edit_Mensaje>

     <Mensaje>

          <Remitente>
               <Nombre>Nombre del remitente</Nombre>
               <Mail> Correo del remitente </Mail>
          </Remitente>

          <Destinatario>
               <Nombre>Nombre del destinatario</Nombre>
               <Mail>Correo del destinatario</Mail>
          </Destinatario>

          <Texto>

               <Asunto>
                 Este es mi documento con una estructura muy sencilla
                    no contiene atributos ni entidades....
               </Asunto>

               <Parrafo>
                 Este es mi documento con una estructura muy sencilla
                    no contiene atributos ni entidades....
               </Parrafo>

          </Texto>
     </Mensaje>

</Edit_Mensaje>
```

Aquí está el ejemplo de código del DTD del documento:

```
<!ELEMENT Mensaje (Remitente, Destinatario, Asunto, Texto)*>
          <!ELEMENT Remitente (Nombre, Mail)>
                    <!ELEMENT Nombre (#PCDATA)>
                    <!ELEMENT Mail   (#PCDATA)>

          <!ELEMENT Destinatario (Nombre, Mail)>
                    <!ELEMENT Nombre (#PCDATA)>
                    <!ELEMENT Mail   (#PCDATA)>

          <!ELEMENT Asunto (#PCDATA)>
          <!ELEMENT Texto  (Parrafo)>
                    <!ELEMENT Parrafo (#PCDATA)>
```

6.1.8 Document type definition (DTD)

La DTD define los tipos de elementos, atributos y entidades permitidas, y puede expresar algunas limitaciones para combinarlos, los documentos XML que se ajustan a su DTD se denominan válidos.

Los elementos deben ajustarse a un tipo de documento declarado en una DTD para que el documento sea considerado como válido.

Un modelo de contenido es un patrón que establece los subelementos aceptados, y el orden en que se aceptan.

Los atributos se usan para añadir información adicional a los elementos de un documento, podemos encontrar los siguientes tipos de atributos:

- Atributos CDATA y NMTOKEN
- Atributos enumerados y notaciones
- Atributos ID e IDREF

XML hace referencia a objetos que no deben ser analizados sintácticamente según las reglas XML, mediante el uso de entidades. Las entidades pueden ser:

- Internas o externas
- Analizadas o no analizadas
- Generales o parametrizadas

Los espacios de nombres XML permiten separar semánticamente los elementos que forman un documento XML.

6.1.9 Implementación

Se ha utilizado la librería Xerces para poder extraer el texto de los documentos, en concreto la API de DOM, debido a su importancia JRE viene con una implementación por defecto. Hay que destacar que la implementación libre más importante de DOM es Xerces.

El *Document Object Model* (Modelo en Objetos para la representación de Documentos), abreviado DOM, es esencialmente una interfaz a través de la cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML. Su objetivo es ofrecer un modelo orientado a objetos para el tratamiento y manipulación de esos dos tipos de documentos.

6.1.10 Elección de API

SAX es un API orientada a eventos cuya finalidad es obtener fácilmente la información almacenada en un archivo XML, destaca por su pequeño consumo de recursos debido a que prácticamente sólo almacena en memoria la información del nodo que acaba de ser analizado.

Por otra parte DOM es un API diseñada para mantener en memoria toda la información de un documento, y en algunas aplicaciones es un requisito tener toda la información del documento en memoria para poder acceder más rápidamente a los datos, o incluso combinarlos o añadir más información.

Puede darse la circunstancia de que a priori no conozcamos la información que vayamos a necesitar de un documento o que sea necesario recorrer en varias ocasiones el documento. Estos requerimientos se dan en aplicaciones como editores XML/SGML, o procesadores XSLT. DOM también es útil en aplicaciones que requieran crear un

documento XML dinámicamente y la generación del documento no sea lineal, sino que se requiera añadir nodos a partes ya generadas del documento.

Una de las ventajas de DOM o un API similar para obtener información de un archivo XML ya que se pueden recorrer los nodos muy cómodamente, y esto se hace una sola vez para extraer el texto, en nuestro caso no tenemos ninguna limitación de recursos, así que es una de las causas por las que se escogió DOM en vez de SAX. El proyecto también se hubiera podido realizar utilizando SAX, sin embargo, la implementación de las funcionalidades necesarias hubiera sido más engorrosa.

6.2 Tratamiento del texto

6.2.1 Introducción

Las operaciones previas a la clasificación que hay que realizar sobre el texto, consisten sobre todo en filtrar el conjunto de los datos de entrada para agilizar el procesamiento y preparar los datos para que puedan ser clasificados mejor posteriormente. Para ello una primera actuación consiste en pasar a minúsculas todo el texto y quitar los acentos de las palabras que los tengan, esto servirá para que una misma palabra independientemente de las múltiples variaciones ortográficas en las que aparezca en el texto se identifique como la misma palabra.

6.2.2 Stop Words

Posteriormente se realiza un filtrado sobre el conjunto del texto eliminando las palabras que no aportan nada de información, esto es, ningún tipo de distinción para poder asignar el texto a alguna categoría, son las denominadas *stop words*, se trata de los determinantes, pronombres, y conjunciones.

Se ha creado una lista de *stop words* en los idiomas inglés y español para dar mayor compatibilidad idiomática, con la particularidad de poder procesar archivos en idioma inglés o idioma español de manera independientemente, pero no mezclados ya que no hay una relación directa entre las palabras de ambos idiomas.

Para saber el idioma en el que se ha escrito un documento para poder aplicar la lista de *stop words* correspondiente, se ha diseñado una función que selecciona el idioma del documento basándose en el conteo de las propias *stop words* de cada idioma que

contiene un documento, por tanto si el número de *stop words* es mayor en el idioma español, el documento tratará de ese idioma, y si hay más en el idioma inglés el idioma del documento será inglés.

6.2.3 Lematizadores

Una vez que se ha quitado del texto las *stop words*, se buscan los lexemas de cada una de las palabras restantes con la finalidad de quitar todas aquellas palabras que derivan de un mismo lexema. Este tipo de palabras que comparten un mismo lexema serán tratadas como si fuesen la misma palabra, esto es especialmente útil para las palabras que tienen diferente género y número porque aunque se escriban de manera diferentes, comparten el mismo significado.

Para poder hallar los lexemas de cada una de las palabras se ha utilizado un lematizador desarrollado en java llamado *Snowball,* que es ampliamente utilizado en temas de recuperación de la información y que da soporte a varios idiomas entre ellos al inglés y al español. Un ejemplo de su funcionamiento se muestra en la siguiente tabla.

Palabra	**Lexema**
Example	Exampl
Of	Of
Stemming	Stem
Algorithm	Algorithm

Tabla 6.1 Conversión de palabras a lexemas

Una consecuencia directa del uso del lematizador consiste en que nos permite filtrar todavía más el texto, porque se suprimen todas aquellas palabras que se muestran con variantes pero que en el fondo significan lo mismo, esto afecta a los tipos de palabras siguientes: nombres, adjetivos, adverbios y verbos, pero no a las conjunciones y preposiciones, porque se suprimieron en el proceso de filtrado de *stop words*.

Lista de palabras	Lista única de palabras
Buffer	Buffer
Buffered	
Bug	Bug
Bugs	
Build	Build
Builders	

Tabla 6.2 Conversión a lista de palabras única

Se observa como las palabras que tienen un mismo lexema son consideradas como la misma palabra, es importante este paso porque de otra forma sería más difícil encontrar relaciones entre los documentos, ya que se considerarían como diferentes palabras todas aquellas que variasen en una sola letra, y en el caso de los verbos por ejemplo sería muy difícil identificar un mismo verbo debido a la cantidad de variantes que pueden tener debido a los diferentes tiempos verbales.

6.2.4 Lista de palabras única

Para identificar al conjunto de los documentos, es necesario crear una lista de palabras ordenadas alfabéticamente que se forma a partir de las palabras de todos los documentos, con los requisitos de que una misma palabra no debe aparecer repetida dos o más veces, ni se debe contener *stop words*, ni dos palabras que provengan un mismo lexema.

La manera de quitar las palabras repetidas consiste en ordenar una lista de palabras alfabéticamente y las palabras que sean iguales a la palabra anterior en la lista serán consideradas palabras repetidas, y por tanto serán eliminadas de la lista de palabras.

Para confeccionar esta lista de palabras única, el método que se utiliza consiste en generar por cada documento una lista de palabras única parcial, es decir, en la que no haya ni palabras repetidas ni se encuentren dos palabras con el mismo lexema ni queden *stop words*, posteriormente se ordena de manera alfabética esta lista de palabras.

Cuando se ha terminado de hacer esto con cada uno de los documentos se confecciona una lista única de palabras global para todos los documentos utilizando las listas de palabras parciales anteriormente generadas en cada documento. Al procesar de manera

individual cada uno de los documentos, es más rápido confeccionar la lista de palabras única global, porque en este proceso se ha realizado un filtrado de muchas palabras que introducen una cantidad de procesamiento extra.

6.2.5 Representación vectorial de un documento

Al final del apartado anterior hemos obtenido una base de un espacio vectorial en el que representar cada uno de los documentos. Bastaría con anotar cuántas veces aparece cada una de las palabras en un documento dado formando así un vector cuya longitud sería igual que la longitud de la lista de palabras. Este concepto recibe el nombre de *term frequency* (tf).

Sin embargo, no todas las palabras tienen la misma importancia para distinguir entre documentos puesto que hay palabras que se encuentran en todos los documentos (como puede ser el verbo "tener") y que por lo tanto no nos servirán para distinguir unos documentos de otros.

Es por ello que la representación vectorial anterior es modificada para cada documento de forma que pierdan peso en el vector aquellas palabras que no sirven para distinguir entre documentos. Para esto hay que aplicar la siguiente fórmula:

$$tf' = tf * \log\left(\frac{N}{df}\right)$$

donde N que es el número total de documentos que hay en el corpus, y df (*document frequency*) el número de documentos del conjunto del corpus en el que aparece esa palabra. De esta forma vemos que si una palabra aparece en todos los documentos (como "tener"), su valor en el vector después de esta transformación será nulo.

El conteo de las palabras se realiza utilizando como referencia la lista de palabras única que se ha generado anteriormente. Se busca en una lista de palabras ordenadas alfabéticamente para cada uno de los documentos el número de veces aparece cada una esas palabras repetidas en el texto.

Al término de este procesado del texto obtenemos una matriz que posee para cada una de las palabras de la lista única un número asociado que es su peso con respecto a los demás documentos, que será utilizada posteriormente para el análisis y la clasificación.

Esta matriz que denominaremos matriz de palabras-documentos, y denotaremos por la letra V, posee una dimensionalidad de los datos muy grande que esta en función del conjunto de los documentos. En nuestro caso esta matriz tiene más de 5 millones de números con precisión *double*. Se muestra en la siguiente ilustración un ejemplo del contenido de esta matriz.

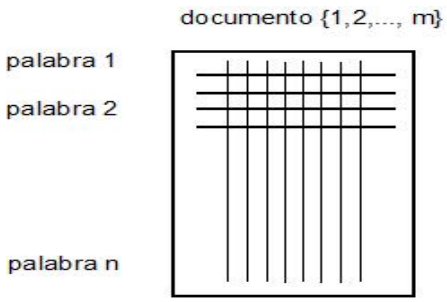

Ilustración 6.1 Matriz V

6.2.6 Filtrado de la matriz de palabras-documentos

La matriz V todavía contiene demasiada información para conseguir realizar los grupos de manera óptima, y será necesario filtrarla para conseguir mejorar los resultados. Para ello realizamos los siguientes tipos de filtrado que se describen a continuación. La Ilustración 7.2 muestra los vectores correspondientes a tres documentos tal cual se obtienen del proceso anterior.

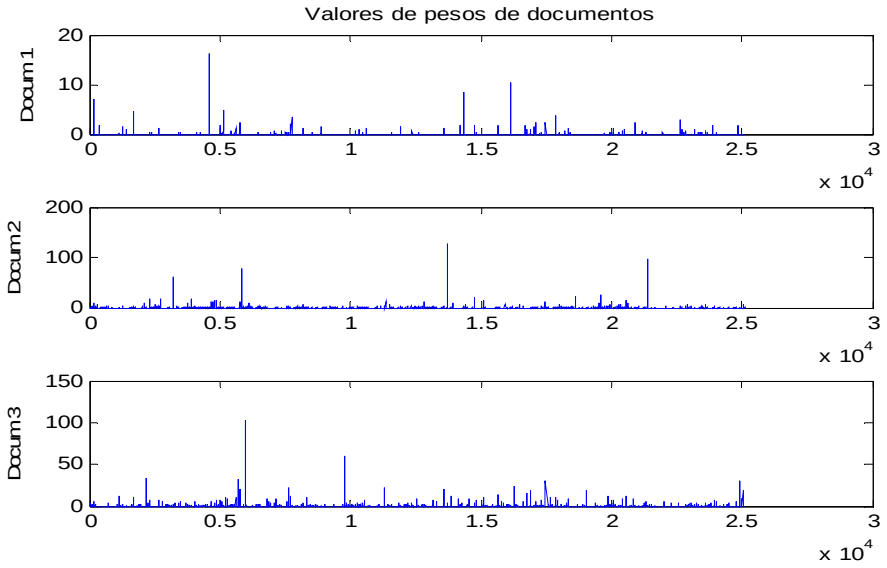

Ilustración 6.2 Matriz V sin filtrar

6.2.7 Filtrado por longitud de palabra

Dentro del listado de todas las palabras, debemos suprimir aquellas que tienen menos de 2 letras, ya que no suelen resultar ni discriminantes, ni aportan significados léxicos útiles, por tanto se realiza un filtrado sobre esta matriz que contiene las relación de pesos en cada palabra de cada documento, quitando una fila completa de datos cada vez que encontramos una palabra corta, de esta manera se quita una palabra con sus correspondientes valores en todos los documentos.

6.2.8 Filtrado de palabras infrecuentes

Debido al cálculo que se realiza en el apartado 7.2.5, existen casos en los que no obtenemos información discriminante para hacer los clusters o grupos, esto ocurre cuando existe una palabra que no es habitual, que aparece en muy pocos documentos.

La relevancia de una palabra se observa mirando la magnitud de su peso y en este caso su valor es muy grande y no se corresponde con su relevancia real, esto se debe a que en la fórmula del peso, al estar dividiendo df o frecuencia en documentos, que es precisamente el número de documentos en el que aparece esa palabra, su valor en la fórmula se dispara provocando resultados que se escapan mucho de los valores habituales de los pesos, por tanto se van a filtrar todas aquellas palabras que no aparezcan en un mínimo número de documentos, como por ejemplo de 10 a 15. A continuación se muestra una gráfica en la que se indica el número de palabras que aparece en un número mínimo de documentos.

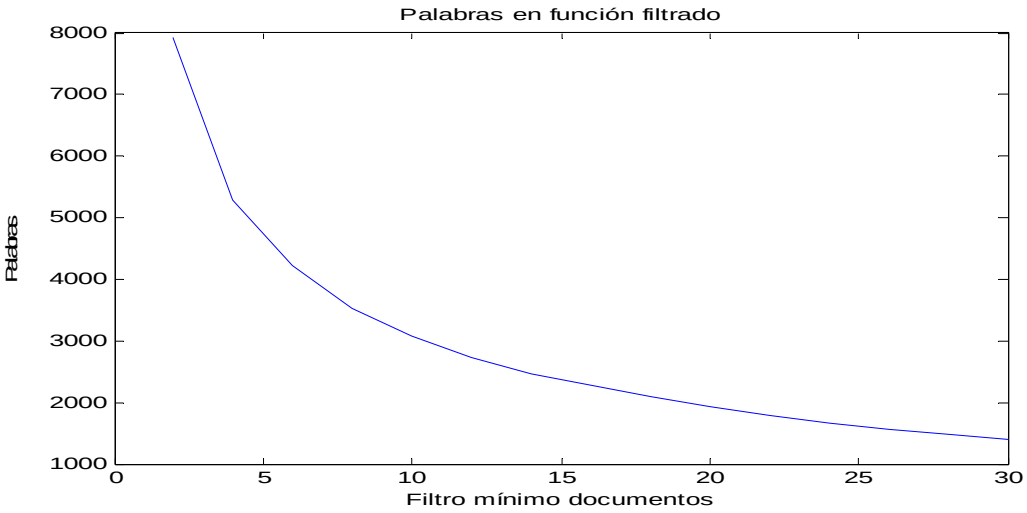

Ilustración 6.3 Cantidad de palabras en función del filtrado

Realizando un filtrado consistente en quitar todas aquellas palabras que hemos denominado como infrecuentes, ya que no se repiten en muchos documentos en la matriz que contiene la relación entre las palabras, los documentos y los pesos calculados, que hemos llamado matriz V, se observa como el número de palabras restantes se hace cada vez más pequeño cuando aumentamos el requisito de mínimo número de documentos en el que debe de aparecer una palabra para no ser filtrada.

Este filtrado tiene la ventaja añadida de que además de quitar las palabras que no son discriminantes entre los documentos, se suprimen solamente las palabras que tienen un peso muy elevado descritas anteriormente.

El número de palabras que quedan en la matriz V después del filtrado, es mucho menor en el caso de incrementar la condición del mínimo número de documentos, esto tiene sentido porque antes de realizar el filtrado existen menos palabras que aparecen en muchos documentos, y muchas palabras que aparecen en pocos documentos y que no resultarán discriminantes, que son precisamente las que vamos a quitar.

Elevar el número mínimo de documentos en el que debe de aparecer una palabra para ser tenido en cuenta, tiene la ventaja de disminuir el tiempo de procesado, porque disminuye el tamaño de la matriz V, ya que quitamos para cada una de las palabras una fila entera de esa matriz de datos, porque recordemos que una palabra tiene relación con todo el conjunto de los documentos. No se debe filtrar excesivamente en este paso superando el valor del mínimo número de documentos un 40% del total de los documentos, porque filtraríamos de manera excesiva y empezaríamos a quitar palabras que sí son importantes para la clasificación.

Después de haber realizado el filtrado de longitud mínima de palabra y el filtrado de palabras infrecuentes hacemos una gráfica con los valores de los pesos para los mismos documentos que se habían seleccionado anteriormente (ver Ilustración 11), se puede observar que han desaparecido los valores más elevados que corresponden a las palabras más infrecuentes, pero no son fácilmente observables aquellos valores que se han quitado porque no cumplían el mínimo de longitud de palabra, porque no tienen valores característicos.

Hay que destacar la gran cantidad de datos que se han quitado:

- Los datos originales ocupaban en memoria 35,8 MB con un total de 5.076.664 valores de pesos repartidos en 25.132 palabras en 202 documentos.

- Los datos filtrados ocupan en memoria 2,75 MB con un total de 334.916 valores de pesos repartidos en 1.658 palabras en 202 documentos.

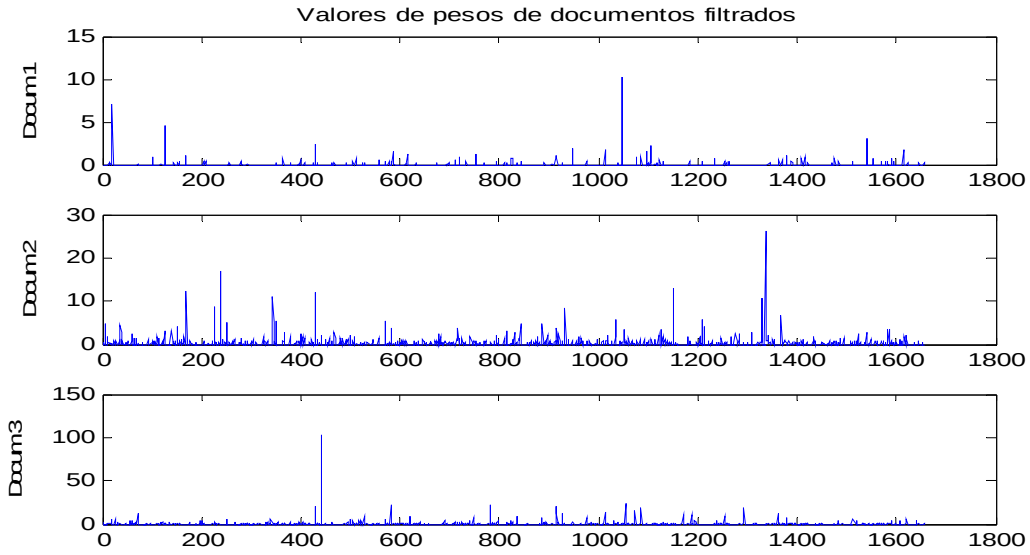

Ilustración 6.4 Vector de pesos

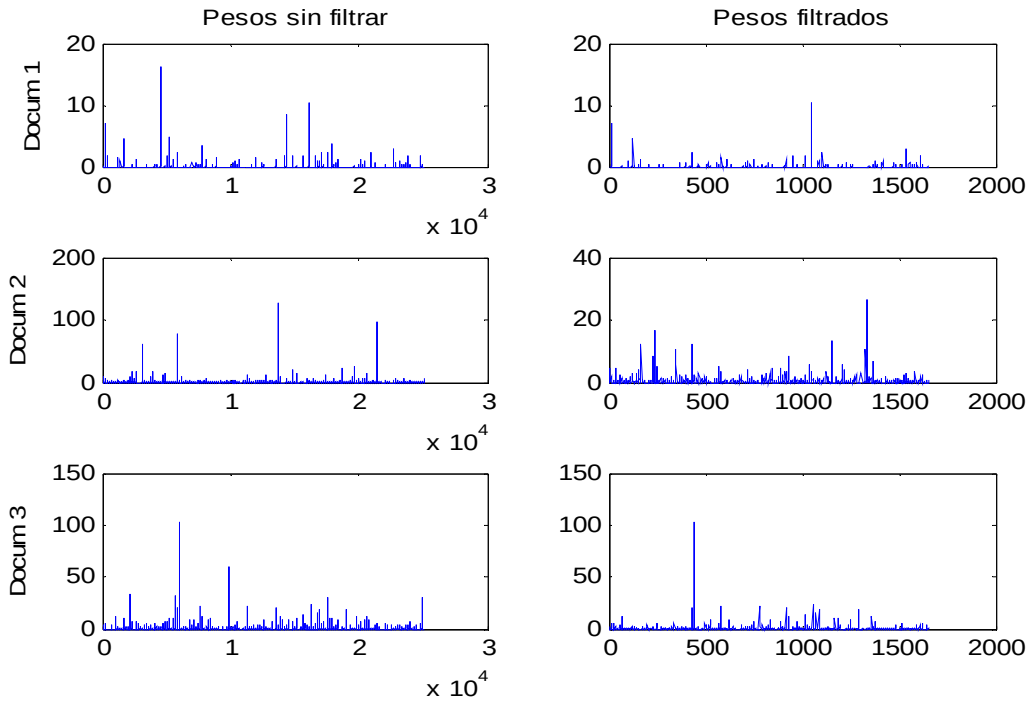

Ilustración 6.5 Comparación datos matriz V sin filtrar con filtrados

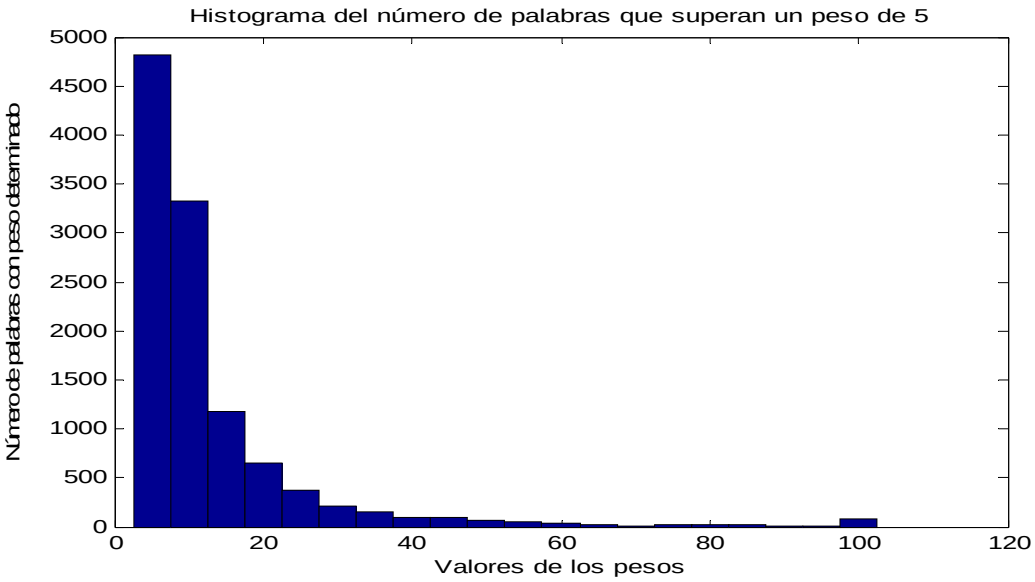

Ilustración 6.6 Histograma de pesos

Este histograma muestra los pesos de las palabras que poseen un peso mayor de 5 para mostrar sólo las palabras relevantes, se observa que la cantidad de palabras decrecen con el aumento del valor del peso, salvo en un valor de 100 porque agrupa a todos aquellos que quedan por encima de este valor, pero la tendencia es que existen menos palabras que tienen un peso muy elevado, y es preciso filtrarlas porque su importancia relativa tras la ponderación es muy grande como hemos mencionado anteriormente.

