TESIS DEFENDIDA POR

Dora Alicia Alvarez Medina

Y APROBADA POR EL SIGUIENTE COMITÉ

Dr. Hugo Homero Hidalgo Silva

Director del Comité

Dr. Edgar Leonel Chávez González

Miembro del Comité

Dr. Carlos Alberto Brizuela Rodríguez

Miembro del Comité

Dr. Vitali Kober

Miembro del Comité

Dr. Pedro Gilberto López Mariscal

Coordinador del programa de posgrado en Ciencias de la Computación

---

Dr. David Hilario Covarrubias Rosales

Director de Estudios de Posgrado

28 de agosto de 2008

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR DE ENSENADA

PROGRAMA DE POSGRADO EN CIENCIAS
EN CIENCIAS DE LA COMPUTACIÓN

MODELO DE VISUALIZACIÓN DE DOCUMENTOS EN BASES DE DATOS CON ALTA DIMENSIONALIDAD PARA LA IDENTIFICACIÓN DE CONGLOMERADOS Y VALORES ATÍPICOS (VA)

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de
DOCTOR EN CIENCIAS

Presenta:
DORA ALICIA ALVAREZ MEDINA

Ensenada, Baja California, México, Agosto de 2008.

RESUMEN de la tesis de **Dora Alicia Alvarez Medina**, presentada como requisito parcial para la obtención del grado de DOCTOR EN CIENCIAS en Ciencias de la Computación. Ensenada, Baja California. **Agosto de 2008.**

MODELO DE VISUALIZACIÓN DE DOCUMENTOS EN BASES DE DATOS CON ALTA DIMENSIONALIDAD PARA LA IDENTIFICACIÓN DE CONGLOMERADOS Y VALORES ATÍPICOS (VA)

Resumen aprobado por:

Dr. Hugo Homero Hidalgo Silva

En este trabajo se propone un modelo de visualización de documentos por clases (tópicos) e identificación de estructuras: conglomerados, sub-conglomerados y valores atípicos (VA). Para la identificación de los VA se definen los conceptos de ruido y VA en documentos, se propone una clasificación de VA basada en el tipo de palabras utilizadas (de propósito particular, general o compartidas). De los diferentes algoritmos de proyección de datos el de generación de mapas topográficos (GTM) ha tomado gran importancia en el marco probabilístico. Para ser utilizado con documentos se requieren ciertos cambios; algunas modificaciones propuestas consideran variables binarias y multinomiales, con resultados no satisfactorios. Dos algoritmos son propuestos: proyección y visualización de documentos (VL-ZIP) y separación de clases. El algoritmo VL-ZIP considera aplicar la función de distribución inflación de ceros con Poisson (ZIP) y un nuevo espacio latente. La eficiencia del algoritmo se evalúa con dos índices, uno basado en el clasificador de Fisher que mide la dispersión entre los datos y el error de Sammon, que mide la preservación de la topología. Dicha evaluación compara los resultados obtenidos con el modelo GTM para las distribuciones Gaussiana, Multinomial, Poisson y con el modelo de asignación Latente de Dirichlet; observando mejor desempeño con el algoritmo VL-ZIP. La segunda parte del modelo propuesto es la separación de las estructuras en los datos proyectados (conglomerados y VA). La evaluación del clasificador se realiza con las proyecciones de VL-ZIP y GTM con la función de distribución multinomial; en ambas se observa gráficamente la separación de los conglomerados. También se presenta un análisis detallado de algunos documentos proyectados fuera de su clase, identificándolos como VA's.

Palabras Clave: Visualización, conglomerados, valores atípicos, ruido, minería de texto.

BSTRACT of the thesis presented by **Dora Alicia Alvarez Medina** as a partial requirement to obtain the DOCTOR OF SCIENCE degree in Computer Science. Ensenada, Baja California. México **August 2008.**

DOCCUMENT VISUALIZATION MODEL IN DATA BASE WITH HIGH DIMENSION TO IDENTIFY CLUSTERS AND OUTLIERS

A document visualization and classification methodology is proposed. The document visualization is based on a generative probabilistic model consisting of a mixture of Zero-inflated Poisson distributions. The performance of the method is evaluated in terms of cluster forming for the latent projections with an index based on Fisher's classifier, and the topology preservation capability is measured with the Sammon's stress error. A comparison with an implementation of the Generative Topographic Mapping (GTM) algorithm with Gaussian, multinomial and Poisson distributions and with a Latent Dirichlet model is presented, observing a greater performance for the proposed method. A graphic presentation of the projections is also provided, allowing to observe the advantage of the developed method in terms of visualization and class separation. A detailed analysis of some documents projected on the latent representation is presented. The class-separation algorithm is developed to further analyze the cluster structures on the latent space. The classifier is applied to the latent proposed model and to the multinomial implementation of GTM.

Key words. Visualization, cluster, outliers, noise, text mining.

A mi padre Celestial
mis padres terrenales Lolita y Pepe
mis amadas hermanas: Elda y Sara
y los dos primeros retoños de la familia:
Are y Norma.
Los amo con mi mente y corazón...

Agradecimientos

Agradezco a mi padre celestial por su gracia infinita que en los años de doctorado, me sostuvo y me proveyó lo necesario para comer y tener un espacio para vivir. Me levantó en los días de desánimo, angustia y tribulación y puso paz en mi corazón para que mi mente para continuar con el doctorado. Por que "todo tiene su tiempo y todo lo que se quiere debajo del cielo tiene su hora"(Eclesiastés 3). Y a ti Mami, gracias por tu grande amor y apoyo que siempre me diste, forjaste en mí fuerza y tenacidad para lograr mis objetivos. Siempre me acompañaste y me diste palabras de aliento para continuar y esperar ese día tan anhelado: satisfacer la curiosidad de mi mente. De igual forma le agradezco a toda mi familia que siempre me amaron y estuvieron conmigo apoyándome en los fracasos y en las victorias, y me brindaron tantos momentos felices.

Le agradezco al CONACYT y a CICESE por haberme otorgado una beca para sostener mi estancia en CICESE. A Juan por su apoyo incondicional en todos los aspectos, gracias por tu amistad y ayudarme en los momentos difíciles. En todo el camino siempre me acompaño el Dr. Hugo Hidalgo que apoyó mis ideas y me apoyó en todo trámite; sinceramente: ¡¡Gracias!!.

En el camino del doctorado conocí y conviví con diversas personas, algunas se fueron y otras se quedaron. Yazmín García, muchas gracias por tu amistad y apoyo eres muy especial en mi vida y me alentaste en todo momento a continuar con mi trabajo de investigación. Karina Guzmán: agradezco con el alma haberte conocido antes de terminar el doctorado. Amiga, gracias por tus consejos y tu amistad que alimentaron mi alma para levantarme cuando me caí y ayudarme a ver la vida con otra perspectiva. Fuiste un elemento importante para obtener la conciencia que me llevó a la conclusión de esta tesis; me ayudaste a esclarecer mi mente, a vivir por objetivos y eliminar el ruido emocional que obscurecía mi mente. Yazmín Chávez, como olvidarme de ti gracias por tu compañía y esas charlas que enriquecieron mi vida con tantos matices, colores y sabores: gracias amiga. Le doy gracias a Dios por haber conocido a Mariela Yevénes con quien compartí una materia en UABC y me apoyó en todo momento, gracia amiga. Erika Ramos eres muy especial en

mi vida, gracias por tu amistad y apoyo incondicional; por escucharme y ayudarme en todos los sentidos a encontrar una solución optima a mis problemas.

Lydia y Carito, como olvidarme de ustedes: sus risas, comentarios, amistad y ese café matutino que alegraba mi día, sin olvidar el gran apoyo administrativo me evitaron muchos dolores de cabeza: gracias por su amistad chicas. A Lupita Morales, gracias por tu ayuda. Y como olvidar a aquellas personas tan especiales que aportaron fortaleza y palabras de aliento para continuar: Otilia, Rosi Osuna, Cleud, Pau, Hiroshi, Milka, Hna. Sisi, Hna. Dina Martínez y todas las personas del grupo de oración de la mañana, Ana Luisa, Magda, Sonia Mendoza, Joaquín Cabrera, Edith: muchas ¡Gracias!.