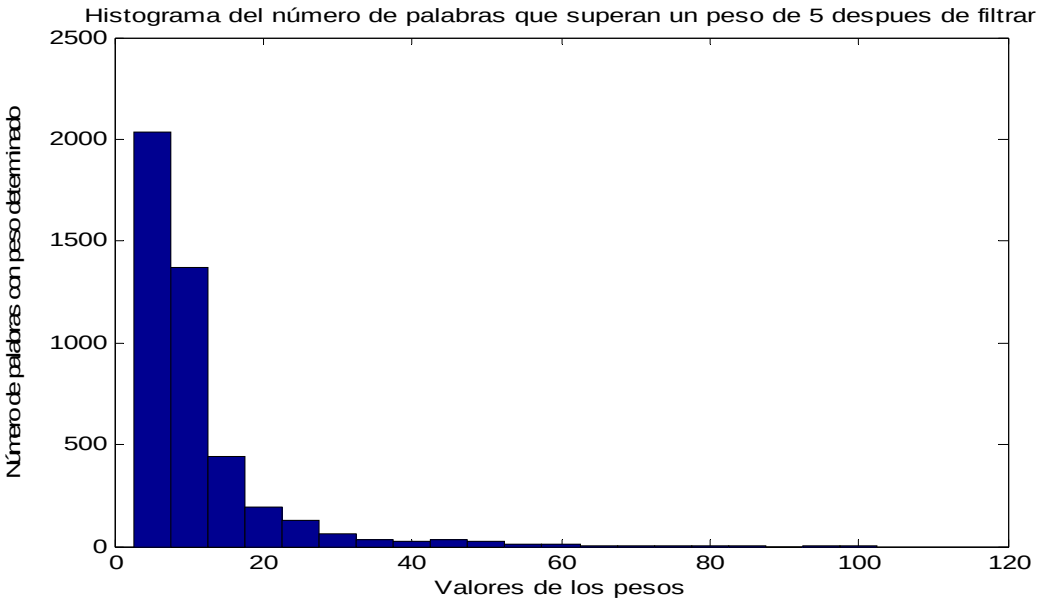

Ilustración 6.7 Histograma de pesos después de filtrar

Tras haber aplicado el filtro del número de palabras que aparecen en un mínimo número de 15 documentos, y mirando aquellas palabras con un peso superior de 5, al igual que en la anterior gráfica, se observa que ha disminuido la cantidad de palabras con un peso alto debido a la tendencia de la fórmula. Este filtrado es selectivo porque afecta sólo al colectivo de palabras que tienen un peso elevado y no son palabras relevantes, sin afectar a todas aquellas palabras que tienen un peso elevado pero debido a que son palabras relevantes.

Habría sido más fácil quitar los pesos que son altos, pero no respetaría aquellas palabras importantes para la clasificación, es decir con un peso alto que ha sido bien calculado por la fórmula, por tanto este filtro va a tener resultados visibles y que se pueden constatar mirando en el programa las palabras restantes que identifican de verdad a un tema determinado.

6.3 Reducción de la dimensionalidad de los datos

Después de aplicar los filtros descritos anteriormente, seguimos con valores muy elevados de dimensionalidad en la matriz V, este hecho no favorece el posterior procesamiento de los algoritmos de clustering ya que todavía existe un número muy elevado de datos en la matriz V.

Por tanto hay aplicar el algoritmo de factorización de matrices no negativas descrito anteriormente sobre la matriz V, la cual contiene los pesos de cada una de las palabras en cada uno de los documentos.

Los parámetros que hay que indicar en el momento de hacer una llamada para el algoritmo de factorización de matrices no negativas, son el parámetro de la dimensión de salida que se desea obtener y los datos a los cuales es necesario reducir su dimensionalidad.

A la salida de este algoritmo se obtienen dos matrices que vamos a llamar matrices W y H, cada una de las cuales contiene en sus filas o en sus columnas una nueva dimensión que es menor que la dimensión inicial de la matriz V, esta dimensión tiene relación con los temas existentes en los documentos. Se define un tema como el representante de un conjunto de palabras, las cuales van a identificar el contenido del tema, por lo tanto una

ventaja de aplicar esta técnica es la capacidad de descubrir los temas más diferenciados que se encontrarían en el conjunto de los documentos.

Se comenta a continuación de manera más detalla el contenido y la utilidad de las siguientes matrices W y H. La matriz V se puede descomponer en las dos matrices que se muestran en la siguiente imagen W y H, a partir de la siguiente expresión:

$$V=WH$$

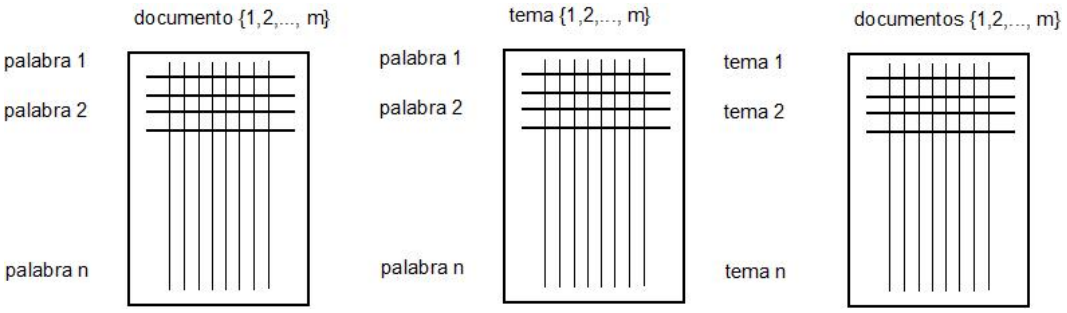

Ilustración 6.8 Matrices V, W y H

Se puede comprobar el tipo de datos que contiene cada una de estas matrices en sus filas y en sus columnas, para el caso de la matriz V, la primera de ellas contiene en sus filas las palabras, que son las palabras más importantes de todos los documentos, como se ha comentado con anterioridad, y en las columnas el conjunto de los documentos que se especificaron a la entrada de los datos. La matriz W, la que ocupa el lugar central posee como datos las palabras en sus filas, y los temas que se encontraron en el algoritmo de factorización de matrices no negativas en sus columnas, un tema es una agrupación de unas palabras concretas que describen de manera unívoca ese tema. La matriz H, la última de todas posee como datos a los temas en sus filas y a los documentos en sus columnas, por lo tanto se puede comprobar que multiplicando entre si las matrices W por H, desaparecen los temas en ambas matrices, debido a las propiedades de la multiplicación de matricial, quedando como datos las palabras y los documentos que son los datos que contiene la matriz V.

6.3.1 Matriz W

La matriz W se identifica por contener la relación de palabras que no se han filtrado en sus filas y los temas más importantes en los que se descompuso la matriz V en sus columnas, como se puede ver en la siguiente ilustración.

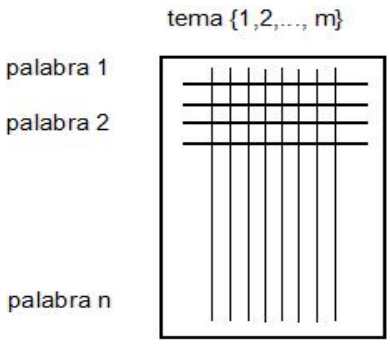

Ilustración 6.9 Matriz W

Para el conjunto de los datos pertenecientes a la matriz V resultantes de calcular los pesos de cada una de las palabras de todos los documentos con un total de más de 5E6 datos, se obtienen 1658 palabras y 20 temas tras realizar todos los filtrados sobre y aplicar el algoritmo de factorización de matrices no negativas sobre la matriz V.

Se muestran en conjuntos de 5 gráficas para ver más cómodamente para cada tema la relación de palabras que los forman hasta llegar a un total de 20 temas, que son el total de temas que se han especificado como parámetro de entrada para el algoritmo NMF en este ejemplo. Los valores estarán comprendidos entre 0 y 1 ya que los datos se han normalizado.

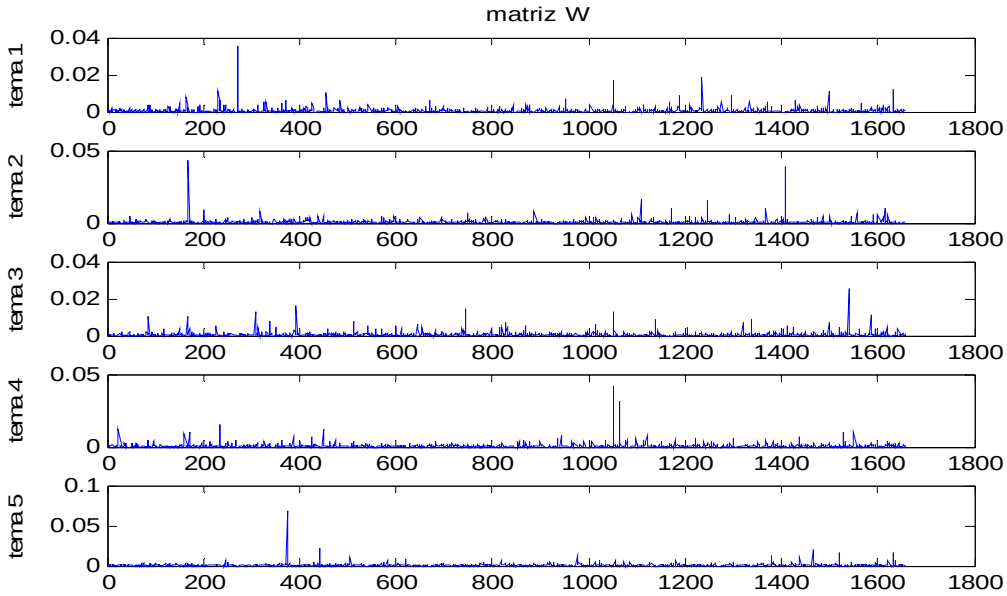

Ilustración 6.10 Temas pertenecientes a la matriz W

La importancia que una palabra tiene para un tema determinado se muestra en la magnitud que presente esta palabra en el eje de ordenadas, las palabras se encuentran colocadas alfabéticamente en el eje de abscisas.

Cada uno de los temas se identifica con un conjunto de palabras diferente y eso se puede comprobar a simple vista en las gráficas, porque no hay 2 temas que se compongan exactamente de las mismas palabras.

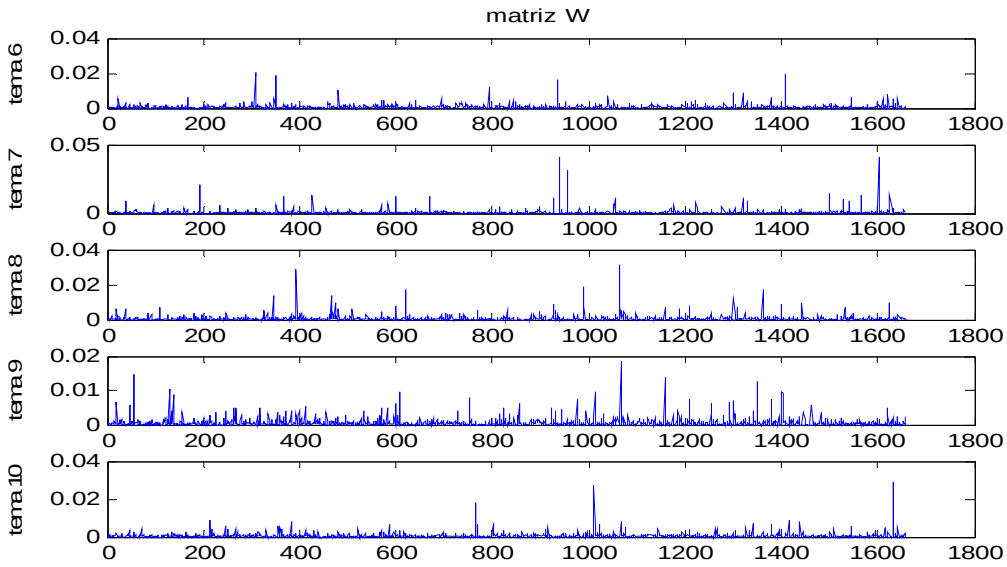

Ilustración 6.11 Temas pertenecientes a la matriz W

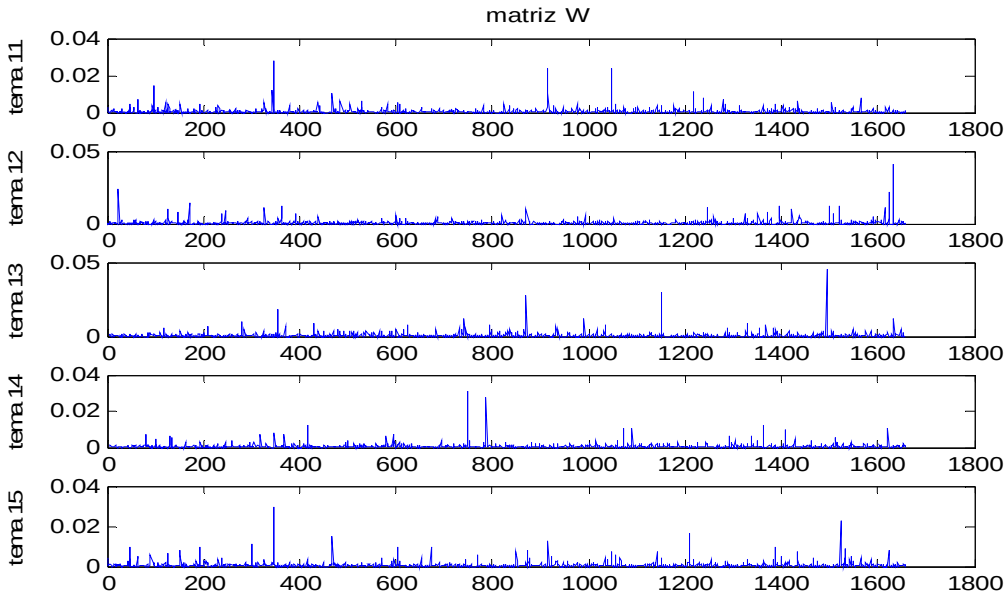

Ilustración 6.12 Temas pertenecientes a la matriz W

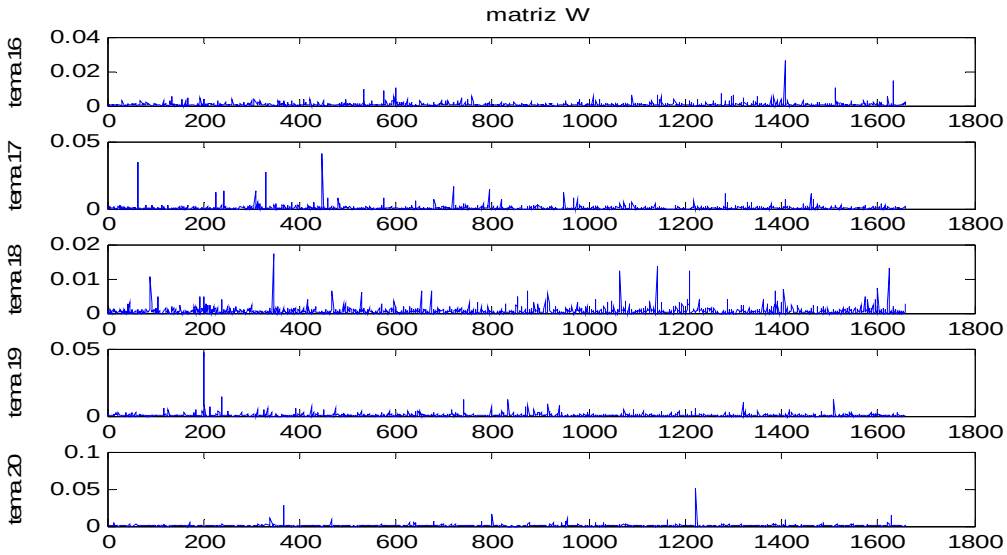

Ilustración 6.13 Temas pertenecientes a la matriz W

Sumando en la matriz W todas las palabras que componen un tema se obtiene como resultado la unidad para cada uno de los temas debido a que previamente se normalizó esta matriz.

Otra manera de interpretar los datos puede ser colocando de forma ordenada en un vector aquellas palabras que poseen mayor importancia para cada uno de los temas sin hacer distinción entre los temas, con lo que se obtienen en orden decreciente de importancia, las palabras más representativas del conjunto de los documentos como se puede ver en la siguiente gráfica.

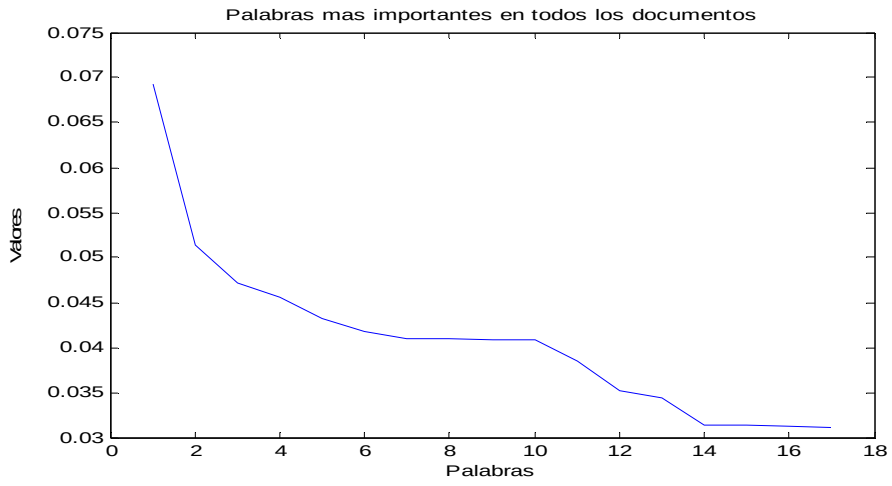

Ilustración 6.14 Valores de los pesos de las palabras más importantes

Se pueden interpretar las palabras que poseen unos valores más altos de pesos en la matriz W, como las palabras que serán más útiles para el clasificador, debido a que aportan una distinción mayor entre los temas porque son palabras más específicas, es decir, que aparecerán en un menor número de documentos, se muestran estas palabras en la siguiente tabla.

Palabras más importantes de todos los documentos
Denoising
Registration
Cell
Telomeres
Box
Patient
Motion
Velocities
Drop
Wavelets
Splines
Compress
Angles
Pet
Interpolants
Myocardial

Tabla 6.3 Palabras más importantes

Se selecciona para cada uno de los 20 temas la palabra que contiene un peso mayor de todas las que componen un mismo tema, y se hace esto para cada tema, mostrando el valor máximo del peso de cada una de esas palabras en la siguiente gráfica, con lo que se observa que la palabra considerada como más importante de todos los documentos se encuentra en el tema 5 que es *denoising*.

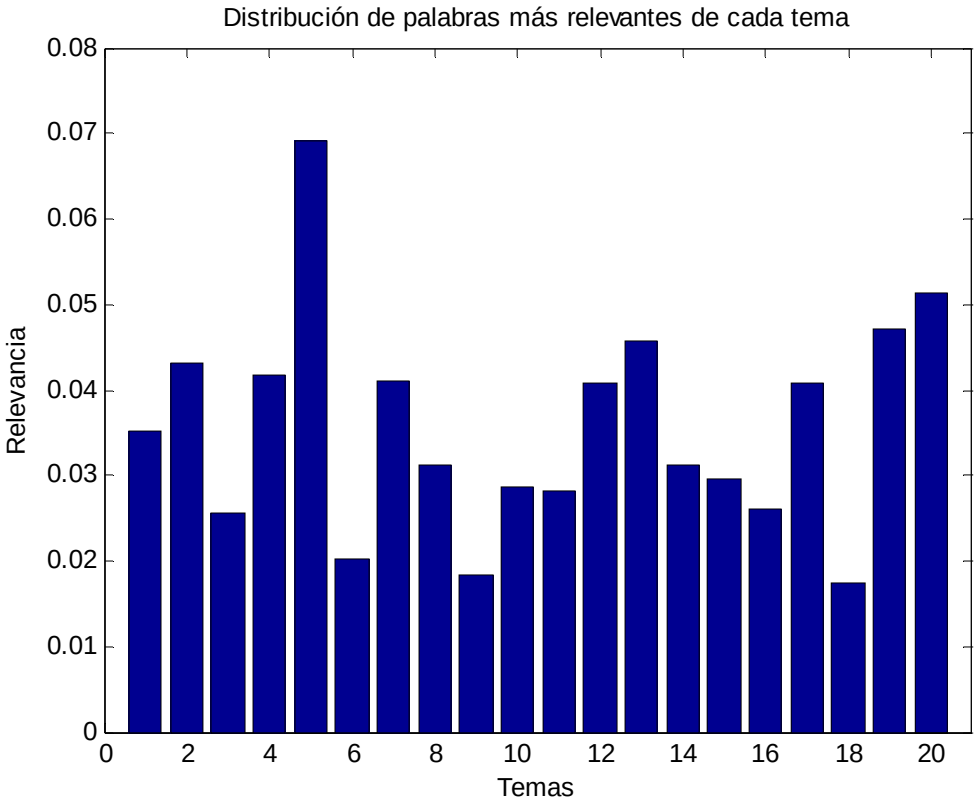

Ilustración 6.15 Pesos de palabras más relevantes por temas

En la siguiente gráfica de la izquierda se muestra el valor que tienen las palabras más importantes de todos los documentos sin hacer discreción entre los temas en los que pueda aparecer esa palabra, en cambio en la gráfica de la derecha se muestra el valor que tienen las palabras más importantes de cada uno de los 20 temas existentes ordenadas descendentemente por el valor de sus pesos, por lo tanto solamente encontraremos una palabra de cada tema. Sólo serán iguales la gráfica de la derecha y la gráfica de la izquierda en el caso en el que las palabras más importantes de todos los documentos ordenadas descendentemente fuesen de cada uno de los temas.