Juan Carlos (Cerda), Cynthia, Pedro Santana, Norma Fuentes, Lupita Rubio, Lupita Aguilera, Dania Covarrubias, Diana Escalante, Minerva Robles, Norma, María Inés Pech, Abigaíl y todos los que me brindaron una hermosa sonrisa, y amistad ¡Gracias!. Laura Franco y Alicia de Ramírez gracias por su apoyo y amistad. Luz y Lupita, amigas muchas gracias por invitarme a tijuas a desintoxicar mi cerebro y renovar mi ánimo. A mi pastor Raúl, que me ayudo a vivir en fe ciega y con gran paz en mi corazón. A Carmen Monrroy que siempre me apoyó espiritualmente y me acompaño en el proceso del examen de conocimientos básicos, gracias amiga. Gracias a Ricardo Campa que cambió mi mentalidad para transformar lo abstracto de las matemáticas en algo imaginable y me apoyó en mi examen de conocimientos básicos. Jorge Niebla y Jorge Soria, gracias chicos por que sin su ayuda me hubiera quedado sin compu en varias ocasiones. Mil disculpas si alguien no fue mencionado y a los mencionados, ¡Gracias!.

Contenido

Página

Tabla de Contenido (Continuación)

Página

Lista de Figuras

Lista de Figuras (Continuación)

Lista de Figuras (Continuación)

Lista de Tablas

Capítulo I

Introducción

De los grandes avances en la ciencia, las tecnologías de información han alcanzado logros impresionantes en las últimas dos décadas. Hace más de tres décadas el manejo de la información estaba distribuido en resmas de hojas y los procesos se realizaban de forma manual; a mediados de los setentas solo algunas compañías poseían equipo de cómputo sofisticado y las herramientas necesarias para manipular su información (Sistemas de información y bases de datos relacionales) y la información solía duplicarse (se almacenaba de forma descentralizada). Con el crecimiento acelerado de la ciencia el uso de la computadora y sus herramientas ha dejado de ser único para las grandes compañías. Siguiendo con su ciclo de desarrollo y evolución, continuamente se propone equipo más sofisticado, con mayor capacidad de procesamiento y de almacenamiento (Barquin y Edelstein, 1997; Fayyad *et al.*, 1996; Devlin, 1997; Han y Kamber, 2000).

Como resultado de todos esos avances se está dando la revolución de la digitalización (Ramírez, 2002; Darnton, 2008; Norton, 1996); existe gran controversia para dejar de almacenar la información en formatos impresos y cambiarlos a formato digital. Uno de los resultados más complejos en la evolución de las tecnologías de la información es la extracción del conocimiento y la disponibilidad de la información. Se requieren mayores esfuerzos para el manejo de la información, por lo que la minería de datos se está convirtiendo en una *necesidad* para resumir la información y acceder fácilmente a ella.

El uso adecuado de dicha información apoya a las organizaciones en sus necesidades críticas y básicas: generar estrategias, planear las inversiones, entre otras; de forma general resume el conocimiento generado a través del tiempo. La minería de datos es un conjunto de técnicas y herramientas que descubren el conocimiento inmerso en las bases de datos (BD), por medio de la exploración y el análisis de la información que encuentra de las

relaciones entre los datos. Existen diferentes teorías que definen el proceso de minado, por una parte Han y Kamber (2000) la definen como la extracción del conocimiento que procede de grandes cantidades de información. Kumar y Joshi (1998) la expresan como la búsqueda de información valiosa en grandes cantidades de datos. Y Turaisingham (1999) dice que es el proceso de plantear una búsqueda para extraer la información importante y las tendencias conocidas y desconocidas de grandes cantidades de datos.

Aunque los conceptos varían al definir el inicio del proceso, ya sea de forma exploratoria o como la búsqueda de algo particular. La minería de datos ofrece grandes beneficios, a continuación se describen algunas de las tareas de la minería de datos (Han y Kamber, 2000):

- *Clasificación.* Es la asociación de una observación con una etiqueta que proviene de un mismo conjunto (Kennedy *et al.*, 1997) y los datos son distinguibles entre sí por clases o conceptos.

- *Predicción.* Existen dos formas de predicción: encontrar la clase a la que pertenece un elemento y predicción con series. La primera se refiere a un registro del cuál se desconoce el atributo que indica la clase a la que pertenece; este puede ser el caso de registros incompletos. Por ejemplo en una base de datos que almacena las medidas de dos especies de cangrejos, si más de algún registro no poseen el atributo *genero* con la predicción se puede obtener atributo perdido. La segunda se refiere a la predicción en series, que encuentra un valor en una secuencia de mediciones. Un ejemplo de este último caso es cuando se quiere saber el consumo mensual de un producto x, entonces se utiliza el atributo del consumo mensual en un periodo determinado de tiempo y con la predicción en series se obtiene el valor correspondiente al siguiente mes de la serie en orden cronológico.

- *Análisis de conglomerado (cluster).* A diferencia de la clasificación que posee etiquetas, este análisis agrupa a los objetos sin previo conocimiento de una etiqueta que distingue la clase a la que pertenece. La agrupación está basada en una distancia o grado de disimilitud entre los objetos.

- *Reglas de asociación.* Establecen las condiciones de los eventos que ocurren frecuentemente en un conjunto de datos; es decir, son las reglas independientes que predicen la ocurrencia de un elemento basado en la ocurrencia de otros (Kumar y Joshi, 1998). Frecuentemente es comparada con la predicción, la diferencia es que las reglas se establecen basadas en el conocimiento a priori del comportamiento del sistema.

- *Análisis de valores atípicos (outlier).* Los valores atípicos (VA) son los objetos que no coinciden con el comportamiento general de un modelo de datos. La mayoría de los métodos de minería de datos los descartan y los consideran como ruido o excepciones. Sin embargo son necesarios para descubrir anomalías tales como fraudes, errores en los datos, en las series de tiempo, entre otras.

- *Caracterización y discriminación.* La caracterización proporciona la mejor representación de los atributos de un conjunto de datos, de tal forma que con poca información resume las características de los datos. Y la discriminación es similar a la clasificación, en donde los registros son etiquetados según las características comparadas con las mediciones generales de los elementos.

Se han desarrollado diferentes técnicas y herramientas para realizar dichas tareas, estas deben ser capaces de manejar diferente tipo de información: BD relacionales, imágenes, metadatos, documentos (texto en general), secuencias de ADN. Con la gran apertura que se le ha dado a la digitalización de la información, el almacenamiento de documentos ha crecido descontroladamente. Por ejemplo, la gran cantidad de información en la Web, revistas, libros, tesis electrónicas y cualquier tipo de documento complica el proceso de búsqueda y/o alguna tarea de minado. Por lo que se han generado necesidades diferentes, y se crea una nueva área de investigación, *la minería de texto*. Esta área ha evolucionado con el paso del tiempo, desde la simple tarea de preprocesar documentos hasta el preprocesado de conceptos o extracción del conocimiento de estructuras lingüísticas.

La minería de texto involucra la mayoría de las tareas de la minería de datos orientadas a texto (reglas de asociación, visualización de documentos, identificación de conglomerados, predicción de texto, identificación de VA, entre otras). También posee otras tareas específicas para el preprocesado de los datos (categorización de texto, extracción de información, extracción de términos, recuperación de información) y su almacenamiento (Feldman y Sanger, 2006). Una forma tradicional de trabajar con minería de texto requiere trasformar los documentos a un espacio vectorial; es decir, representar numéricamente los documentos en una estructura de almacenamiento virtual.