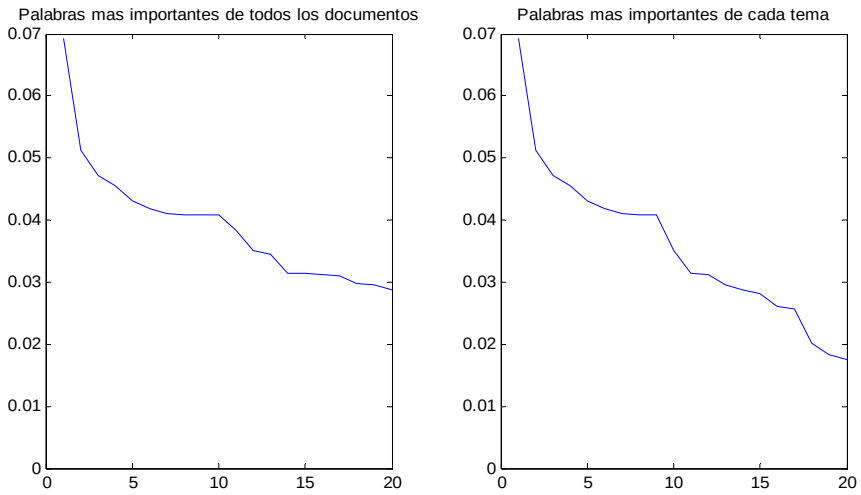

Ilustración 6.16 Palabras importantes de los documentos y de los temas

Se muestran en la siguiente tabla las palabras más relevantes de cada uno de los 20 temas que se han mostrado previamente.

TEMAS	PALABRAS
Tema 1	compressed
Tema 2	box
Tema 3	trace
Tema 4	patient
Tema 5	denoising
Tema 6	contour
Tema 7	motion
Tema 8	pet
Tema 9	phase
Tema 10	wavelets
Tema 11	micrographs
Tema 12	active
Tema 13	telomeres
Tema 14	interpolants
Tema 15	ctf
Tema 16	splines
Tema 17	drop
Tema 18	crystals
Tema 19	cell
Tema 20	registration

Tabla 6.4 Palabras más importantes de cada tema

A continuación se muestra un listado de las palabras más relevantes de cada uno de los 20 temas que es el número de temas en los que se descompuso la matriz V.

TEMA 1	TEMA 2	TEMA 3	TEMA 4	TEMA 5
compressed	box	trace	patient	denoising
rendered	spline	tracking	pet	dot
patient	prefilter	detectors	clinics	sured
wavelets	resampled	interfaces	active	threshold
classify	sites	paths	dynamically	wavelets
templates	ville	contour	treatments	noise
echo	quasi-interpolation	user	tissues	snr
ray	cell	argument	brain	estimating
saved	matrix	border	blood	subbands
bodies	convolution	program	mri	gaussian
multiscaling	interpolants	shaped	principal	orthonormality
heart	truncated	criteria	disease	fittings
clinics	convolution	convolution	study	white
classify	polynomial	templates	desco	peaked
enhancing	van	segmented	monitorization	convolution
delayed	sample	line	scan	coefficients
stressing	vector	optimal	regional	let
cost	reconstruction	graphically	positron	random
quality	tensor	manually	platform	colores
rigid	transforming	circle	emission	jun
ring	dual	extraction	components	transactions
correction	fourier	stabilizing	treat	model
severe	figs	frame	decreases	statistical
sizing	domain	length	preprocessed	scaled
decomposition	targeted	recognition	pages	minimized
pass	aliasing	converges	sites	distance
binaries	dilated	list	manages	filter
ultrasound	ieee	growing	plan	poisson
displacing	operability	detectable	area	soft
view	filter	stopping	radiologic	transforming
conventional	discretization	vision	medicalicinal	microscope
subbands	green	level	reson	probability
border	kernels	curve	soft	variance
matches	colores	threshold	reports	optimal
maps	scheme	windows	tomography	nonlinear

Tabla 6.5 Listado de palabras de cada tema

TEMA 6	TEMA 7	TEMA 8	TEMA 9	TEMA 10
contour	motion	pet	phase	wavelets
spline	velocities	detectors	analogies	optimal
curve	myocardial	nucleo	pulses	wave
moment	cardiac	gate	signal	isotropic
knots	temporal	simulation	band	channel
energy	displacing	crystals	frequencies	stack
scaled	ultrasound	elasticity	spectral	subbands
segmented	wall	scanner	optimal	phase
convolution	heart	subset	spectrum	depth
parametrization	deformational	emission	beam	laplacian
smoothing	frame	voxels	intrinsic	sharp
boundaries	segmented	spectrum	snr	orthonormality
transactions	mode	mode	recording	iteration
windows	peaked	frame	noise	soc
ieee	ventricular	reconstruction	bandwidth	focusing
video	tissues	nuclear	scan	coefficients
cubic	tracking	sci	amplitudes	transforming
wavelets	sequence	tomography	sample	daubechies
active	age	pulses	acquisition	plane
looping	registration	phantom	frame	microscopies
filter	flow	attenuation	magnet	ville
regional	patient	energy	reson	windows
gradient	assessment	events	aliasing	december
figs	field	acquisition	reconstruction	separating
multiscaling	radial	affinity	supportpress	optimalization
localize	cycle	ray	acquire	decomposition
shaped	clinics	list	digits	experimental
conf	rigid	iteration	compensated	restore
september	medical	modules	mirrored	retrieved
sequence	desco	physical	vivo	reconstruction
edge	left	correction	fittings	propagation
multiresolution	axis	figuration	scanner	colores
constraint	colores	medicalicine	figs	application
machine	echo	pixel	components	complex
pattern	regional	electron	light	textural

Tabla 6.6 Listado de palabras de cada tema

TEMA 11	TEMA 12	TEMA 13	TEMA 14	TEMA 15
micrographs	active	telomeres	interpolants	ctf
particle	voxels	protein	kernels	tilted
assigned	brain	maps	dirac	reconstruction
cryo-electron	decomposition	databases	sinc	electron
refinable	temporal	web	polynomial	microscopies
electron	wavelets	nuclear	piecewise	constraint
ctf	spatial	interactive	convolution	herman
microscopies	threshold	concept	spline	sorzano
replicative	correlate	distance	cubic	frank
ultramicroscopy	ville	sequence	degree	carazo
angular	resampled	sites	approximates	alignment
ring	axial	pairing	formula	tomography
experiment	statistical	knowledge	convolution	marabini
structured	maps	generate	finite	biology
enhancing	coefficients	modules	sample	convolution
frank	biased	ceu	band	particle
correlate	slice	molecules	shannon	structural
termined	detectable	del	theorem	projection
domain	clustered	automatization	basic	macromolecules
averaged	tested	similar	asymptotic	axis
background	significantly	instance	strange	volume
rotated	sensitivity	softwares	ieee	biology
likelihood	framework	rules	periodized	art
sorzano	level	pages	theory	iteration
alignment	null	workshop	fourier	penczek
fittings	rest	mont	support	miss
noise	van	http	interpolatory	angles
frequencies	subbands	framework	orders	specimens
carazo	hypothesis	integrability	error	transfer
penczek	domain	conference	signal	gridded
estimating	non	org	interval	pages
biology	scaled	future	scheme	ultramicroscopy
simulation	daubechies	source	communication	phantom
log	human	translating	satisfied	packag
structured	connectivity	internal	constant	protein

Tabla 6.7 Listado de palabras de cada tema

TEMA 16	TEMA 17	TEMA 18	TEMA 19	TEMA 20
spline	drop	crystals	cell	registration
wavelets	angles	projection	clustered	deformational
theorem	cos	voxels	textural	landmark
fraction	inf	phantom	live	warping
experiment	knots	reconstruction	intensities	mutual
filter	coef	convolution	segmented	registered
riesz	contour	art	microscope	criterion
scaled	multiplies	vector	cellular	elasticity
proofs	cient	sphere	morphological	spline
polynomial	rotated	gridded	marker	pyramid
basic	support	electron	channel	histograms
fourier	energy	marabini	discriminated	jointly
convolution	filter	herman	labeling	multiresolution
proposition	edge	sorzano	manually	overlapping
invariant	neighborhood	experiment	count	regularization
solute	let	microscopies	automatization	optimal
signal	surface	cell	detectable	accuracy
operability	spline	atoms	regional	gradient
satisfied	noise	united	emission	convolution
cardinal	homogeneous	carazo	controlled	geometric
bound	gradient	madrid	background	modality
integer	reflectance	recognition	popular	brain
smoothing	distance	relaxes	dynamically	mri
self	derivatives	rowing	camera	mask
theory	pixel	optimal	correction	targeted
differentiability	rules	variance	pixel	gold
regularization	series	clustered	individual	coordinating
compacting	polynomial	dimensional	stain	convolution
norm	shaped	simulation	matrix	pixel
continued	controlled	update	tested	subpixel
properties	profile	particle	classify	distorting
biorthogon	vertical	plane	protein	alignment
generalized	smoothing	fourier	automatization	robust
causal	converges	spain	red	coarsely
equivalently	journal	methodological	sensitivity	slice

Tabla 6.8 Listado de palabras de cada tema

Cada uno de los 20 temas que se han mostrado, quedará identificado por las palabras que lo describen en su listado de palabras. Se observa que algunos temas comparten algunas de las palabras más importantes con otros temas, esto se debe a que esos temas tienen cierto parecido entre sí.

6.3.2 Matriz H

Otra matriz resultante de la factorización de la matriz V, es la matriz H que está formada por los temas en sus filas y los documentos en sus columnas como se puede ver en la siguiente ilustración:

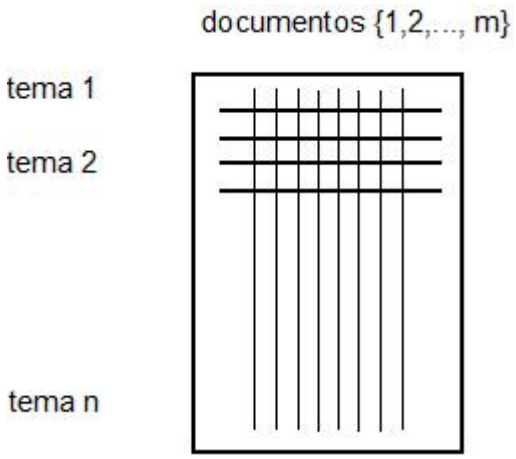

Ilustración 6.17 Matriz H

Se muestra a continuación de manera gráfica el contenido de la matriz H, donde se pueden ver todos los documentos que son relevantes para un tema determinado, es decir, aquellos que tienen un valor más elevado resultante de la factorización de matrices no negativas, desde el tema 1 hasta el tema 20.

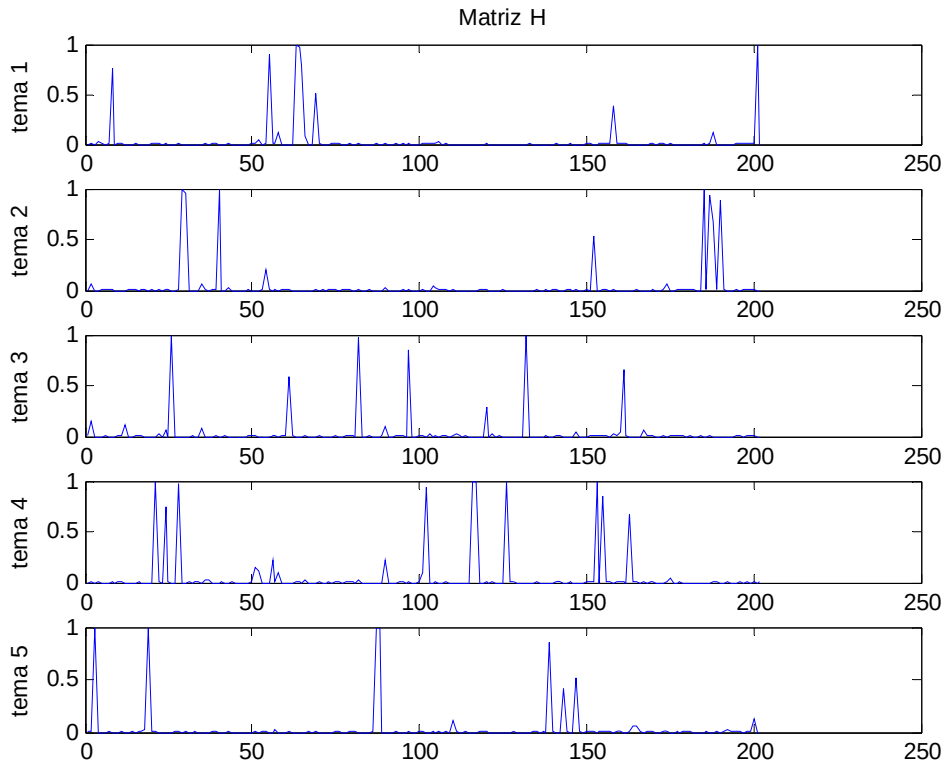

Ilustración 6.18 Composición de temas de la matriz H

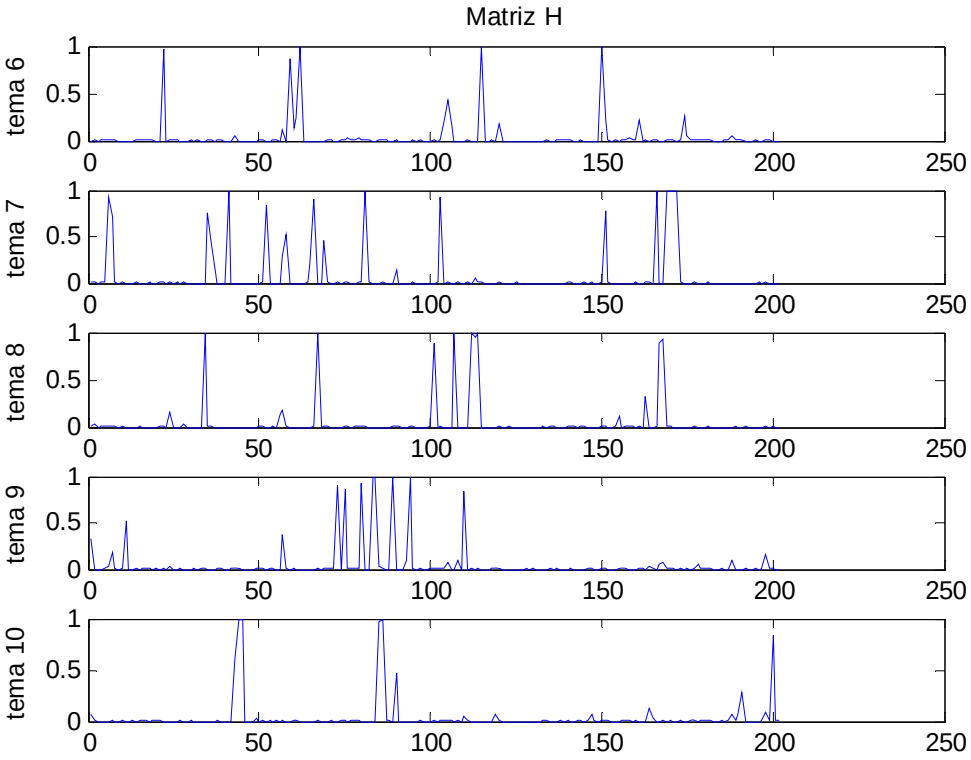

Ilustración 6.19 Composición de temas de matriz H

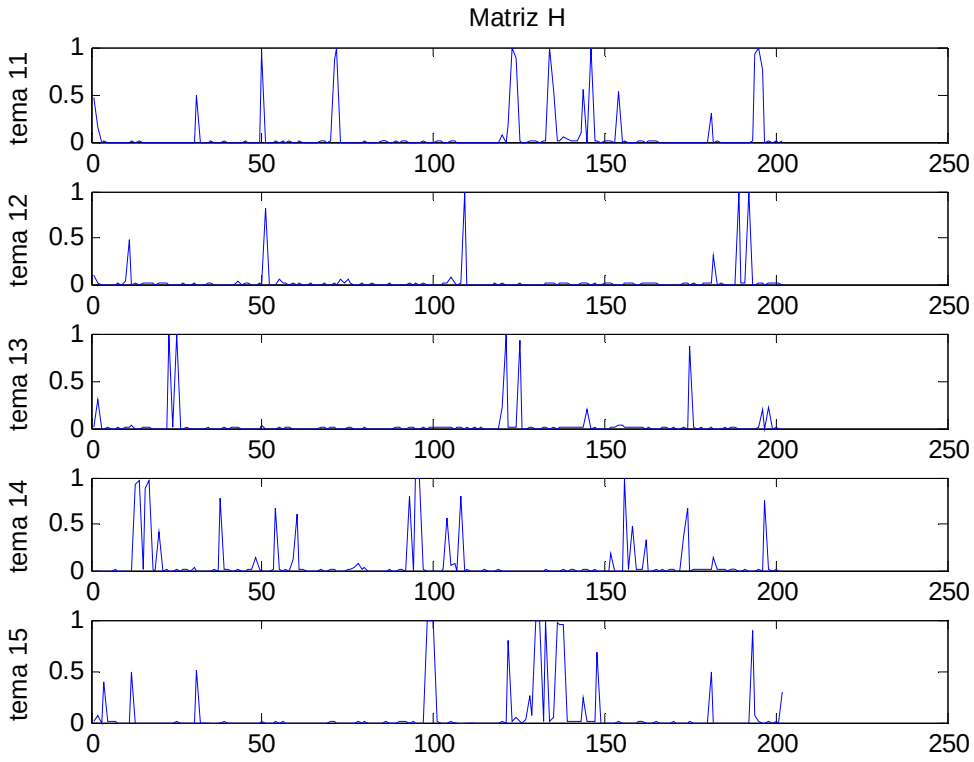

Ilustración 6.20 Composición de temas de matriz H

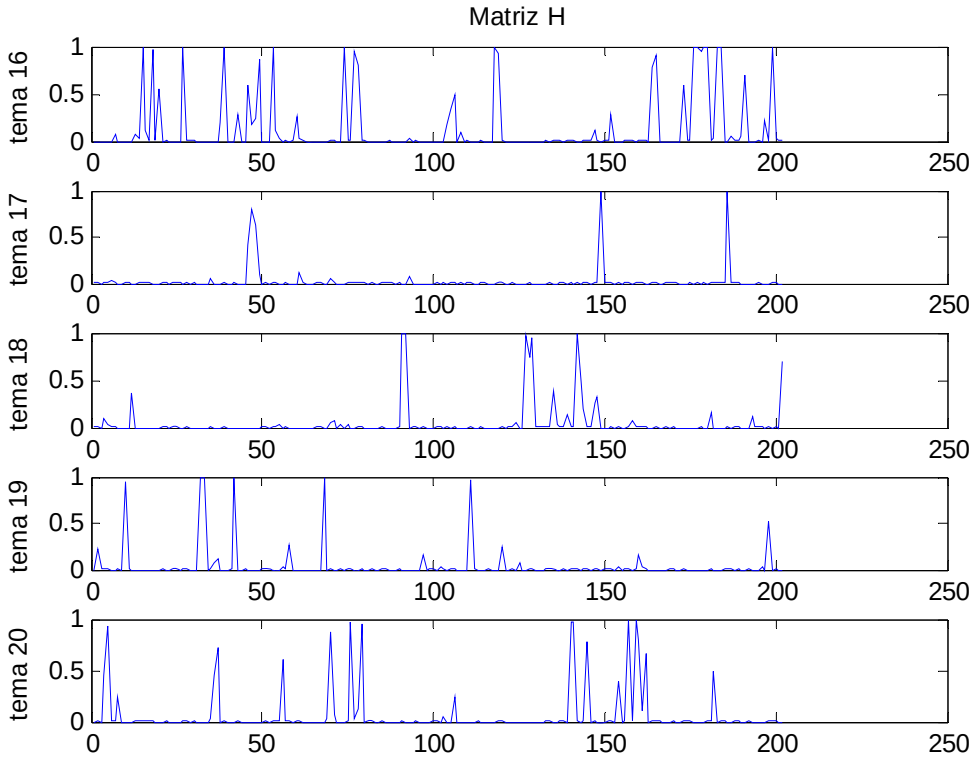

Ilustración 6.21 Composición de temas de matriz H

El hecho de que aparezcan en la gráfica pequeños máximos, se debe a que los documentos que hablan de temas parecidos son del mismo autor porque se encuentran ordenados alfabéticamente, por lo tanto son correlativos en cuanto al número que los identifica como documento, y hay grupos de documentos que alcanzan los máximos valores de pertenencia con un tema en lugares próximos en el eje de abscisas.

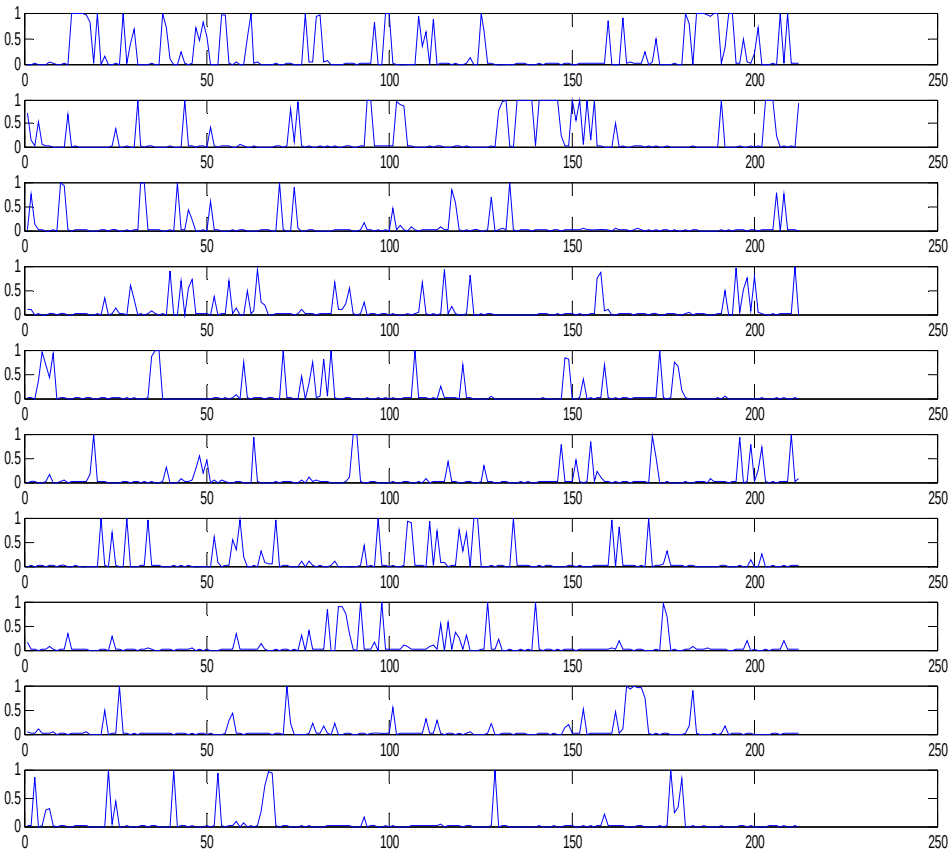

Ilustración 6.22 Descomposición temática para ejecución con 10 temas

En este ejemplo se observa el contenido de la matriz H para una descomposición de 10 temas resultado de otra ejecución diferente del algoritmo de factorización de matrices no negativas, donde seleccionando como parámetro de entrada menos temas, se puede comprobar de manera visual que los vectores que componen la matriz H contienen sus valores máximos en lugares más diferenciados que en el caso de especificar un mayor número de temas, esto es debido a que cuantos menos temas se especifiquen para la factorización, más diferentes serán los temas que se encuentren.

Con este requisito de introducir la máxima distinción posible en los temas seleccionando para ello un número menor de temas, se refleja de una forma más acentuada que documentos con nombres similares, es decir, del mismo autor, hablan del mismo tema, y los documentos de otros autores hablarán de otros temas distintos entre sí, por tanto no vamos a encontrar un conjunto de documentos del mismo autor hablando de 2 temas muy diferentes simultáneamente.