A continuación se definen algunos conceptos básicos; una palabra es la estructura mínima de un documento que en el espacio vectorial representa un índice. Y un documento es un multi-conjunto de palabras, es una unidad dentro de una colección mayor, homologamente a la teoría de BD relacional un documento representa a un registro. Colección o BD, es el conjunto de documentos, no necesariamente relacionados. Tópico, es la temática a

la que un documento se refiere; es decir la clase a la que pertenece. El índice de palabras está formado por todas las palabras que aparecen en los documentos de una colección de datos.

La representación vectorial del documento es parte del pre-procesamiento de los datos, existen diferentes técnicas para lograr éste fin. Por cuestión de tiempo y alcance de la investigación en la tesis no se realizó una investigación al respecto y se usó una técnica sencilla basada en recuperación de información (RI) (Salton y McGill, 1983). Se hace uso del modelo de *bolsa de palabras*, el cual, considera a todas las palabras del documento y el espacio vectorial de los documentos. A la representación de todos los documentos de una colección se denomina matriz de frecuencias.

La matriz de frecuencias se forma contabilizando el número de ocurrencias de cada palabra en el documento (*frecuencia de la palabra*); cada índice en todos los vectores de una colección se refiere a la misma palabra; por lo tanto aquellas palabras que no se mencionan en un documento van a tener frecuencia 0. Esto genera una característica importante en la matriz de datos, posee muchos ceros; es decir, que no todas las palabras están presentes en todos los documentos. Feldman y Sanger (2006) llama a esta condición *feature sparsity*, es decir dispersión de las palabras. En el trabajo de investigación se considera esta característica para que el modelo se apegue mejor a los datos.

En minería de texto no existe una definición que especifique las características del ruido y VA en documentos o índice de palabras. En do˜Prado y Ferneda (2008) se revisan algunos trabajos de investigación donde relacionan el ruido según el uso de palabras en: aplicaciones de búsquedas en el Web y la selección de palabras utilizada en otras tareas. En las búsquedas en el Web denominan ruido a las palabras fuera del contexto buscado; y en la selección de palabras consideran que todas las palabras que interfieren en los resultados esperados con los algoritmos de clasificación o categorización son ruido. Por otra parte en Feldman y Sanger (2006) los VA y el ruido son palabras que no aportan información para el proceso de separación de clases o categorización. Mientras que en (do˜Prado y Ferneda, 2008) son los elementos diferentes al promedio de la colección, representados al final del histograma. Por otra parte, Castellanos (2004) considera que el ruido en los documentos sólo es un proceso de mala redacción, donde alguna palabra fue mal escrita o tuvo mal empleo. Se considera que en minería de texto en un documento o el índice de palabras el concepto de ruido se apega al definido por Castellanos (2004), ya que un documento es considerado como un todo donde las mediciones corresponden a medidas completas sin

que una porción de palabra sea considerada como un error.

Por otra parte, los valores atípicos (VA) estadísticamente son lecturas que poseen medidas diferentes al resto de la población de la misma clase, mientras que el ruido es considerado como un error en la medición. De acuerdo a la definición mencionada anteriormente, el ruido en documentos es pequeño y no interfiere en la eficiencia de los resultados de algoritmos. Debido a la gran cantidad de palabras utilizadas en ellos la probabilidad de que una o dos palabras hayan sido mal escritas es muy pequeña para que el documento se vea afectado por ruido. En cuanto a los VA en documentos se considera que son documentos escritos con palabras diferentes a las de su clase, o palabras muy comunes en todas las clases. Castellanos (2004) identifica como ruido a las palabras mal empleadas, pero creo que solo son VA, porque el mal uso de palabras solo se puede dar con pocos elementos del documento completo. En el capítulo de diseño de experimentos se definen los VA y las formas en que se pueden presentar.

El trabajo de investigación en esta tesis se ubica en el área de minería de texto para encontrar un modelo para separar documentos por tópicos e identificar las estructuras formadas como sub-agrupaciones, VA y dispersión en los datos.