Otra manera de interpretar la matriz H consiste en mirar de qué temas se compone cada uno de los documentos. Debido a la cantidad de documentos que existen en el cuerpo de documentos, más de 200 documentos, se escogen 6 documentos a modo de ejemplo para mostrar los valores que contienen los vectores de esos documentos seleccionados.

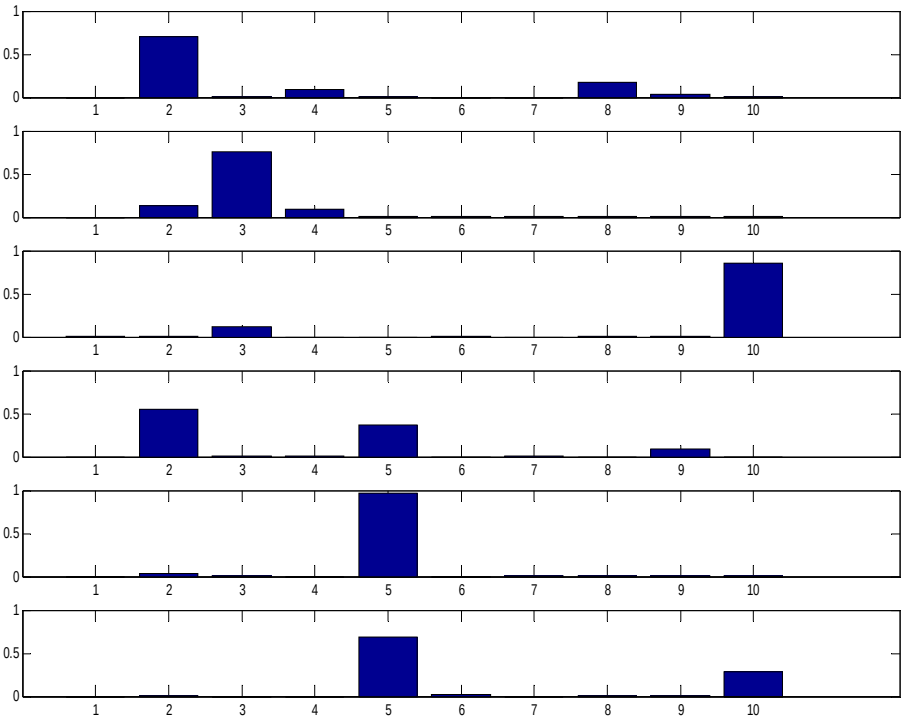

Ilustración 6.23 Descomposición temática por documentos para ejecución con 10 temas

El diagrama de barras alcanzará valores elevados si es importante el número de tema que refleja el eje de abscisas. La mayoría de los documentos tienen relación con varios temas, esto ocurre porque un documento cualquiera pocas veces habla solo de un tema.

Aumentando el número de temas que se calculan con NMF a 20, se encuentran temas más parecidos entre sí, por lo tanto es más probable que un documento cualquiera tenga más relación con diferentes temas como se puede ver en la siguiente gráfica.

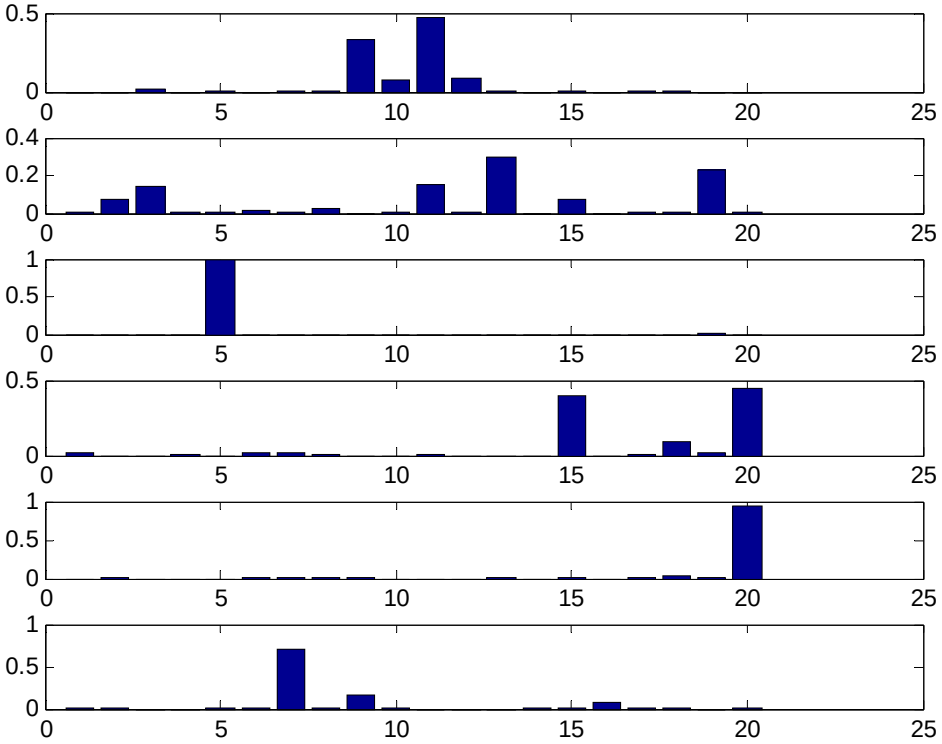

Ilustración 6.24 Descomposición temática por documentos para ejecución con 20 temas

El algoritmo NMF empieza creando las matrices W y H con números aleatorios, por lo tanto con los mismos datos de entrada, el resultado a la salida no va a ser siempre el mismo, para evitar cualquier tipo de aleatoriedad se ejecuta varias veces NMF con el objetivo de promediar los resultados. Existen varias técnicas para conseguir obtener datos más precisos que se describirán de manera individual más adelante.

6.4 Recuperación de la matriz V

Al multiplicar las matrices W y H entre sí, se obtiene como resultado la matriz V inicial que contiene la relación de palabras y documentos con sus correspondientes pesos, hay que precisar que no se necesita la matriz recuperada para realizar agrupamientos o cálculos posteriores, pero permite comprobar que la factorización de esta matriz se

realizó correctamente en el caso de que el error entre la matriz V original y la matriz V recuperada sea pequeño.

Para comprobar de manera práctica en qué casos se produce menos error, se calcula el error de recuperación como la desviación estándar existente entre la matriz V original y la matriz V recuperada, en función el número de temas seleccionados en el algoritmo de factorización de matrices no negativas, y se observa en la siguiente gráfica que este error tiende a decrecer cuanto mayor es el número de temas.

El error de recuperación se calcula a partir de los datos de la matriz V recuperada, comparando con la matriz V original, de la siguiente forma:

V_recup=WH

Diff=V-V_recup

Cada uno de los documentos que componen la matriz V, es un vector de datos que contiene los pesos de todas las palabras, por lo tanto para cada uno de los vectores de la matriz diferencia calculada anteriormente se calcula su desviación estándar de la siguiente manera:

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$s = \sqrt{\frac{\sum_{i=1}^{n} x_i^2 - n\overline{x}^2}{n-1}}$$

Una vez que se obtiene la desviación estándar para cada uno de los documentos, sólo queda representar el error del conjunto de los documentos, se calcula la media de este vector de desviaciones estándar a partir de la siguiente expresión:

$$\overline{s} = \frac{1}{N}\sum_{i=1}^{N} s_i$$

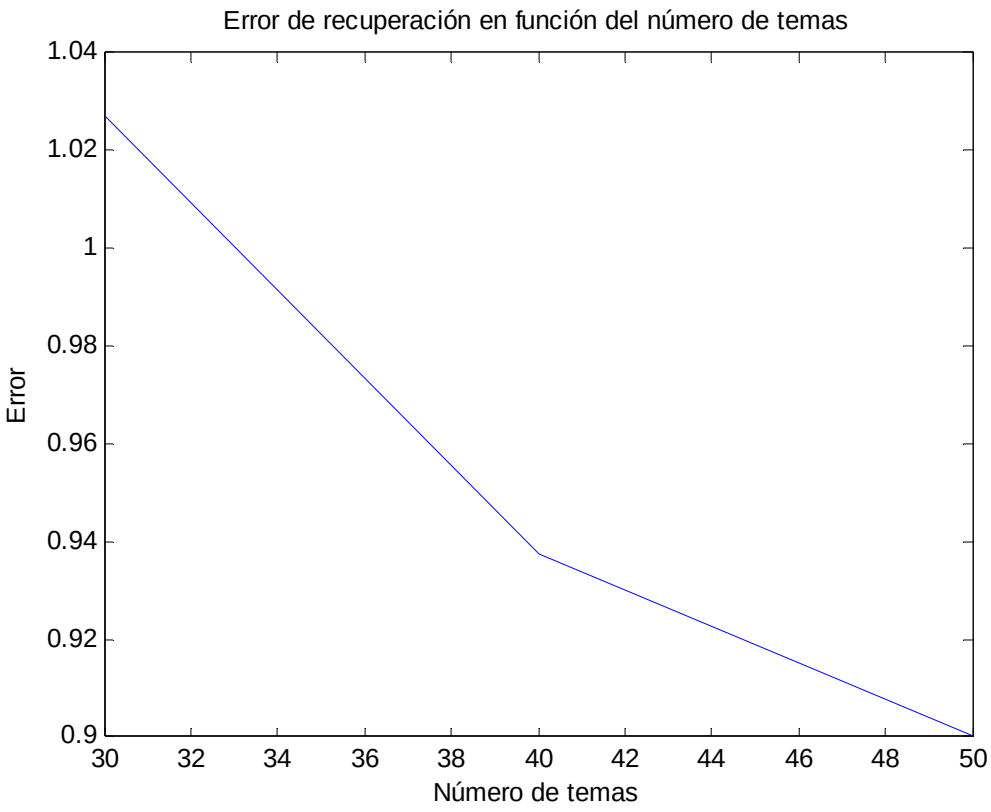

Ilustración 6.25 Error de recuperación en función del número de temas

El hecho de que la magnitud del error de recuperación tiende a decrecer con el aumento del número de temas, tiene sentido debido al funcionamiento del algoritmo NMF, lo que ocurre cuando se introduce un número de temas menor es concentrar la misma información de V que tiene alta dimensionalidad en menos dimensiones de salida, con lo que existirá menos espacio para guardar los mismos atributos y se perderá algo de precisión.

Si se ejecuta el algoritmo NMF para los datos de otra observación diferente de la anterior y se vuelve a calcular el error entre V y V' o recuperada, se observa que la curva del error sigue la misma tendencia de bajada, con lo que se puede comprobar claramente que es mejor para tener un error mínimo de recuperación aumentar el número de temas.

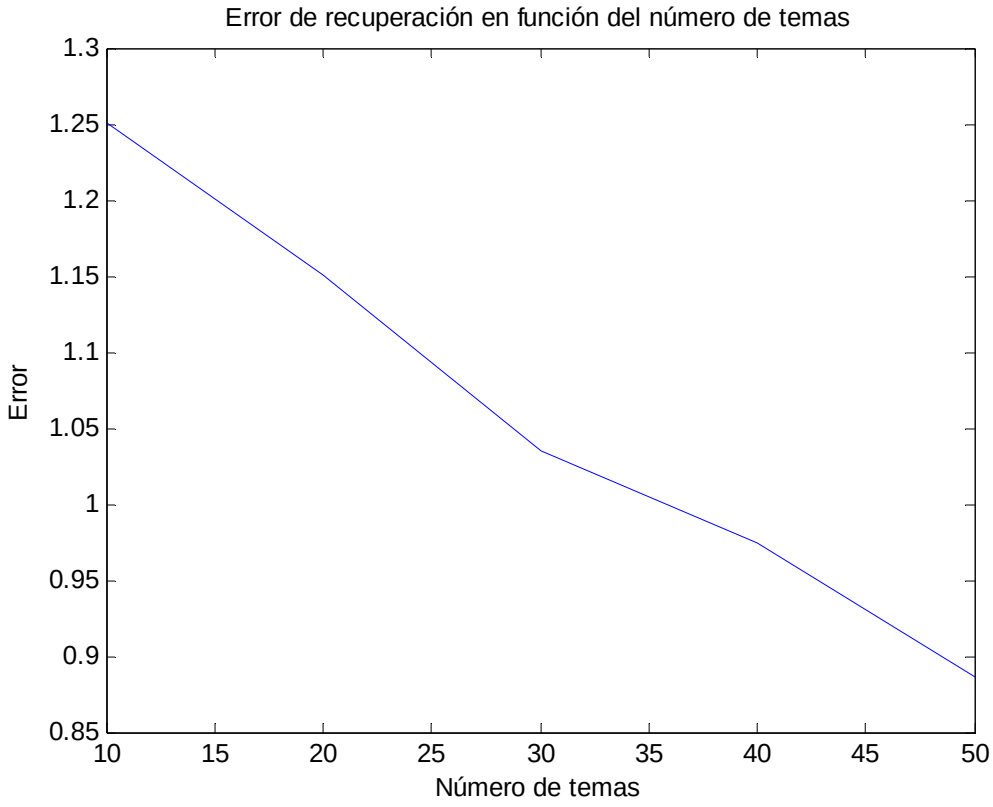

Ilustración 6.26 Error de recuperación en función del número de temas

Se podría deducir de manera errónea que es mejor seleccionar un número de temas elevado en el algoritmo NMF para disminuir el error, el problema que se deriva de introducir más temas, es que se va a reducir mucho menos la dimensionalidad de los datos en las matrices W y H, y se crean más datos a la salida, lo que genera un problema con tratamiento que dan los algoritmos de clustering a esta cantidad de datos generados que están menos discriminados por el algoritmo de factorización de matrices no negativas, además el programa tardará más tiempo en procesar todos estos datos extra, pero este hecho es asumible porque no tenemos impuesto ninguna limitación en tiempo de procesamiento.

En la siguiente gráfica se muestran las consecuencias que tiene aumentar el número de temas observando el número de elementos que posee el cluster que más elementos tiene a la salida del algoritmo K-means, todo ello promediado en diferentes ejecuciones para obtener mayor fiabilidad de los datos.

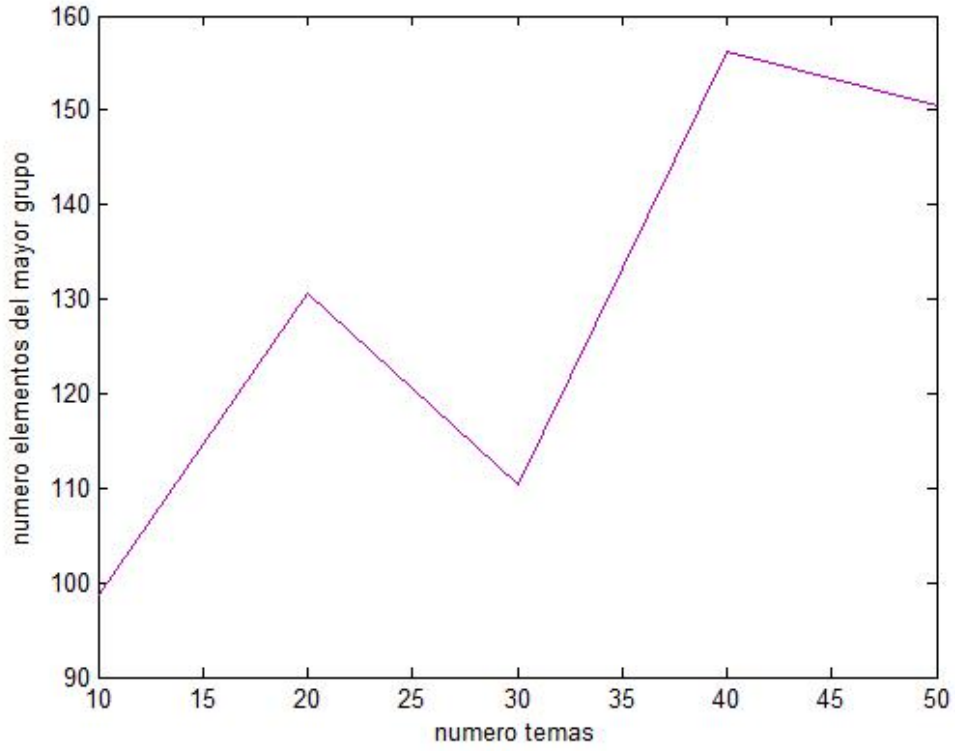

Ilustración 6.27 Número de elementos de los grupos más grandes en función del número de temas

Se deduce de esta gráfica que prefijando un número de temas muy elevado, existe un cluster que tiene muchos más elementos dentro de un mismo grupo que si hubiésemos seleccionando un menor número de temas, por tanto no es bueno para los algoritmos de clustering que los datos les lleguen menos cribados o lo que es lo mismo con una dimensionalidad más elevada porque tienen tendencia a encontrar menos diferencias entre los datos y hacen mayores agrupaciones en un solo grupo.

Por estas razones, se ha tomado la determinación de fijar en torno a 20 ó 30 el número de temas que usaremos en el algoritmo NMF, para tener un tiempo de procesamiento y un error aceptables y no comprometer el funcionamiento de los algoritmos de clustering.

En la siguiente gráfica, se muestra uno de los documentos de la matriz V, en el primer caso utilizando los datos resultantes del procesado de los documentos, es decir la matriz V original, y en el segundo caso utilizando los datos de una matriz V reconstruida, calculada a partir de las matrices W y H. Se observan parecidos entre los datos, pero no son exactamente iguales debido al error de reconstrucción previamente comentado.

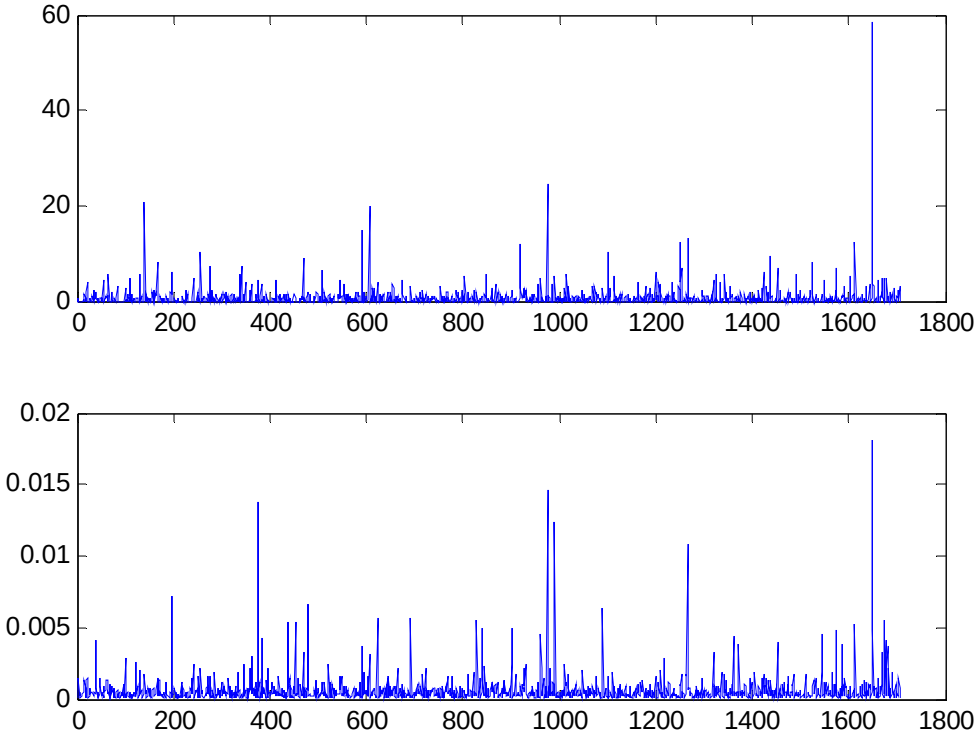

Ilustración 6.28 Comparación entre datos originales y datos recuperados en un vector de V

Se comprobó que el error disminuía cuando el número de temas aumentaba, como hemos demostrado anteriormente con la gráfica del cálculo del error, esto se refleja en la similitud de las gráficas al recuperar los datos. La gráfica anterior se ha generado con 10 temas, y la siguiente gráfica se genera con 50 temas, en la que se aprecia una reconstrucción más fiel. La diferencia de escalas entre las dos gráficas se debe a que la matriz reconstruida se calcula con la matriz H normalizada automáticamente por el programa con el fin de que el algoritmo K-medias trabaje mejor, y la matriz V original es la que se obtiene de realizar el cálculo de los pesos y no está normalizada.

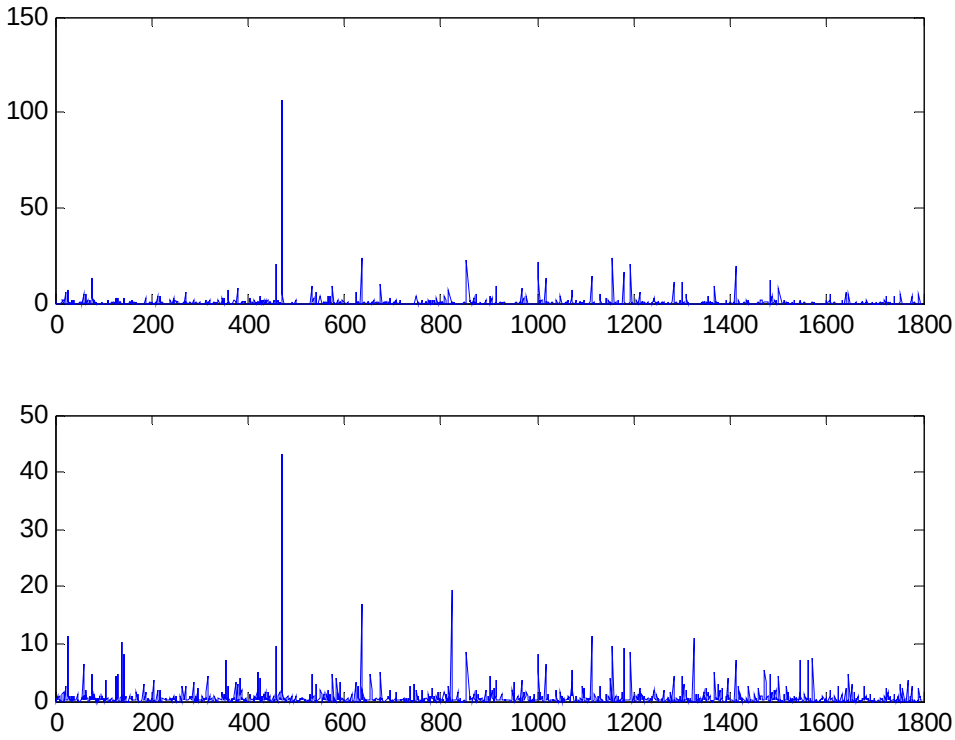

Ilustración 6.29 Comparación entre datos originales y datos recuperados en un vector de V

6.5 Kmeans

Para conseguir agrupar los documentos en relación a su contenido, se aplica el algoritmo K-means sobre los datos de la matriz H, porque esta matriz contiene las relaciones entre los documentos y los temas que se han descrito anteriormente.

Los parámetros de entrada que utiliza este algoritmo son el conjunto de datos que se quieren agrupar, y el número de grupos en los que se desea agrupar esos datos de entrada. A la salida de la función se obtiene un vector que contiene el número de grupo al que pertenece cada uno de los documentos, como se puede ver en la siguiente ilustración.