Diversos algoritmos de separación de conglomerado se han definido en la literatura; algunos trabajos con modelos generales de análisis de conglomerado han reportado cierta dificultad para trabajar datos con alta dimensión. En el análisis reportado por Knorr *et al.* (2000) y Wang *et al.* (2002) el problema para agrupar los datos de manera eficiente surge cuando la dimensión es mayor a cuatro. En la práctica éste tipo de problemas se presenta cuando la expectativa del algoritmo generado espera separación adecuada en la mayoría de los datos. Por lo que la búsqueda de algoritmos que mejoren el desempeño de separación de clases aún sigue vigente. En el caso del manejo de documentos, se presentan dos condiciones que hacen que la separación de documentos por clases sea una tarea ardua. Por una parte la matriz de frecuencias posee alta dimensión, y la otra característica que afecta el desempeño de los modelos es la existencia de gran número de ceros (dispersión de la distribución de palabras). Estas dos características dificultan la identificación de conglomerados con la presencia de VA.

Diferentes alternativas se presentan para la reducción de la dimensión en la matriz de frecuencias. Por una parte se encuentra el filtrado de la bolsa de palabras, que se realiza por medio de modelos de selección de palabras para reducir en lo posible la dimensión.

Y en otro sentido se encuentran los modelos de proyección de datos que transforman el espacio de alta dimensión a otro menor. Diferentes modelos de selección de palabras se han propuesto, la mayoría de ellos están basados en conocimiento a priori. La selección de palabras para minimizar la dimensión de la matriz es un área grande de investigación (Feldman y Sanger, 2006), en la que hay que trabajar para lograr un algoritmo que logre buenos resultados sin conocimiento a priori.

Los algoritmos de visualización de datos se han convertido en una alternativa para la separación de clases, y obtienen una representación gráfica de los datos en alta dimensión. De acuerdo al propósito de la tesis, se propone un algoritmo de proyección de datos para buscar algunas estructuras en la proyección. Diferentes métodos de proyección están disponibles, uno de los más representativos en proyección lineal es el análisis de componentes principales (PCA), limitado a encontrar correlaciones lineales. En este trabajo de investigación se requiere que la proyección proporcione mayor información (dispersión de los datos, sub-grupos e identificación de VA). En la búsqueda de algoritmos de visualización, se encontraron diferentes alternativas como, los Mapas Autorganizados (SOM) (Kohonen, 1988), el modelo de asignación Latente Dirichlet (LDA) (Blei *et al.*, 2003) ó el Mapa Topográfico (GTM) (Bishop *et al.*, 1998).

De estos modelos han surgido nuevas propuestas de adaptación a documentos, como el WEBSOM (Honkela *et al.*, 1996; Kaski *et al.*, 1996; Kohonen *et al.*, 1996; Lagus *et al.*, 1996) que cambia la estructura de elementos relacionales a estructuras más complejas como los documentos formados por palabras. Según Bishop *et al.* (1998) la definición probabilística del SOM es débil, por lo que propone el modelo GTM para un conjunto de datos continuos. Este modelo está basado principalmente en un espacio latente modelado por una función de densidad que ajusta los datos. De tal forma que Kabán y Girolami (2001) proponen una estructura latente con una función de densidad que modela datos discretos (como los documentos). Los resultados de proyección captan la estructura de similitud de los datos, pero es difícil el proceso de identificación de clases.

A pesar de que un algoritmo de visualización proyecta los datos en un plano de menor dimensión, las estructuras deseadas solo pueden ser identificadas visualmente sin el conocimiento de los elementos que la forman. Una forma de complementar la información deseada es aplicar un algoritmo de separación de datos. El algoritmo a utilizar debe de ser de estructura simple, que no incremente el costo computacional. Existen diversas opciones para este fin, en Jain y Dubes (1988) se revisan los diferentes algoritmos de análisis

de conglomerado para separar los datos en clases. De igual manera Kaufman y Rousseeuw (1990) realizan una revisión general de los diferentes modelos existentes y agregan otros algoritmos (PAM y CLARA). Estos modelos están basados en medidas o reglas de similitud.