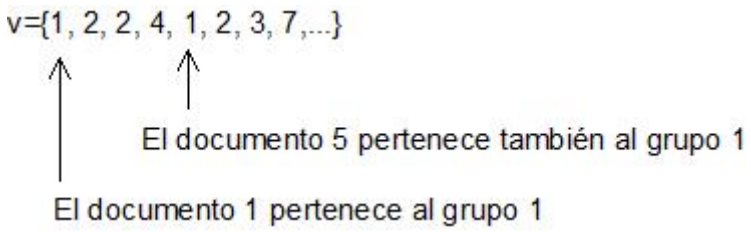

Ilustración 6.30 Composición del vector que relaciona grupos con documentos

Para poder entender de forma más clara la salida de esta función, se hace uso de la siguiente gráfica donde se muestran la silueta de los documentos que pertenecen a diferentes grupos, es decir que los documentos que pertenecen a un mismo grupo, aparecen juntos formando un bloque.

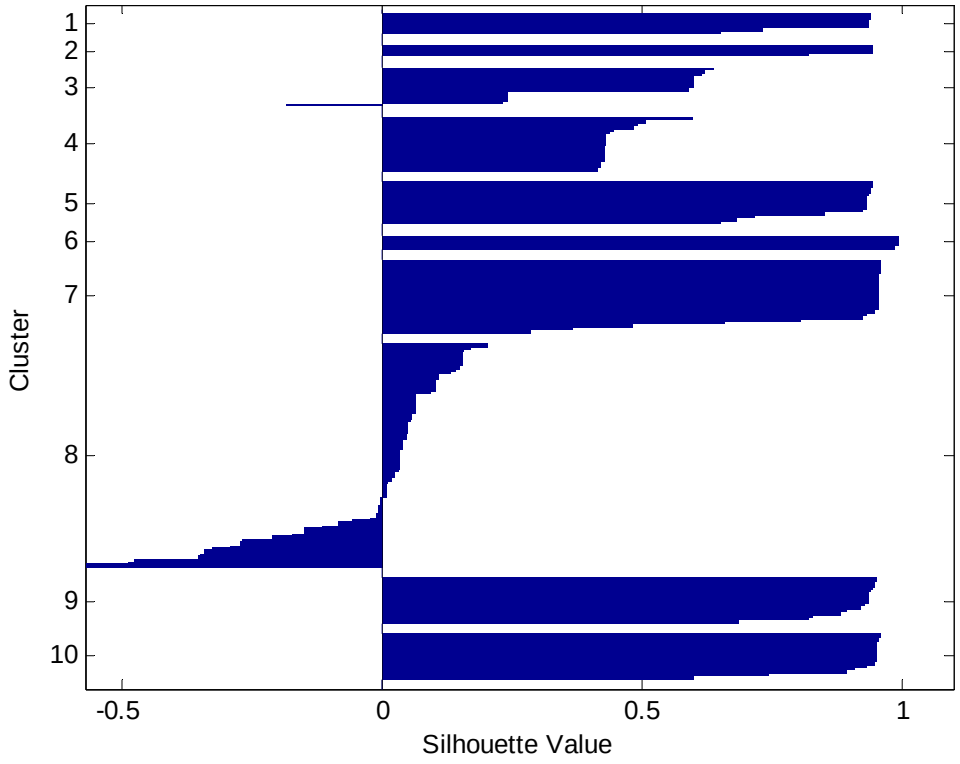

Ilustración 6.31 Silueta de documentos

El valor de la silueta para cada documento es una distancia que muestra lo que se parece cada documento al resto de los documentos dentro su propio grupo y a su vez comparando con los documentos de los otros grupos, tomando valores dentro de un rango de -1 a 1.

$$S(i) = \frac{b(i) - a(i)}{\max[a(i), b(i)]}$$

Cada valor $a_i = dist(i, A)$ es la distancia media desde el documento i hasta el resto de los documentos en su propio grupo, esto es A, y $b_i = dist(i, C)$ es la distancia media de un documento de un grupo a los documentos de los demás grupos, siendo C cualquier otro grupo distinto de A.

Existe otra manera más fácilmente visible de ver una agrupación de los documentos, el dendograma donde se pueden ver en función de la altura que se refleja en el eje de ordenadas los grupos que se van formando, como un indicador de proximidad entre los grupos. Se muestran 2 dendogramas calculados a partir de los mismos datos, en la primera gráfica las distancias entre los documentos se han calculado promediando las distancias entre pares de documentos que pertenecen a diferentes grupos, en el segundo caso se calcula la distancia euclídea sin promediar, entre un documentos perteneciente a un grupo y el más próximo de los documentos de cada uno de los diferentes grupos.

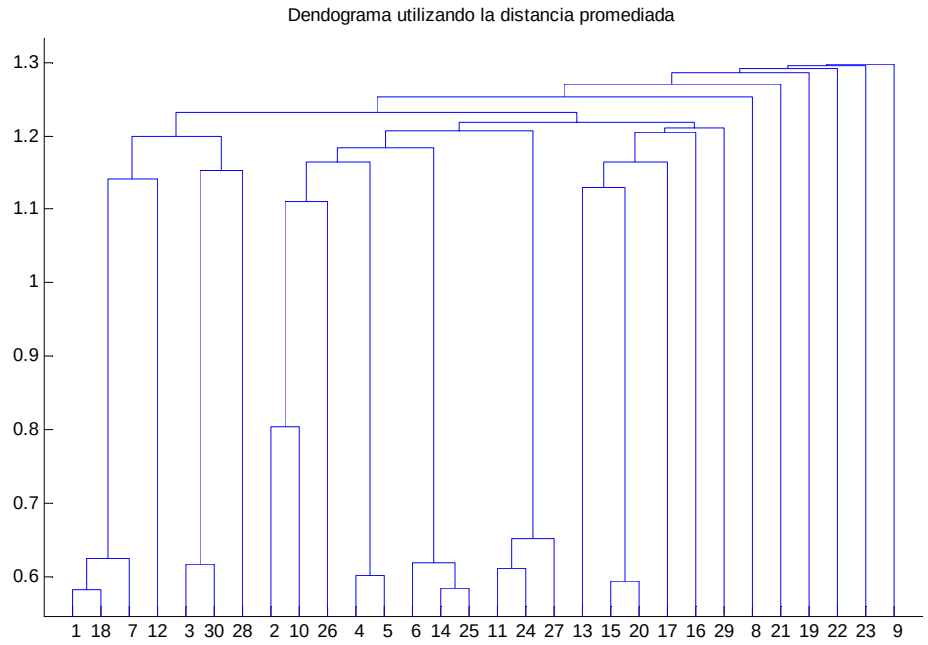

Ilustración 6.32 Dendograma a partir de distancia promediada

Se observan diferencias entre las alturas de ambos diagramas, esto es el resultado de haber aplicado diferentes métodos en el cálculo de las distancias.

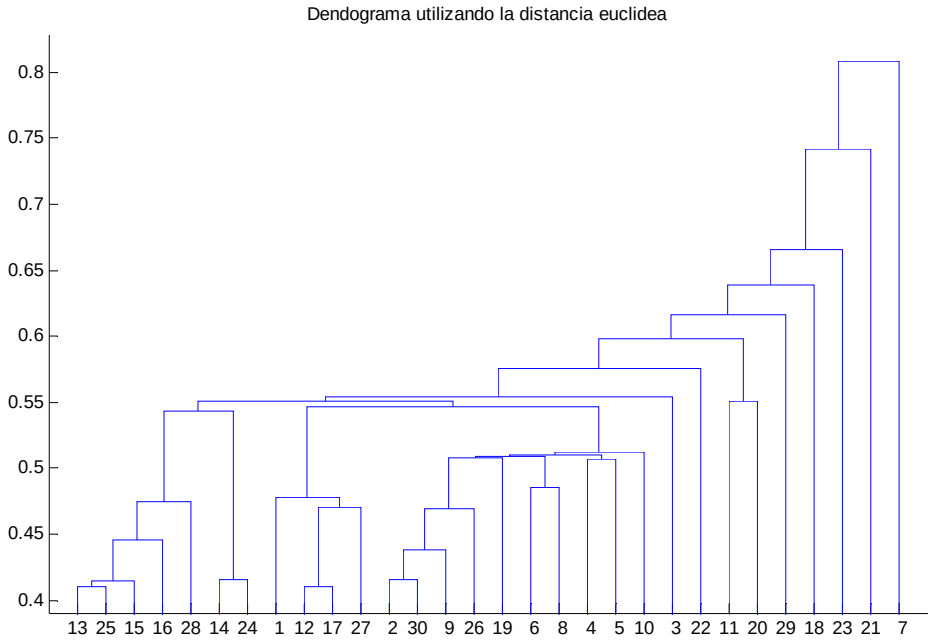

Ilustración 6.33 Dendograma a partir de la distancia sin promediar

Los datos que muestran las agrupaciones que se han realizado para cada uno de los documentos en cada una de las repeticiones de K-means, se van sumando en una matriz de concurrencias finales. Las coocurrencias muestran el número de veces que un documento se encuentra en el mismo grupo que el resto de los documentos. Estos datos se guardan en una matriz cuyas filas y columnas son los documentos, y las relaciones entre un documento y el resto de los documentos quedan definidas por un valor numérico, que en función de su magnitud indica si existe una gran relación entre dos documentos o no, como se puede ver en la siguiente ilustración:

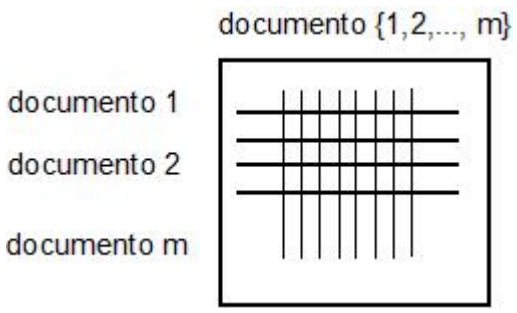

Ilustración 6.34 Matriz de coocurrencias

La gráfica siguiente muestra una imagen que representa la matriz coocurrencia con sus valores pintados más oscuros cuanto más elevados son, se puede comprobar que la diagonal principal tiene los valores más elevados de toda la matriz, debido a que contiene la relación entre un documento y él mismo, su valor numérico real coincide con el número de repeticiones del algoritmo k-means porque en cada ejecución del algoritmo, un documento siempre pertenecerá al mismo grupo que el mismo. También se observa simetría a partir de la diagonal principal porque relaciona a cada documento 2 veces, a modo de ejemplo con dos documentos de partida obtendríamos la relación repetida documento A con documento B y documento B con documento A.

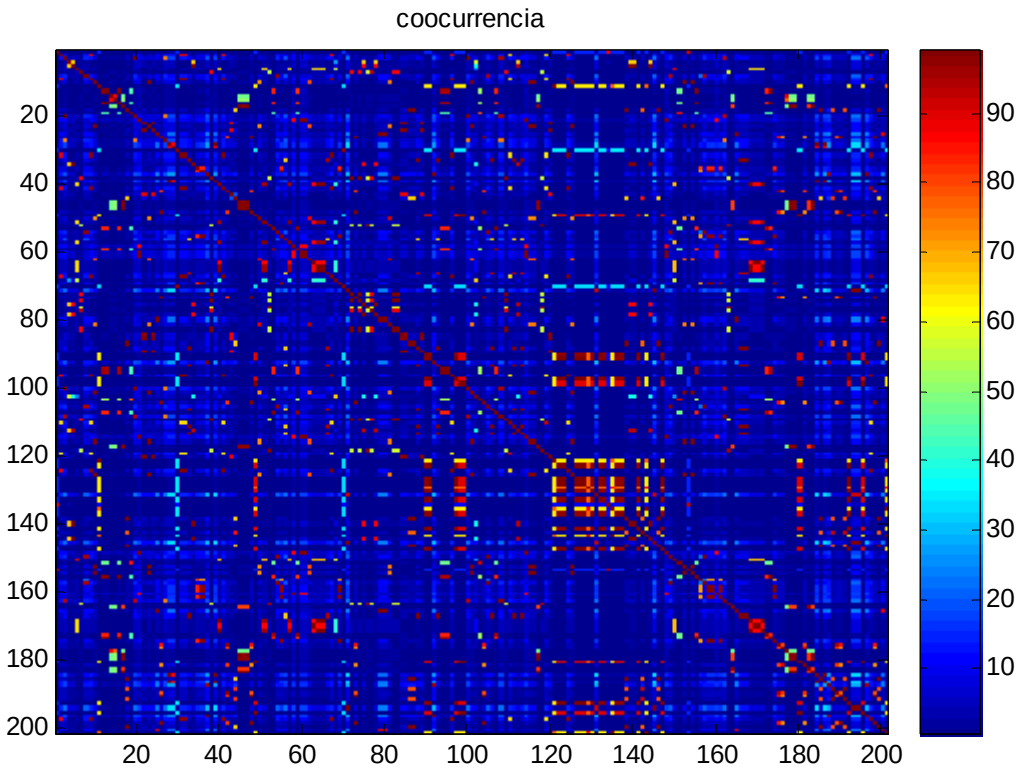

Ilustración 6.35 Matriz de coocurrencias

Existe otra manera de interpretar el contenido de la matriz de coocurrencias, cada una de las filas expresa la relación entre un documento concreto con el resto de los documentos, existiendo un vector de relaciones por cada uno de los documentos, pudiendo observar independientemente las relaciones de cada uno de los documentos, como en la siguiente gráfica.

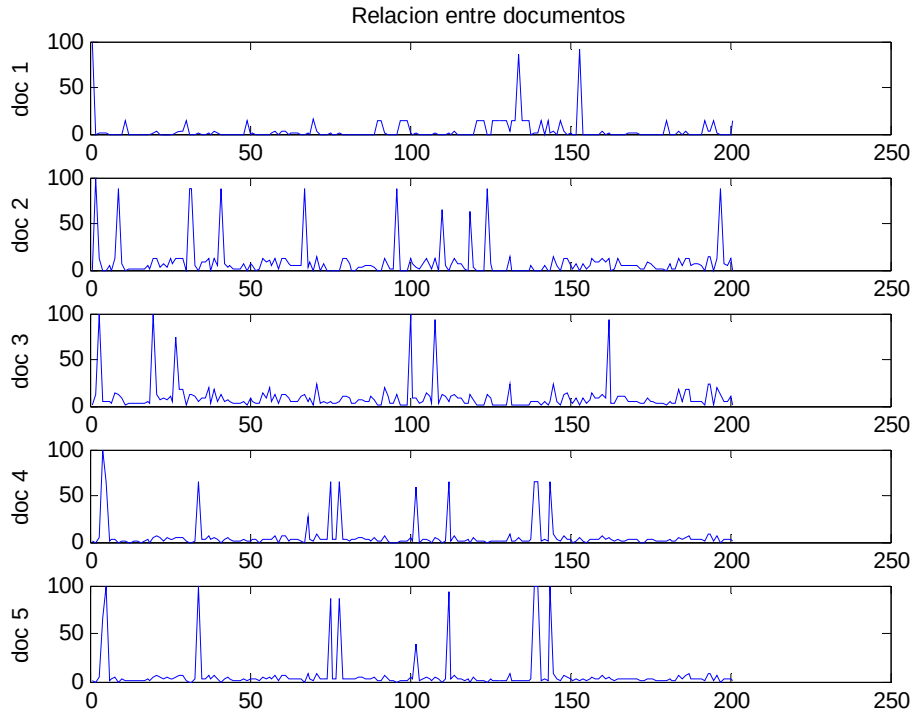

Ilustración 6.36 Relación entre documentos a partir de sus coocurrencias

En la siguiente gráfica se muestran solamente las relaciones existentes entre 22 documentos escritos por el mismo autor, esto se consigue aplicando un filtrado selectivo la matriz de concurrencias calculada a partir de los datos de todos los documentos, se compara el documento Sorzano1998.pdf con el resto de documentos del mismo autor.

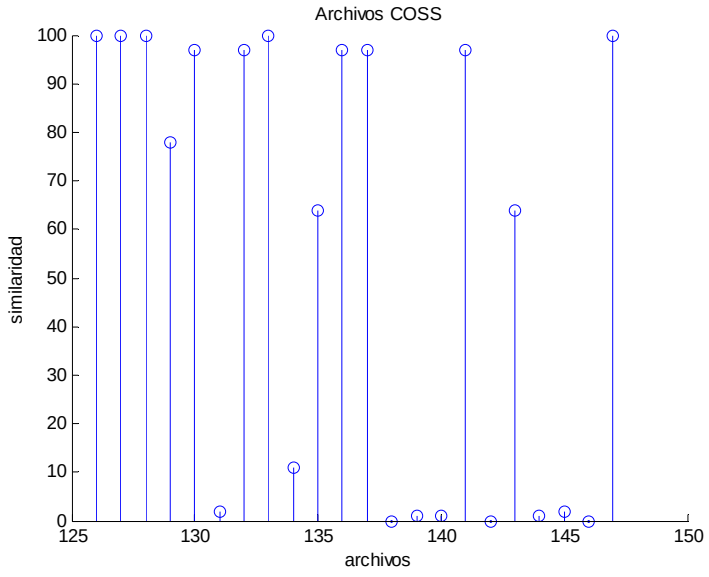

Ilustración 6.37 Relación entre documentos del mismo autor

Todos los documentos que se muestran en la siguiente tabla, van a tener mayor relación entre ellos mismos porque son del mismo autor, y se observa que la mayoría de los documentos tienen valores de coocurrencia altos. Para comprobar que el resultado de esta matriz de coocurrencias relaciona adecuadamente los documentos, podemos observar el título que tiene de cada uno de los artículos.

Sorzano1998.pdf <126>	Sorzano2003c.pdf <133>	Sorzano2004h.pdf <140>
Sorzano1999.pdf <127>	Sorzano2004.pdf <134>	Sorzano2005.pdf <141>
Sorzano1999b.pdf <128>	Sorzano2004b.pdf <135>	Sorzano2006.pdf <142>
Sorzano2001b.pdf <129>	Sorzano2004c.pdf <136>	Sorzano2006b.pdf <143>
Sorzano2002.pdf <130>	Sorzano2004d.pdf <137>	Sorzano2006c.pdf <144>
Sorzano2002d.pdf <131>	Sorzano2004f.pdf <138>	Sorzano2007a.pdf <145>
Sorzano2003b.pdf <132>	Sorzano2004g.pdf <139>	Sorzano2007b.pdf <146>
		Sorzano2007d.pdf <147>

Tabla 6.9 Documentos asociados a su número identificador

El primer archivo Sorzano1998.pdf tiene el siguiente título: *Effects of Uneven Sampling in 3D Reconstructions*.

Los archivos que más tienen que ver con este archivo según los valores de concurrencias que poseen son:

- Sorzano1999.pdf: *Quantitative comparison of 3D reconstruction algorithms under conditions of uneven angular distribution in electron microscopy*.
- Sorzano1999b.pdf: *Quantitative comparison of 3d reconstruction algorithms under conditions of uneven angular distribution in electron microscopy*.
- Sorzano2003b.pdf: *Image processing in biological 3d electron microscopy*.
- Sorzano2003c.pdf: *A multiresolution approach to orientation assignment in 3D electron microscopy of single particles*.

- Sorzano2004c.pdf: *Volumetric spectral signal-to-noise ratio*.
- Sorzano2004d.pdf: *XMIPP: a new generation of an open-source image processing package for electron microscopy*.
- Sorzano2005.pdf: *Multiobjective algorithm parameter optimization using multivariatestatistics in three-dimensional electron microscopy reconstruction*.
- Sorzano2007d.pdf: *Volumetric restrictions in single particle 3DEM reconstruction*.

Los archivos que menos tienen que ver con el primero según el valor de concurrencia que poseen son:

- Sorzano2002d.pdf *Command-line interfaces can be efficiently brought to graphics: COLIMATE (the Command Line Mate)*.
- Sorzano2004f.pdf: Note *on "Wavelets, Gaussian mixtures and Wiener filtering"*.
- Sorzano2004g.pdf: *Elastic Registration of Biological Images Using Vector-Spline Regularization*.
- Sorzano2004h.pdf: *Algorithm for spline-based elastic registration in application to confocal images of gene expression*.
- Sorzano2006.pdf: *Improved Bayesian image denoising based on wavelets with applications to electron microscopy*.
- Sorzano2006c.pdf: *Elastic Image Registration with Applications to Proteomics*.
- Sorzano2007a.pdf: *Fast, robust, and accurate determination of transmission electron microscopy contrast transfer function*.
- Sorzano2007b.pdf: *Mathematical models in image processing*.

Se puede comprobar que los documentos hablan de los mismos temas a la vez contienen unos valores de coocurrencia elevados, y los que tienen menos relación con el primer

documento aunque sean del mismo autor tienen valores de concurrencia más pequeños o incluso de cero.

6.6 HCA

La matriz de coocurrencias que se ha descrito anteriormente contiene valores de similitudes entre los documentos, no las distancias entre los documentos que es lo que necesitamos, por ello hay que convertir los valores de la matriz de coocurrencias que mostraban similaridad entre los documentos en valores que expresen distancias, también se puede llamar matriz de disimilaridad. Los valores que antes eran muy elevados porque eran documentos que se encontraban muchas veces en el mismo grupo que otro documento, ahora serán muy pequeños, porque los documentos que son muy parecidos ahora estarán muy próximos entre si, utilizaremos la siguiente fórmula para obtener un vector de distancias:

$$Dist(k) = 1 - coocurrencia(i, j) / \max(coocurrencia)$$

Se introduce la información en un vector en vez de en una matriz, este vector es capaz de contener la misma información que si fuese una matriz, porque sabemos que la matriz de concurrencias es simétrica y que la diagonal principal no aporta información debido a que relaciona a un documento con el mismo documento, por lo tanto se elimina la información redundante.

No es posible interpretar fácilmente el contenido del vector de distancias recién calculado para poder sacar algún tipo de conclusión, porque la posición que ocupa cada documento de este vector esta desplazada con respecto a la que ocupaba en formato de matriz, debido a que se han suprimido los elementos de la diagonal principal y todos los elementos que quedaban debajo de esta por ser simétricos, pero se sigue manteniendo la relación entre un documento y el resto de los documentos como se puede ver en la siguiente gráfica:

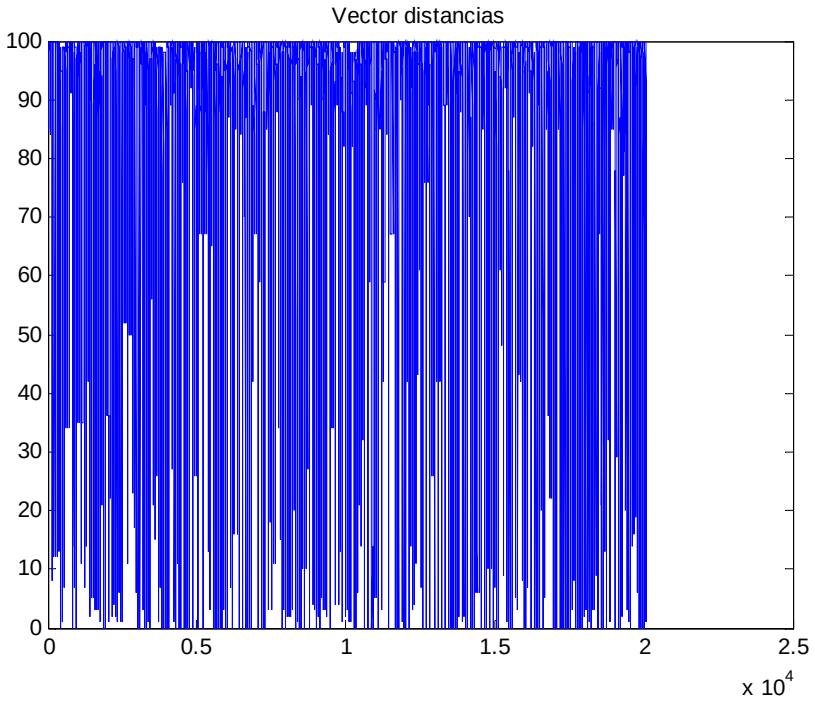

Ilustración 6.38 Representación vector distancias

Para poder entender el tipo de información que guarda este vector de manera gráfica hay que convertirlo en una matriz, pero sólo a modo de ejemplo, porque se trabajará con la información de distancias en formato de vector.

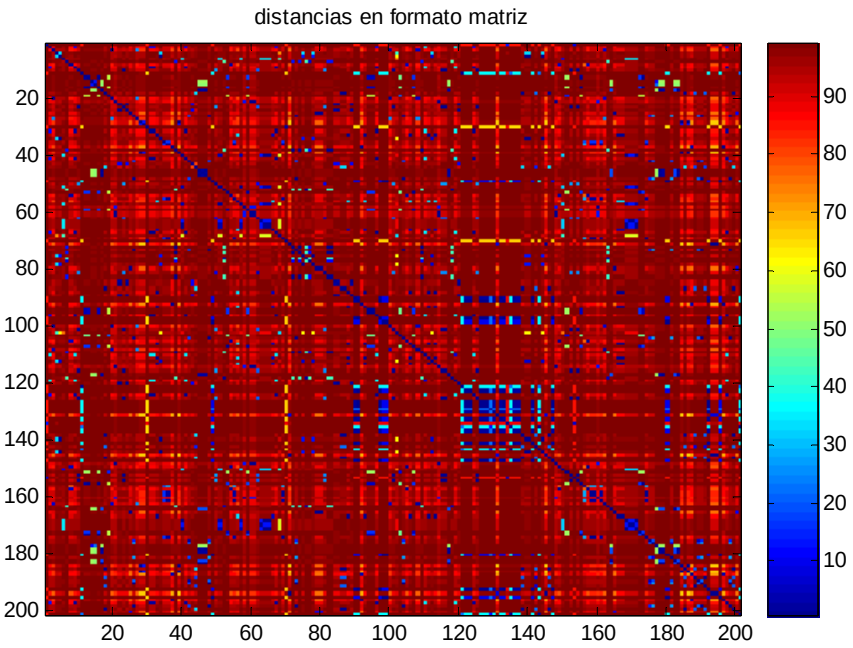

Ilustración 6.39 Matriz de distancias

Se calculan con el vector de distancias los grupos a los que pertenecerán los documentos con el algoritmo HCA, y los lugares que ocuparían los documentos para satisfacer esas distancias con el algoritmo MDS.

6.7 MDS

Se utiliza el algoritmo de escalado multidimensional para obtener un gráfico que contenga las relaciones entre los documentos reconstruidas a partir del vector de distancias, cuanto más cerca aparezcan los documentos mas parecido habrá entre ellos.

En este caso se hace una representación en 3 dimensiones, es decir cada documento tiene 3 componentes x,y,z.

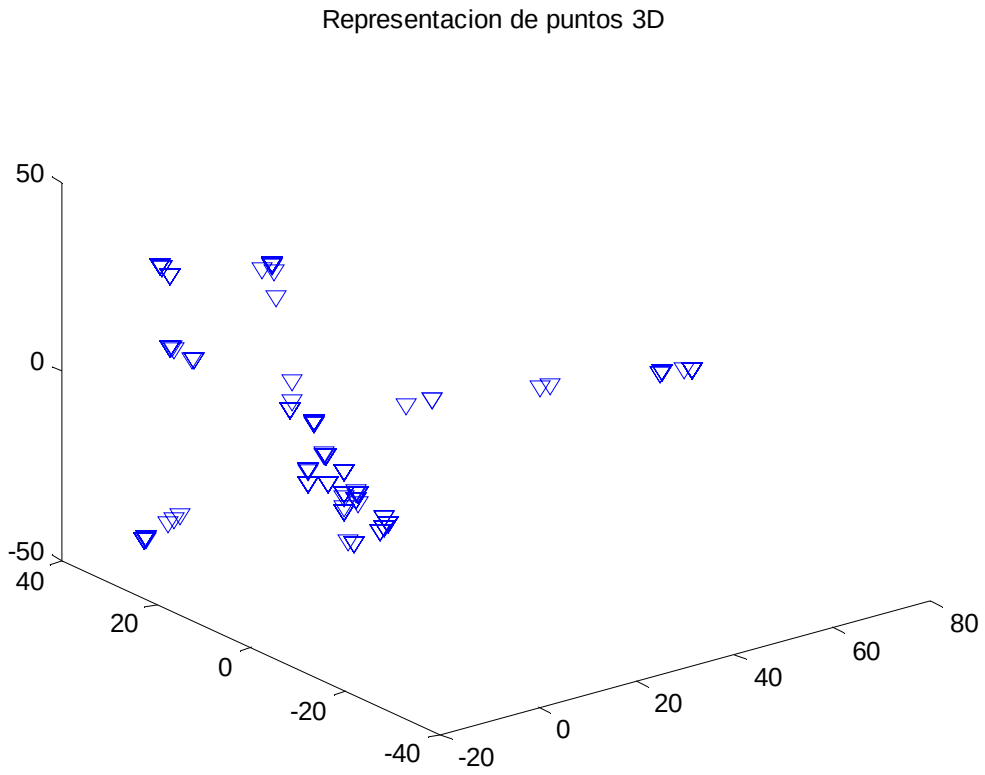

Ilustración 6.40 Representación de documentos en 3 dimensiones

Si proyectamos esas 3 dimensiones en 2 dimensiones, en función de la dirección en la que se proyecte, ocurrirá que puntos que aparecen distantes entre ellos pueden aparecer cercanos.

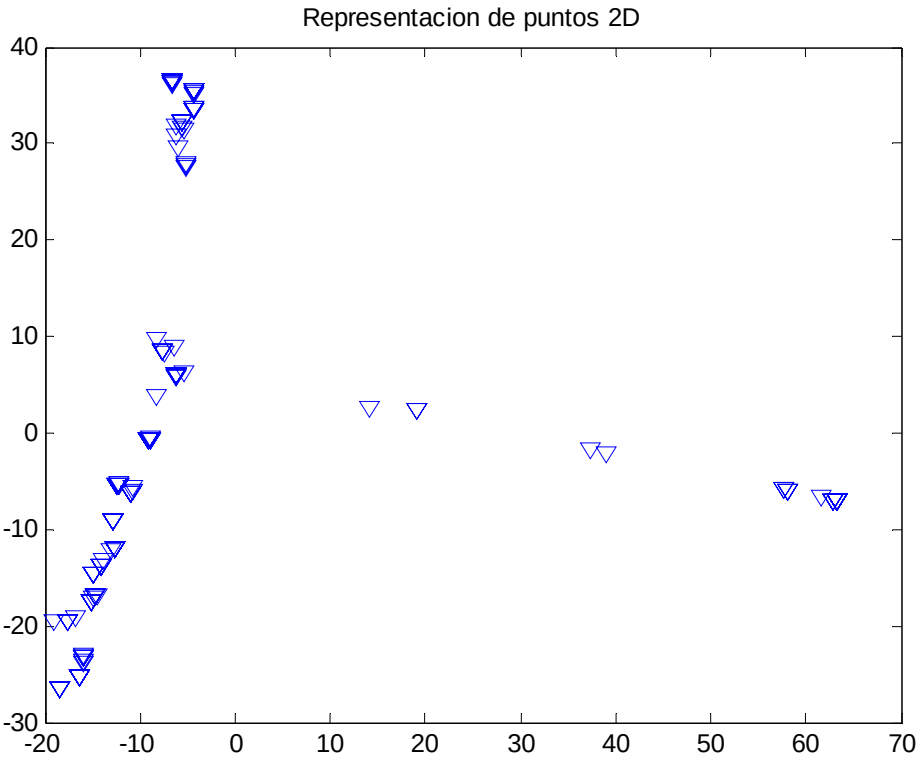

Ilustración 6.41 Representación de documentos en 2 dimensiones

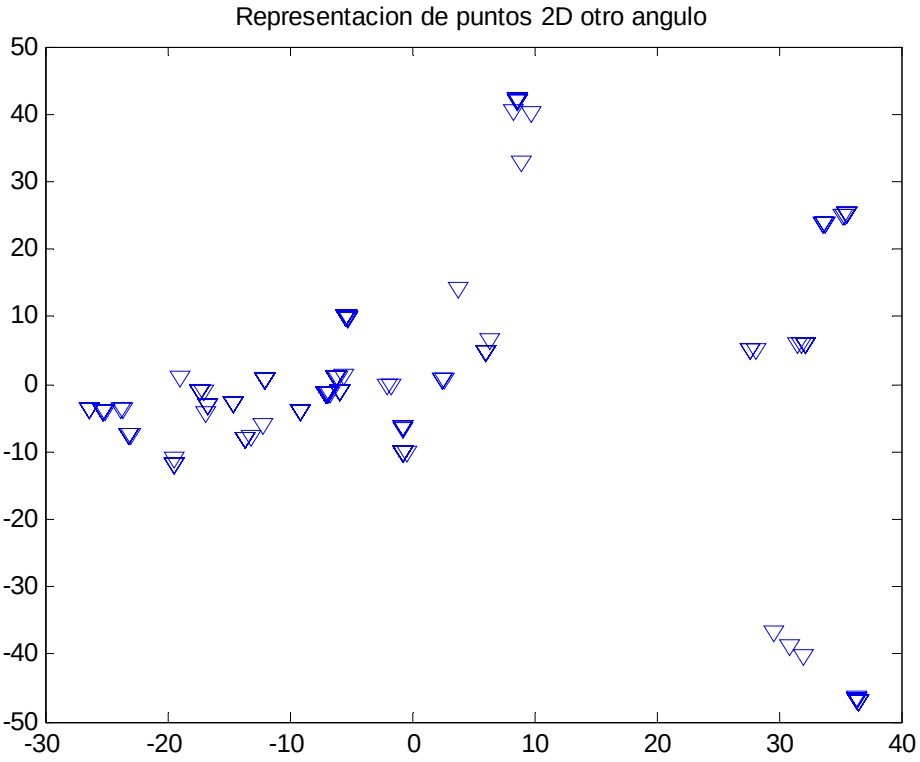

Ilustración 6.42 Representación de documentos en 2 dimensiones

Se muestran dos proyecciones en 2 dimensiones para los mismos con los que se realizó la gráfica de 3 dimensiones, y se observa que los puntos que representan los documentos, no aparecen en ambos casos en las mismas posiciones, esto se debe a que en los dos casos se ha utilizado un plano diferente para proyectar los datos, es la consecuencia de perder una dimensión, se pierde información.

6.8 Métodos de promediar los datos

6.8.1 Promediador de W con reconstrucción de H

Como hemos mencionado existen varias formas de promediar los datos intermedios, para eliminar la aleatoriedad y conseguir que los datos sean más fiables.

En el caso descrito a continuación a partir de la matriz V que contiene los pesos de las palabras de cada documento, se realizan 20 ó 30 repeticiones del algoritmo de factorización de matrices no negativas para reducir la dimensionalidad de los datos y posteriormente se ejecuta K-means con las matrices W que se obtienen de cada una de las repeticiones de NMF, de esta forma se obtienen tantas matrices de concurrencias como repeticiones del algoritmo NMF, que se promedian para tener una sola matriz de concurrencias.

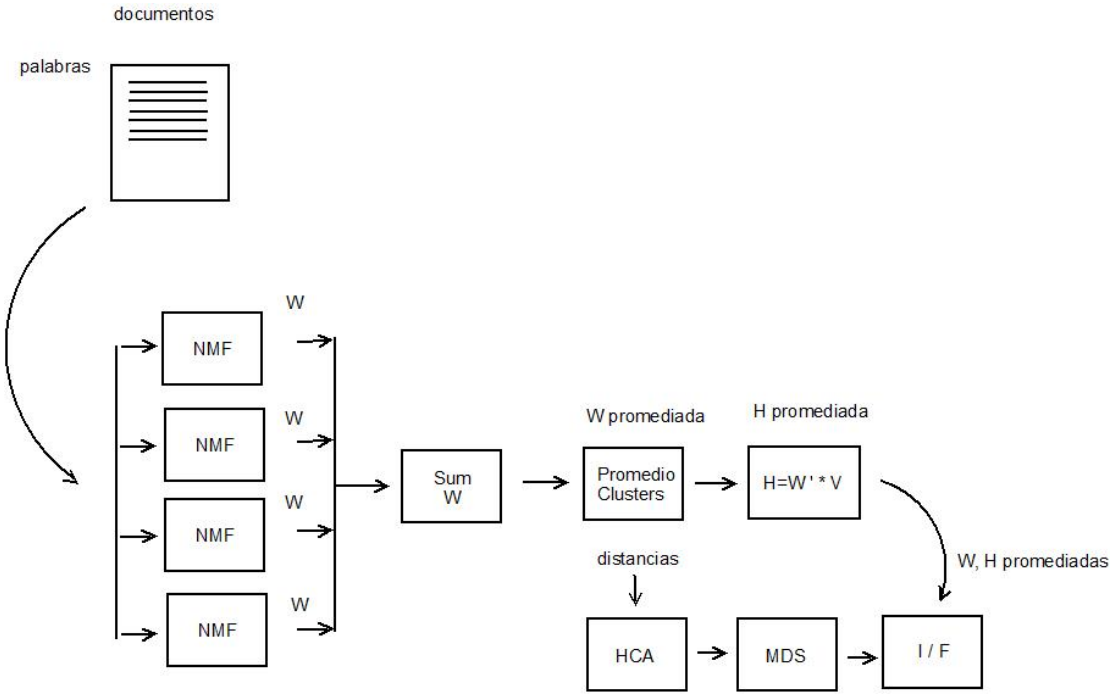

Ilustración 6.43 Promediador de W

Con cada una de las matrices W que se obtienen de la ejecución de NMF, se forma una sola matriz que contiene los resultados de todas las matrices intermedias que tendrá como dimensiones en sus filas el número de palabras multiplicado por el número de repeticiones de NMF, y en sus columnas los temas, la matriz obtenida se llamará matriz W suma, porque contiene a todas las matrices W resultantes de la repetición de la factorización de matrices no negativas.

Con estos datos se hace un promediado para hallar los temas de consenso, que se consiguen a partir de la función clasificador que se describe a continuación.

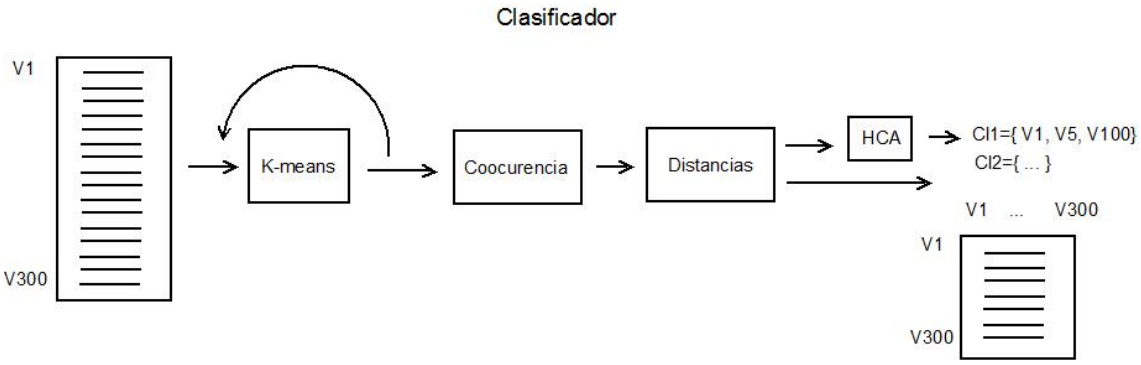

Ilustración 6.44 Clasificador

Esta matriz W suma que tiene unas dimensiones muy grandes, debido a que contiene a todas las matrices W resultantes de cada una de las repeticiones de NMF, se introduce como dato de entrada en el algoritmo K-means obteniendo una matriz de coocurrencia, a continuación se hallan las distancias relativas entre cada uno de los elementos que compone la matriz de concurrencia, los datos de las distancias se introducen en algoritmo HCA con el objetivo de agrupar los temas de la matriz W suma para generar una nueva matriz W promediada.

Para promediar estos datos que obtenemos del clasificador hay que ir seleccionando de la matriz W suma, los vectores en función de la clasificación que se ha devuelto del algoritmo HCA. Aquellos vectores de la matriz H suma que pertenecen al mismo grupo, en este caso el mismo tema, se van sumando sucesivamente en una matriz nueva que será la matriz H promediada, y al final se divide la suma total de los temas entre el número de veces que se ha sumado para obtener el promedio.

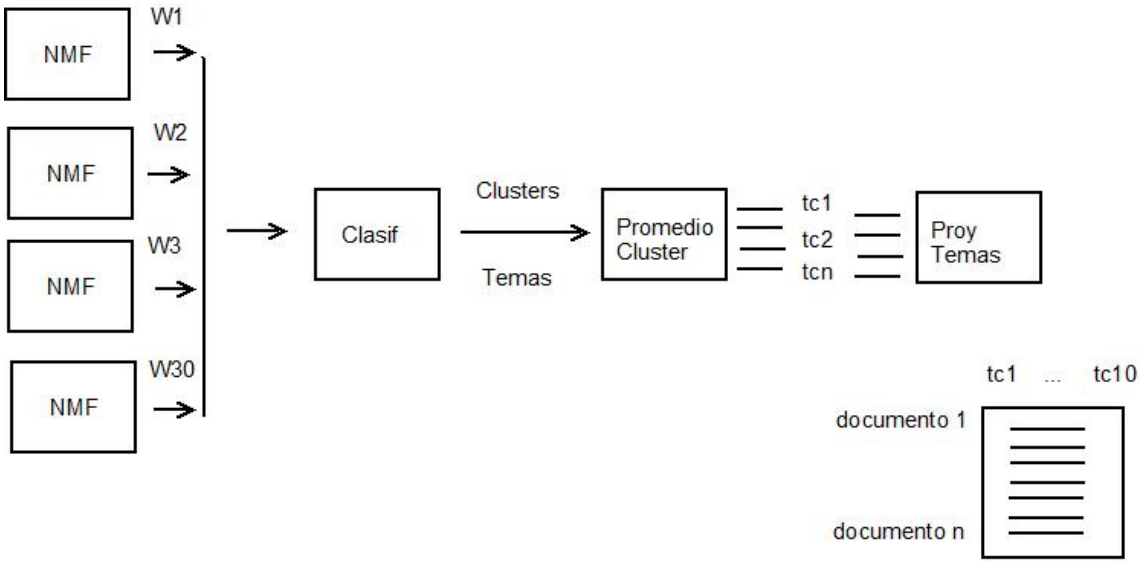

Ilustración 6.45 Recuperación de H

Para obtener la matriz H promediada, se aprovechan las propiedades de la matriz inversa, que permiten calcular esta matriz H conociendo previamente las matrices W' y V, a partir de la siguiente fórmula:

$$H=W'V;$$

Matriz Pseudo inversa

Para hallar la pseudo inversa, se ha utilizado el método de Moore-Penrose, este método está implementado en la librería Jama que es compatible con Java, pero tras incorporar la librería al proyecto y hacerla funcionar con nuestros datos, no era capaz de calcular la matriz pseudo-inversa de la matriz W, debido a que el rango de la matriz W es muy bajo.

Se intentó calcular la matriz pseudo-inversa de la siguiente forma:

$$pinv = A^T \left(A * A^T \right)^{-1}$$

Pero existía el mismo problema, si el rango de W es muy pequeño, no se puede calcular la matriz pseudo-inversa, por lo tanto tampoco es viable el uso de este método debido a la naturaleza de nuestros datos.

Se decidió implementar una función propia que calculase la matriz pseudo-inversa basándose en una descomposición de valores singulares (SVD):

Dada una matriz $A \in \mathbb{R}^{mxn}$, entonces existen 2 matrices $U \in \mathbb{R}^{mxm}$ y $V \in \mathbb{R}^{nxn}$ tal que es posible descomponer la matriz A de la siguiente forma:

$$A = U \sum V^T$$

siendo $\sum$ una matriz m x n, diagonal de la siguiente forma:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sigma_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_p & 0 \end{bmatrix}$$

$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_p \geq 0$ siendo $p = \min\{m, n\}$ y σ_i los términos de aplicar SVD o *singular value decomposition* de la matriz A. Las columnas de U son denominados vectores singulares izquierdos y las columnas de V los denominados términos vectores singulares derechos.

Para hacerse una idea geométrica de lo que son los SVD de una matriz, son los semiejes de la elipsoide que define la operación:

$$E = \left\{ \vec{z} \mid \vec{z} = A\vec{x}; \left\| \vec{x} \right\| = 1 \right\}$$

El cálculo de la matriz pseudo-inversa se apoya en el cálculo SVD y se puede conseguir de forma fácil, utilizando la siguiente fórmula:

$$pinv(A) = V \sum^{-1} U^T$$

siendo $\sum^{-1}$ de la siguiente forma:

$$\Sigma = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_3} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{\sigma_p} & 0 \end{bmatrix}$$

En este caso solo obtendremos un cero en la diagonal principal sólo si alguno de los σ_i son cero.

Con este método se consigue calcular la matriz pseudo-inversa, y por tanto se puede reconstruir la matriz H promediada.

6.8.2 Promediado basado en suma de distancias

Uno de los métodos más sencillos que se puede implementar para conseguir promediar los datos es el que se describe a continuación.

Se ejecutan tantas repeticiones del algoritmo NMF como se desee a partir de la matriz V como datos de entrada, en cada una de estas repeticiones obtenemos las matrices H y W.

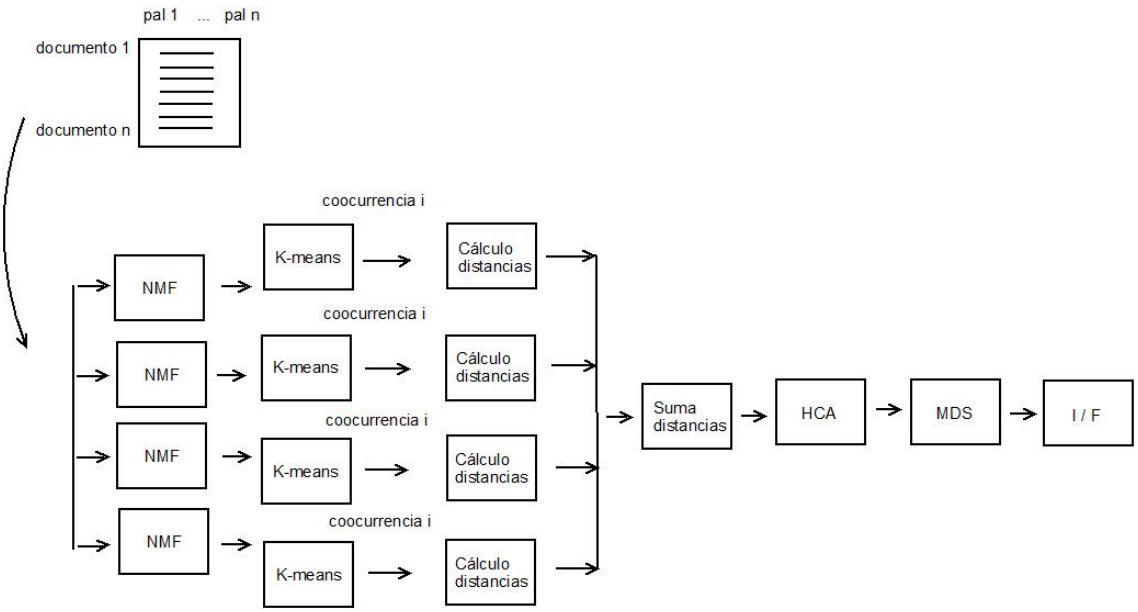

Ilustración 6.46 Promediador basado en suma de distancias

Se normalizan los datos cada una de las matrices H y se introducen como parámetro en el algoritmo K-means, a su salida se obtiene una matriz de concurrencias para cada una de las ejecuciones de Kmeans y se calcula el vector de distancias asociado a cada una de las matrices de concurrencias.

Para promediar todas estas distancias, se suman los vectores distancias en un solo vector para obtener un vector distancias más fiable.

Posteriormente el algoritmo HCA calcula los agrupamientos finales aprovechando el vector de distancias anterior.

Para obtener una representación de los documentos, se introduce el vector de distancias promediado como parámetro en el algoritmo de escalado multidimensional para obtener los puntos que representarán cada documento.

6.8.3 Promediado de coocurrencias

Se implementó otra manera de promediar los datos, generando una matriz de coocurrencia utilizando para ello las coocurrencias parciales.

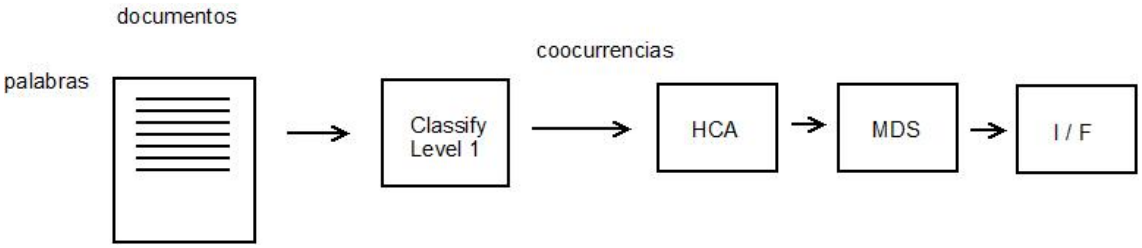

Ilustración 6.47 Promediado de coocurrencias

Los datos de entrada corresponden a la matriz V, que se compone de los pesos de los documentos y las palabras que los forman, estos datos van a ser procesados con el objetivo de obtener las coocurrencias promediadas.

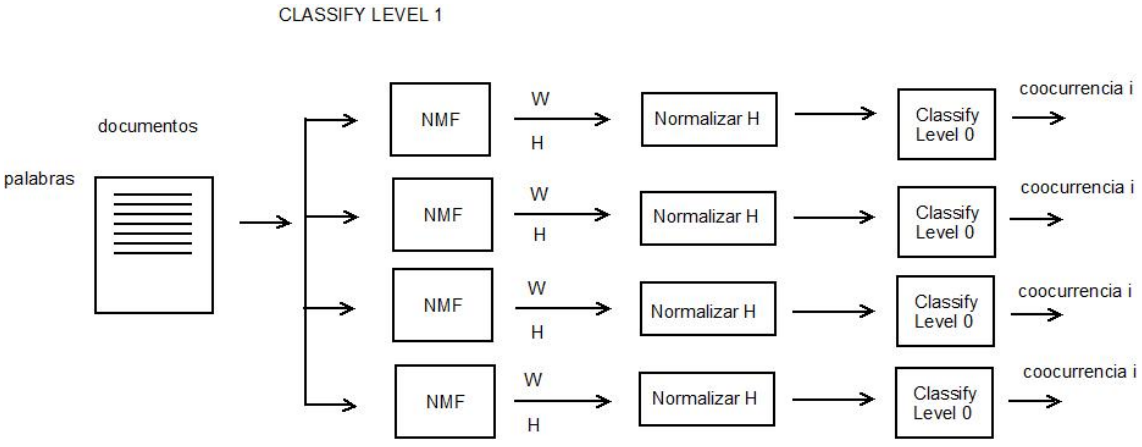

Ilustración 6.48 Clasificador de primer nivel

Una vez que hemos obtenido tantas matrices H normalizadas como ejecuciones de NMF se hayan realizado, se aplica K-means a cada una de estas matrices H normalizadas para

agrupar los documentos. A partir de los vectores que agrupan los documentos se calcula la coocurrencia para cada una de las matrices H, es decir no promediada.

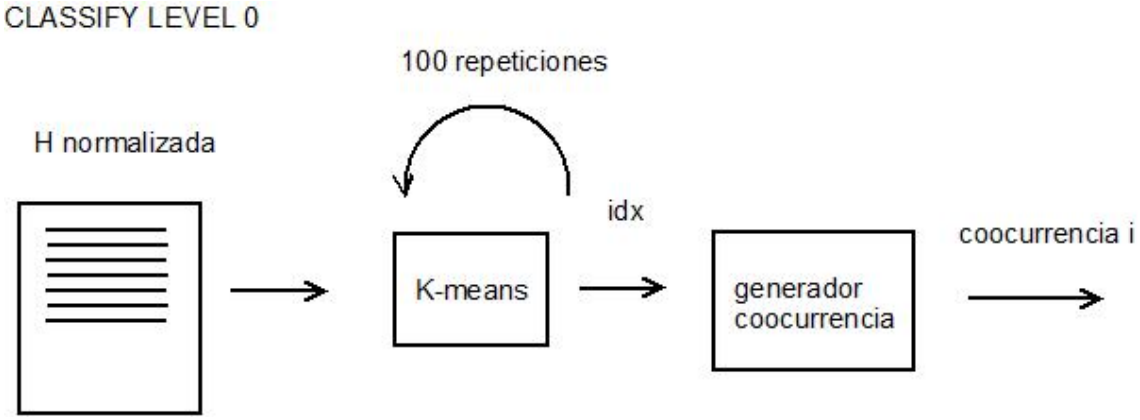

Ilustración 6.49 Clasificador base

Para obtener la concurrencia promediada, se suman los datos de las matrices de concurrencias parciales en una sola matriz que contendrá las relaciones entre los documentos. A partir de esta matriz se aplica clustering jerárquico para obtener los grupos a los que pertenecen los documentos y multidimensional scaling, para representar los documentos gráficamente en 2 dimensiones.

6.8.4 Promediado de distancias reales

Los datos de partida son los que componen la matriz V, y se ejecutan 20 ó 30 repeticiones del algoritmo NMF donde se obtienen las matrices W y H. Directamente sobre la matriz H, se calcula la distancia euclídea entre los vectores que la componen, obteniendo una matriz que contiene las distancias entre cada documento y el resto de documentos.

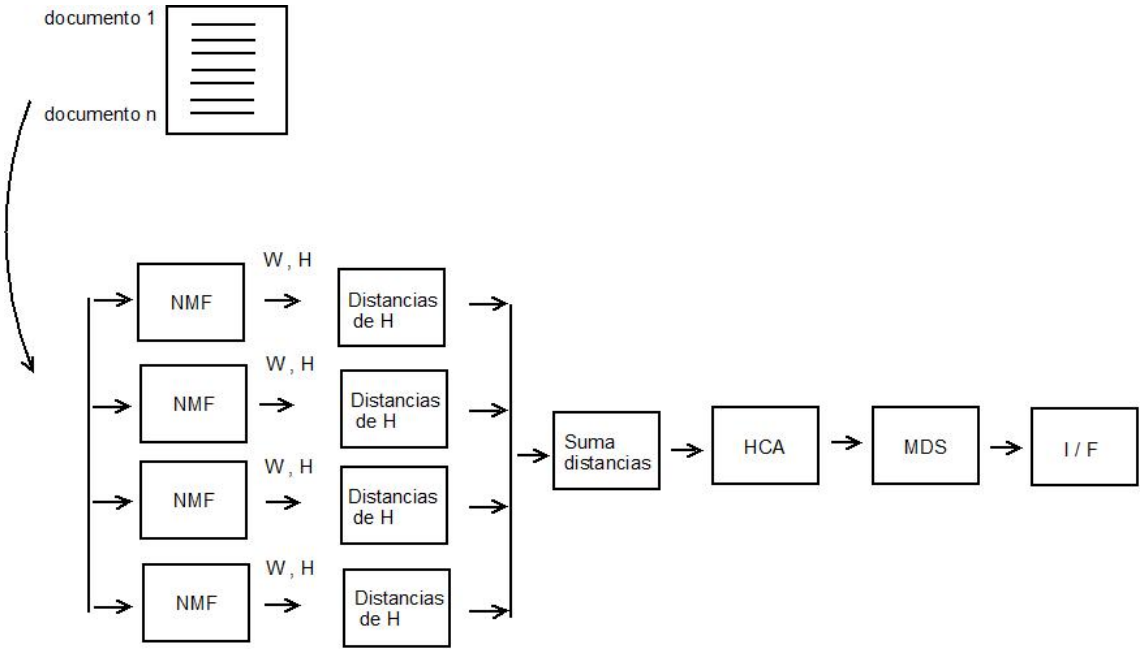

Ilustración 6.50 Promediador de distancias reales

La diferencia con los otros métodos consiste en que las distancias no se calculan a partir de las coocurrencias del algoritmo Kmeans, lo que tiene como ventaja que siempre se obtendrán las mismas distancias para los mismos datos de entrada, cosa que no ocurría con el cálculo de las distancias cuando se utilizaba Kmeans, debido a la aleatoriedad que introduce este algoritmo. Otra ventaja consiste en que las distancias son distancias reales entre puntos, en cambio al utilizar Kmeans las distancias se obtienen a partir de las concurrencias consistiendo en distancias relativas.

A partir de las distancias absolutas que se han obtenido a partir de las matrices H de cada una de las repeticiones de NMF, se obtiene un vector de distancias promediadas a partir de su suma.

Los grupos finales de los documentos, se forman al ejecutar HCA que aprovecha estas distancias ya promediadas.

Con este método se obtienen resultados de grupos bastante buenos, donde se puede ver que los grupos que se forman a la salida son de documentos que hablan de los mismos temas.

6.9 Comparación entre diferentes métodos

6.9.1 Introducción

Cada uno de los métodos expuestos anteriormente para promediar los datos, poseen ventajas y limitaciones que son consecuencia de la forma que se utiliza para conseguir promediar esos datos, se expone a continuación una comparación entre los 4 métodos para promediar los datos con el objetivo de evaluar los puntos fuertes de cada uno de ellos y escoger el mejor método para la resolución del problema propuesto.

Todos los métodos tienen como requisitos de salida 25 agrupaciones, 20 temas en la factorización de matrices no negativas, y 22 documentos como mínimo número de documentos en el que debe de aparecer una palabra para ser considerada como importante.

6.9.2 Promadiador de W con reconstrucción de H

Para este método de promediado hay que mostrar la matriz W promediada, ya que no se calcula como en otras ocasiones a partir de la salida del algoritmo de factorización de matrices no negativas, sino que se forma a partir de realizar búsquedas en la matriz nmfW promediada suma, que es la agrupación de todas las matrices W que se han generado, debido a las repeticiones del algoritmo de factorización de matrices no negativas, en este caso con 10 repeticiones, lo que significa que esta matriz agrupación de matrices W, tiene dimensiones 200 x 1658.

Se muestran a continuación 5 temas de esa matriz W promediada, que se compone de un total de 20 temas.

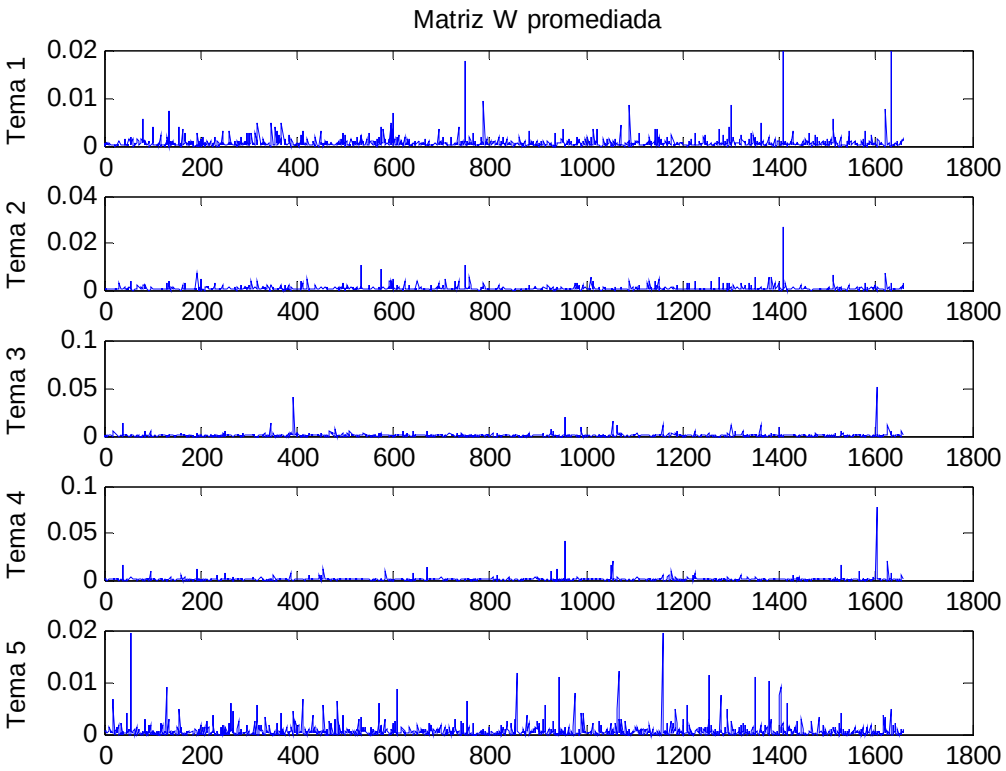

Ilustración 6.51 Matriz W promediada

Con los datos de la matriz W, se calcula la matriz H a partir de la siguiente expresión:

$$H=W^{+}V$$

La matriz pseudo-inversa se calcula con el método de Moore-Penrose, ya que se comprobó que era el único que soportaba matrices con autovalores tan pequeños.

Se puede comprobar que la matriz W sigue el mismo comportamiento que el resto de matrices W calculadas directamente a partir de la factorización de matrices no negativas, alcanzando sus valores máximos en puntos cercanos entre sí.

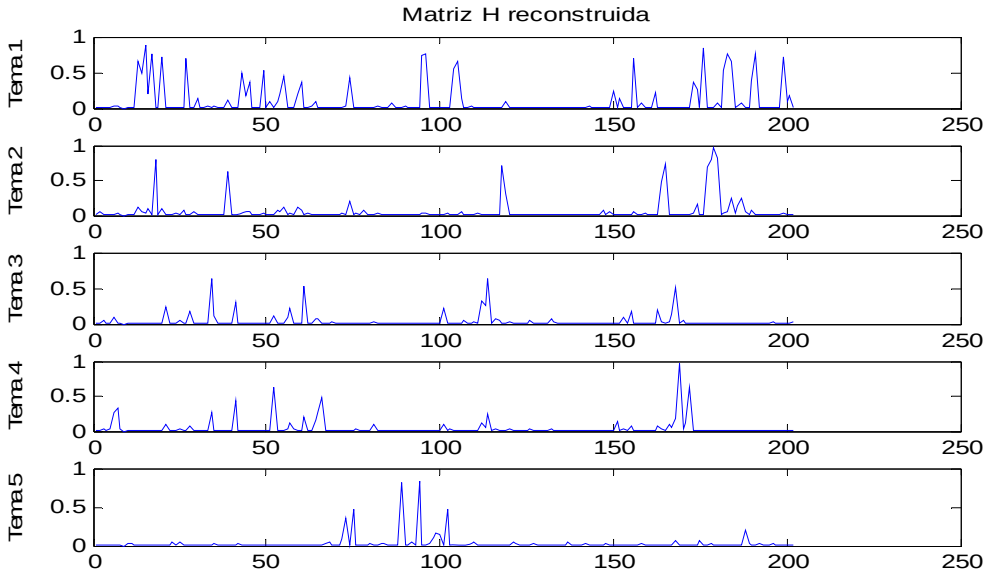

Ilustración 6.52 Matriz H reconstruida

Se puede generar la matriz de distancias a partir del vector de distancias obtenido a la salida del algoritmo Kmeans, donde se puede comprobar que tiene el mismo formato y las mismas propiedades que en los casos expuestos en la parte teórica. La diagonal principal se compone de valores muy pequeños porque un documento se encuentra muy próximo a él mismo, además la matriz posee simetría a partir de la diagonal principal.

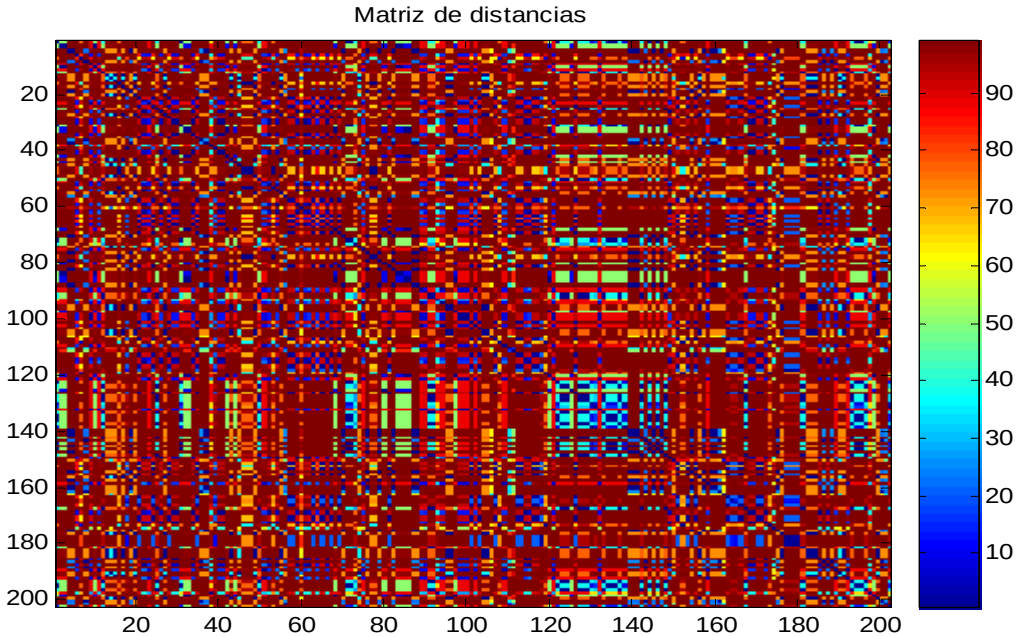

Ilustración 6.53 Matriz de distancias

De los grupos de documentos que se forman a la salida se obtienen grupos muy específicos como el siguiente en el que se agrupan documentos que tienen que ver entre sí:

Cluster 6
Sorzano2001b.pdf
Sorzano2002.pdf
Sorzano2004d.pdf
Zubelli2003.pdf

Tabla 6.10 Documentos pertenecientes al cluster número 6

Se muestran a continuación los temas de los que tratan los archivos pertenecientes para al cluster seleccionado a modo de ejemplo de ejemplo.

Sorzano2001b.pdf: *Transfer Function Restoration in 3D Electron Microscopy via Iterative Data Refinement*.

Sorzano2002.pdf: *New reconstruction conditions greatly improve the recontruction quality*.

Sorzano2004d.pdf: *XMIPP: a new generation of an open-source image processing package for electron microscopy*.

Zubelli2003.pdf: *Three-dimensional reconstruction by Chahine's method from electron microscopic projections corrupted by instrumental aberrations*.

Se muestra otro grupo de ejemplo, en el que se puede ver que se han seleccionado autores que hablan de los mismos temas.

Cluster 2
ramani0602.pdf
unser0503.pdf
unser0504.pdf
unser0506.pdf
unser0701.pdf

Tabla 6.11 Documentos pertenecientes al cluster 2

ramani0602.pdf: *Matérn B-Splines and the Optimal Reconstruction of Signals*.

unser0503.pdf: *Cardinal Exponential Splines: Part I—Theory and Filtering Algorithms*.

unser0504.pdf: *Cardinal Exponential Splines: Part II—Think Analog, Act Digital*.

unser0506.pdf: *Generalized Smoothing Splines and the Optimal Discretization of the Wiener Filter*.

unser0701.pdf: *Self-Similarity: Part I—Splines and Operators*.

Las desventajas de este método tienen que ver con los grupos que genera que no son muy homogéneos en cuanto al número de elementos, es decir que hay grupos con muchos elementos y grupos con muy pocos elementos, por lo que las agrupaciones se pueden mejorar, esto es debido a que el método no es capaz de encontrar más diferencias en los documentos de los grupos grandes para incluirlos en grupos con menos elementos para conseguir tener grupos más homogéneos.

6.9.3 Promediador basado en suma de distancias

El método promediador basado en suma de distancias se basa fundamentalmente en generar un vector de datos de distancias entre documentos, a partir de las distancias obtenidas al repetir el algoritmo de Kmeans, por lo tanto los valores de distancias serán más elevados que en el resto de los métodos, debido a que se suman todos las distancias, como se puede ver en la siguiente matriz de distancias generada a partir del vector de distancias.

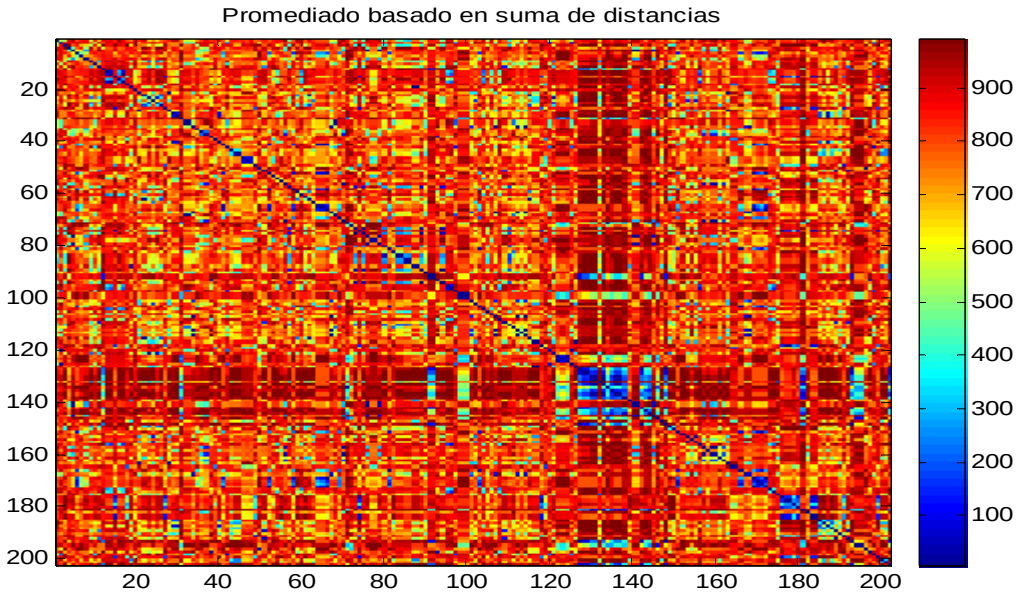

Ilustración 6.54 Matriz de distancias promediadas

Se sigue manteniendo la estructura de los demás casos en la que la diagonal principal se compone de ceros y existe simetría a partir de la diagonal principal.

Se puede comprobar la diferencia entre los histogramas de las distancias con un método que no realice la suma de los valores de distancias, como por ejemplo con el método promediador de W con reconstrucción de H, en este caso los valores de distancias más grandes del histograma serán mucho más pequeños que en el caso de promediado mediante suma de distancias. También hay que darse cuenta que en el caso del promediado mediante suma de distancias los valores altos están más escalonados, es decir menos concentrados en un rango más pequeño de valores, esto es porque los valores tienen más espacio hábil para estar colocados.

Los valores realmente importantes del vector de distancias son los más pequeños, porque indica proximidad entre documentos, y por lo tanto una mayor probabilidad para formar una agrupación.

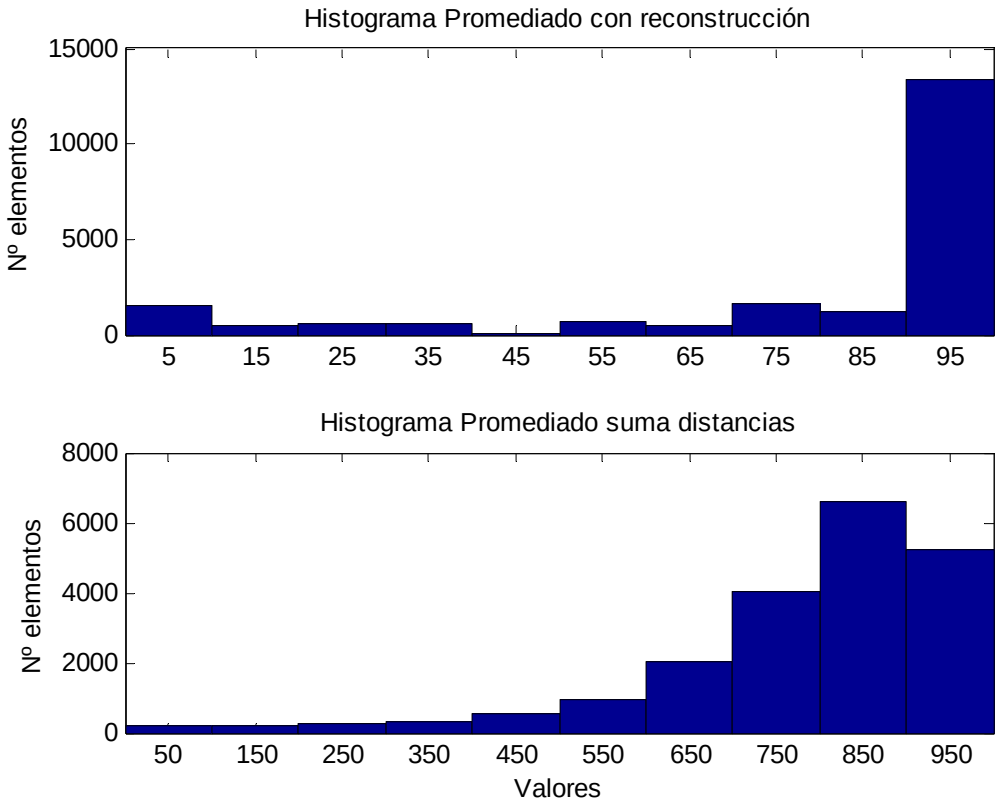

Ilustración 6.55 Comparación de histogramas a partir de matriz de distancias

En cuanto a la clasificación final las agrupaciones de documentos son algo más homogéneas con respecto al caso anterior, y se generan menos grupos con muy pocos elementos. Se muestra un grupo de ejemplo con los documentos que lo componen.

Cluster 22
blu0201.pdf
chaudhury0801p.pdf
forster0701p.pdf
khalidov0601.pdf
ramani0602.pdf
unser0301.pdf
unser0503.pdf
unser0504.pdf
unser0506.pdf
unser0701.pdf
unser9805.pdf
unser9901.pdf
vandeville0503.pdf
vonesch0702.pdf

Tabla 6.12 Documentos pertenecientes al cluster 22

Se lista a continuación el título de cada uno de los documentos pertenecientes al grupo, en este caso hablan de temas matemáticos, y mas concretamente de transformadas.

blu0201.pdf: *Wavelets, Fractals, and Radial Basis Functions*.

chaudhury0801p.pdf: *Construction of hilbert transform pairs ofwavelet bases and optimal time-frequency localization*.

forster0701p.pdf: *Shift-invariant spaces from rotation-covariant functions*.

khalidov0601.pdf: *From Differential Equations to the Construction of New Wavelet-Like Bases*.

ramani0602.pdf: *Matérn B-Splines and the Optimal Reconstruction of Signals*.

unser0301.pdf: *Wavelet Theory Demystified*.

unser0503.pdf: *Cardinal Exponential Splines: Part I—Theory and Filtering Algorithms*.

unser0504.pdf: *Cardinal Exponential Splines: Part II—Think Analog, Act Digital*.

unser0506.pdf: *Generalized Smoothing Splines and the Optimal Discretization of the Wiener Filter*.

unser0701.pdf: *Self-Similarity: Part I—Splines and Operators*.

unser9805.pdf: *Spline wavelets with fractional order of approximation*.

unser9901.pdf: *Fractional Splines and Wavelets*.

vandeville0503.pdf: *Isotropic Polyharmonic B-Splines: Scaling Functions and Wavelets*.

vonesch0702.pdf: *Generalized Daubechies Wavelet Families*.

Se puede comprobar como hay muchos documentos que son del mismo autor, aunque también aparecen otros autores, que hablan de temas muy similares entre sí, de donde se deduce que el método tiene resultados aceptables.

6.9.4 Promediador basado en distancias reales

Este método se diferencia del resto de los métodos por generar las distancias a partir de la matriz H que se obtuvo a la salida del algoritmo de factorización de matrices no negativas.

La distancia euclídea entre pares de puntos se calcula de la siguiente forma:

$$d_{rs}^2 = (x_r - x_s)(x_r - x_s)'$$

Es importante comparar la gráfica que se obtiene ordenando el vector de distancias del método de suma de distancias, que calculaba las distancias a partir de las coocurrencias con un método que calcula las distancias reales entre pares de puntos, en ambos métodos se sigue una gráfica muy parecida, exceptuando en las magnitudes debido a que en un caso se suman distancias, y en el otro se calculan directamente a partir de la fórmula descrita anteriormente.

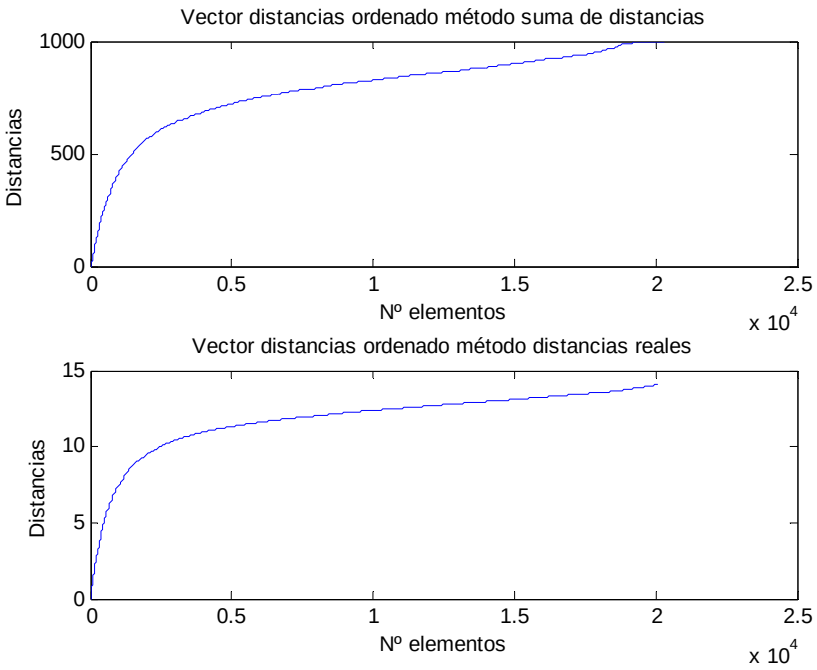

Ilustración 6.56 Comparación entre vectores de distancias ordenados

Los grupos que se obtienen gracias a este método son bastante acertados, en este caso se escoge un grupo que contiene documentos muy especializados en el tratamiento de imágenes de tipo médico, en este grupo incluso se reconoce a dos documentos que poseen un título idéntico, aunque el resto del documento no es exactamente igual.

Cluster 12
aguet0401.pdf
forster0403.pdf
forster0404.pdf
khalidov0502.pdf
khalidov0702.pdf
langoju0401.pdf
leitgeb0701.pdf
leitgeb0801.pdf
liebling0401.pdf
liebling0403.pdf
mallabiabarrena0501.pdf
patil0501.pdf
vandeville0404.pdf

Tabla 6.13 Documentos pertenecientes al cluster 12

Se muestran a continuación los temas de los documentos pertenecientes al grupo número 12, para poder comprobar que hablan de temas similares.

aguet0401.pdf: *On the fundamental limits of nano-particle tracking along the optical axis*.

forster0403.pdf: *Extended depth-of-field for color images in light microscopy: image fusion and 3d visualization*.

forster0404.pdf: *Complex Wavelets for Extended Depth-of-Field: A New Method for the Fusion of Multichannel Microscopy Images*.

khalidov0502.pdf: *Magnetic resonance spectroscopy imaging with field inhomogeneity compensation*.

khalidov0702.pdf: BSLIM: Spectral Localization by Imaging With Explicit B0 Field Inhomogeneity Compensation

langoju0401.pdf: *Resolution enhancement in optical coherence tomography*.

leitgeb0701.pdf: *Complex ambiguity-free Fourier domain optical coherence tomography through transverse scanning*.

leitgeb0801.pdf: *Complex ambiguity-free Fourier domain optical coherence tomography through transverse scanning*.

liebling0401.pdf: *Complex-wave retrieval from a single off-axis hologram*.

liebling0403.pdf: *Autofocus for digital Fresnel holograms by use of a Fresnelet-sparsity criterion*.

mallabiabarrena0501.pdf: *From medical images to numerical blood flow simulations in human vessels*.

patil0501.pdf: *High-resolution frequency estimation technique for recovering phase distribution in interferometers*.

vandeville0404.pdf: *Image Scrambling Without Bandwidth Expansion*.

Se puede comprobar tras leer el título de los documentos, que todos los documentos de este grupo tienen relación entre sí. Este método tiene unos resultados bastante acertados

en cuanto a los grupos que se generan, se consigue eliminar la aleatoriedad del algoritmo Kmeans al calcular las distancias directamente sobre la matriz H.

6.9.5 Promediado de coocurrencias

El método de promediado de coocurrencias se encarga de generar una matriz de coocurrencias a partir de las matrices de coocurrencias parciales de las repeticiones obtenidas a la salida del algoritmo Kmeans.

Se puede ver en la siguiente imagen una representación de la magnitud de los valores de la matriz de coocurrencias promediada. La mayoría de los valores de esta matriz son muy pequeños, esto se debe a que la relación que existe entre los documentos que poseen una coocurrencia pequeña es bastante escasa, solamente será útiles para la clasificación aquellos valores que posean una coocurrencia elevada, ya que implica que esos documentos habrían pertenecido en muchas ocasiones a los mismos grupos.

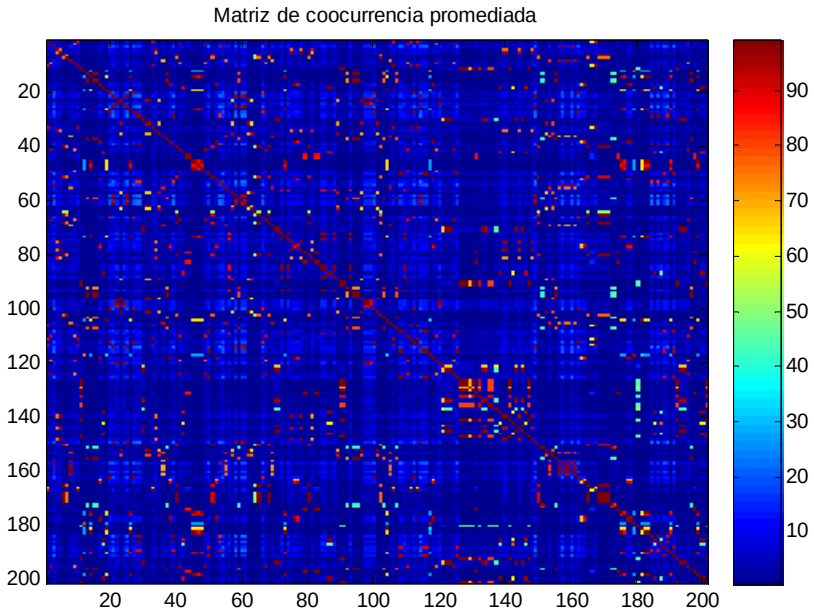

Ilustración 6.57 Matriz de coocurrencias promediada

Se seleccionan los valores de la matriz de coocurrencias de este método y se colocan en un vector para poder ordenar estos valores de manera ascendente, se hace los mismo con los valores de la matriz de coocurrencias del método de promediado mediante la

suma de distancias, ordenando sus valores de manera ascendente en un vector, se va a comparar el resultado de estos vectores en la siguiente gráfica.

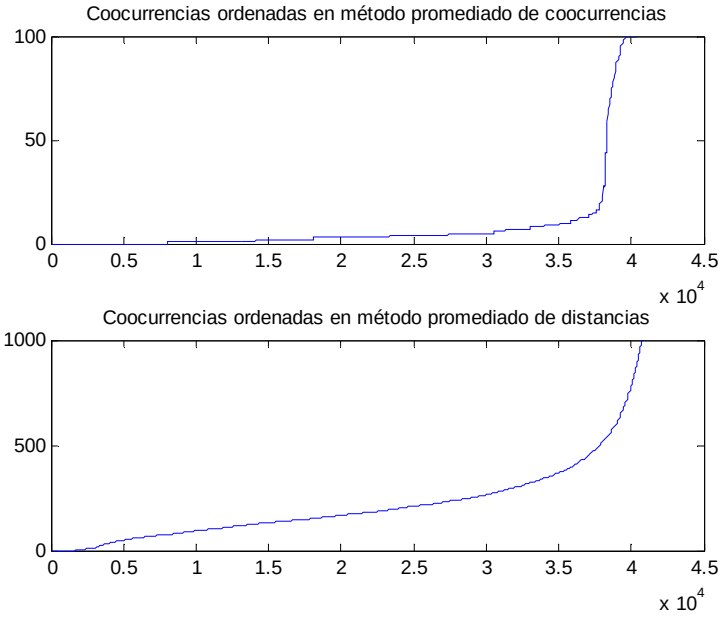

Ilustración 6.58 Comparación entre vectores de coocurrencias ordenados

Se puede comprobar que aunque se hayan calculado las coocurrencias mediante dos métodos diferentes, los datos de coocurrencias en ambas gráficas siguen un comportamiento similar en ambos casos aunque aparentemente ambas variables están relacionadas de forma no lineal.

Se escogen algunos documentos que se han agrupado de manera automática en los que se habla del mismo autor, para comprobar los resultados de este método, como se puede ver en la siguiente tabla.

Cluster 9
Cottevieille2007b.pdf
Jonic2003b.pdf
Jonic2007.pdf
Scheres2005.pdf
Scheres2005b.pdf
Scheres2005c.pdf
Sorzano2003c.pdf
Sorzano2004.pdf
Sorzano2006b.pdf

Sorzano2007a.pdf
Velazquez2002.pdf
Velazquez2002b.pdf

Tabla 6.14 Documentos pertenecientes al cluster 9

En el cluster 9 los documentos hablan fundamentalmente de microscopía, se escogen algunos títulos de ejemplo de los documentos, para hacernos una idea de su contenido.

Cottevieille2007b.pdf: *3D Reconstruction of macromolecular assemblies at subnanometric resolution*.

Jonic2003b.pdf: *Spline-based image-to-volume registration for three-dimensional electron microscopy*.

Jonic2007.pdf: *A novel method for improvement of visualization of power spectra for sorting cryo-electron micrographs and their local areas*.

Scheres2005.pdf: *Grid Computing in 3D-EM Image Processing using Xmipp*.

Scheres2005b.pdf: *Maximum-likelihood Multi-reference Refinement for Electron Microscopy Images*.

Scheres2005c.pdf: *Maximum-Likelihood Refinement of Electrón Microscopy Images*.

Sorzano2003c.pdf: *A multiresolution approach to orientation assignment in 3D electron microscopy of single particles*.

Sorzano2004.pdf: *Normalizing projection images: a study of image normalizing procedures for single particle three-dimensional electron microscopy*.

Sorzano2006b.pdf: *Optimization problems in electron microscopy of single particles*.

Sorzano2007a.pdf: *Fast, robust, and accurate determination of transmission electron microscopy contrast transfer function*.

Velazquez2002.pdf: *Two dimensional ARMA models and parameter adjustment to estimate the CTF of the electron microscope*.

Velazquez2002b.pdf: *A method for estimating the CTF in electron microscopy based on ARMA models and parameter adjustment*.

Este método es uno de los que mejores resultados proporciona, esto es debido a que se consiguen promediar los datos realmente importantes a la hora de clasificar, estos son las coocurrencias, que muestran las verdaderas relaciones entre los documentos.

6.9.6 Tiempo de cálculo

Se quiere comprobar para cada uno de los métodos de promediado existentes, la diferencia de tiempo de ejecución. Cada uno de los métodos se inicializan con los mismos requisitos de partida, es decir, mismos datos para promediar, igual número de grupos de salida, de descubrimiento de temas, y de filtrado de datos, de forma que no haya ningún método que tenga que trabajar de manera extra debido a alguna característica que no sea propia del método.

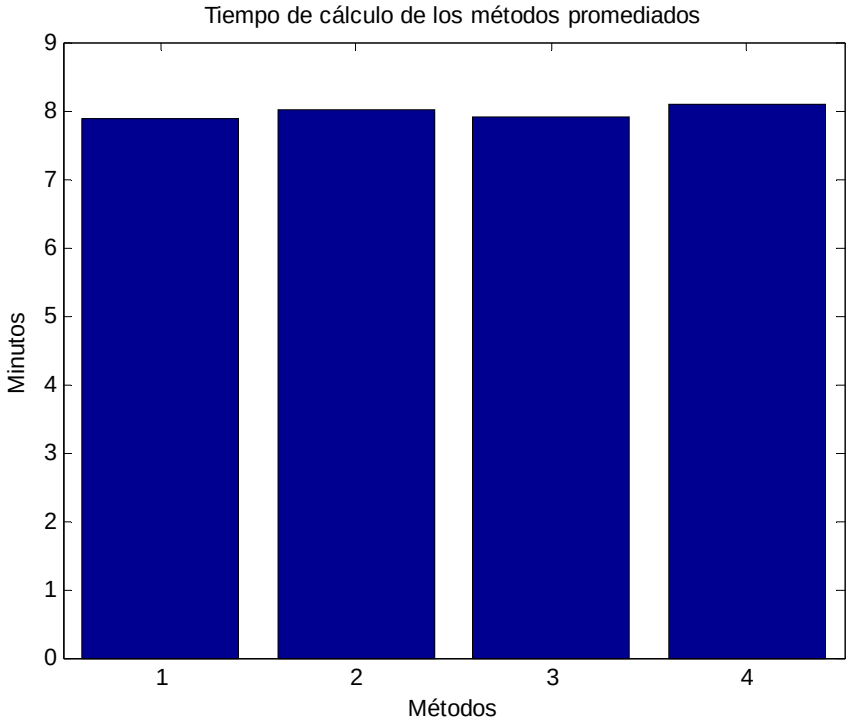

Ilustración 6.59 Tiempo de computación de los métodos promediados

En el eje de abscisas se muestran los métodos de promediado con reconstrucción, promediado basado en suma de distancias, promediado de distancias reales y promediado de coocurrencias, y se observa que los cuatro métodos tardan tiempos muy similares, esto es debido a que los algoritmos que consumen un mayor tiempo de procesamiento se realizan el mismo número de veces, y por lo tanto el tiempo que tardan en desempeñar esas tareas es muy similar, el resto de operaciones para conseguir

promediar los datos no dependen de algoritmos que consuman mucho tiempo de procesamiento, por lo que sus consumos no son significativos en el tiempo total de CPU.

7 CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

7.1 Conclusiones generales

La aplicación que se ha desarrollado combina todas las herramientas necesarias para la completa gestión de los datos desde su primer paso, la adquisición de los mismos, el procesado, con la implementación de los algoritmos necesarios y diversos métodos de resolver un mismo problema, y una interfaz gráfica intuitiva y muy completa para facilitar al usuario el acceso a los datos para que pueda analizar no sólo los resultados de los grupos finales, sino también indagar en aspectos de los documentos y sus relaciones.

Una ventaja es permitir abrir los documentos para su lectura una vez hechos los grupos para poder contrastar que la información que se le presenta es la correcta y centrarse en lo que le interesa.

El desarrollo del proyecto se centró en la idea de clasificar los documentos sin tener completamente definido un método de actuación lo que ha permitido investigar con diversas técnicas para clasificar mejor los documentos que se han ido mejorando para corregir errores encontrados y para hacer los datos finales más fiables, mediante el promedio de los resultados de los algoritmos y el filtrado de todo aquello que no es necesario.

Una ventaja para el usuario es el hecho de no tener que conocer todo lo que hay debajo del programa para obtener unos resultados tan válidos como otro usuario que conoce perfectamente el significado y funcionamiento de los algoritmos, la simplicidad con la que puede trabajar un usuario es destacable.

Se ha invertido un gran esfuerzo en combinar en una sola ventana una interacción activa con el usuario ajustando los datos mostrados a los criterios seleccionados por el usuario mediante la visualización que se actualiza en todo momento, además de integrar todos los diferentes métodos de trabajo permitiendo ajustar con precisión parámetros de los mismos desde la propia ventana. Esto representa una diferencia grande con respecto a los proyectos similares que se limitan a mostrar los datos finales sin permitir al usuario una participación un poco más activa con los resultados.

7.2 Futuras líneas de investigación

Para plantear una mejora al presente proyecto que pueda significar beneficiosa tanto para los grupos finales como para aumentar el número de mejoras y compatibilidades, existen ciertos aspectos no considerados en este documento que podrían ser desarrollados como futuras líneas de investigación.

Entre esas posibles líneas de investigación se propone una compatibilidad con mayor número de idiomas, aunque actualmente se proporciona soporte a inglés y español. Otra mejora puede consistir en utilizar documentos con los idiomas mezclados mediante el uso de un traductor como intermediario.

También es posible añadir mejoras en cuanto a la presentación de los datos utilizando más gráficas que destaquen algunas características de los documentos.

Bibliografía

[1] Chris Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, MA: May 1999.

[2] Carlos Óscar Sánchez Soriano, *Clustering* (Data Mining) , Ph, Madrid, July 2007.

[3] Alberto Pascual-Montano ,Reducción de la dimensionalidad, Universidad Complutense. Madrid: 2007.

[3] Országos Polgárőr Szövetség, *Cluster Análisis: Basic Concepts and Algorithms*, Budapest, 2006.

[4] Pascual-Montano, E. Mejía, F. Tirado, Programación bajo un modelo basado en flujos. La factorización NMF como caso de estudio, Universidad Complutense de Madrid, Septiembre 2006.

[5] García-Martínez, J. Servetto, Yolis, E, Algoritmos genéticos aplicados a la categorización automática de documentos.

[6] Everitt, Brian S. (1993). *Cluster analysis.* Halsted Press, 3ra edición.

[7] Celia Chaín, *Coincidencia y equiparación en los modelos de recuperación de la información*, Documentación de las ciencias de la información N° 27 (2004).

[8] Luis Codina, *Teoría de recuperación: modelos fundamentales y aplicados a la gestión documental*, Information world: en español N° 38 (octubre 1995).

[9] Angel Igelmo Ganzo, *Análisis de datos multivariantes*, Universidad de las Islas Baleares, 2003.

[10] Elias Pampalk, Gerhard Widmer, Alvin Chan, *Structuring of Data with Self-Organizing Maps*, Intelligent Data Analysis Journal.

[11] Wolfgang Hardle, Leopold Simar, *Applied Multivariate Statistical Análisis*. October 2003.

[12] Lola Vicente, Alfredo Vellido, *Review of Hierarchical Models for Data* Clustering and Visualization, Universidad Politécnica de Cataluña.

[13] A. Baraldi, P. Blonda, *A survey of fuzzy clustering algorithms for pattern recognition* ,October 1998.

[14] Pavel Berkhin, *Survey of Clustering Data Mining Techniques*, 2002.