En la revisión bibliografica se encontró que el modelo está altamente relacionado al tipo de dato. Según las características y dimensión de los datos de entrada se presentan diferentes algoritmos evaluados con bajo costo computacional. En Guha *et al.* (1998) y Qian *et al.* (2002) trabajan con datos espaciales, con un modelo basado en la densidad de los datos que proyectan en 2D. Por medio de un proceso de rastreo en el plano proyectado buscan los puntos más densos para encontrar las figuras que forman los conglomerados.

El trabajo de la tesis se enfoca a la minería de texto; extraer la información de un conjunto de documentos y representarlos en un plano de 2D; identificar conglomerados, subconglomerados, VA y tener acceso a las listas encontradas. En la investigación se hace uso de diversas áreas de interés, técnicas y herramientas de minería de texto, modelos de visualización de datos y de separación de clases, y el estudio estadístico de aquellos valores diferentes, los VA. En el algoritmo de visualización se piensa que es conveniente utilizar un algoritmo diferente de variables latentes que actúe de forma robusta, es decir que los resultados sean adecuados a pesar de la presencia de VA. Además de introducir una función de distribución que modele la matriz de frecuencia que posee alta dimensión y un exceso de ceros. Por lo anterior se propone utilizar una mezcla Poisson con ceros inflados (ZIP) en un espacio multidimensional; como su nombre lo indica fue formulada para conjuntos de datos con exceso de ceros tal como la matriz de frecuencia.

I.1. Organización de la tesis

La tesis está estructurada de la siguiente forma, en el Capítulo II se describen los temas de interés en el manejo de la minería de datos y de texto, se describen algunas técnicas de recuperación de información (espacio vectorial, manejo de *stop words*, algoritmo de *stemming*, selección de palabras, entre otras). También se describen dos palabras utilizadas indistintamente en la literatura de minería de texto: *ruido* y *VA*; aunque la mayoría de la gente las usa indistintamente existe una pequeña diferencia que es explicada ahí. En ese mismo capítulo se describen los antecedentes de los algoritmos de visualización y formación de conglomerados desarrollados en esta tesis (antecedentes). En la última parte del capítulo se plantea el problema a resolver y describe los objetivos a cumplir.

El los Capítulos III y IV se definen los algoritmos propuestos para lograr los objetivos de esta tesis. En el Capítulo III se formula el algoritmo de visualización de estructuras latentes basado en la mezcla de funciones ZIP; donde la estructura latente cambia a un espacio de mayor proyección y con estructura robusta. El desarrollo se divide en tres etapas: definición del modelo, entrenamiento y proyección de los datos. Por otra parte en el Capítulo IV se describe un algoritmo para la separación de las clases proyectadas en 2D por el algoritmo de visualización. El algoritmo está basado en estructuras básicas de primitivas geométricas (líneas) y un parámetro de densidad que indique el centro de un conglomerado y detecta la separación de los mismos. El algoritmo de separación de clases tiene una segunda etapa llamada r̈e-entrenamientod̈escrita en el capítulo; para cada clase o pares de clases encontradas vuelve a aplicar el algoritmo de visualización y el de separación de clases.

Antes de presentar los resultados de los algoritmos propuestos, en el Capítulo V se analizan las características deseadas en los experimentos y se obtienen las herramientas necesarias para evaluar la eficiencia de los algoritmos. Para medir la eficiencia del algoritmo, describo los diferentes escenarios donde se desea cierta eficiencia; es decir, establecer las situaciones difíciles donde espero una respuesta adecuada del algoritmo. Se definen las cuatro BD con las que se trabaja, tres de ellas poseen tres clases diferentes: Reuters (S1), 20-Newsgroup (S2) y Medlars (S3); y una posee cinco clases 20-Newsgroup (S4). Enfatizo la importancia y beneficios de realizar un análisis previo de las BD, que ayudan a comprender los resultados obtenidos. En algunos trabajos de visualización cuando obtienen resultados diferentes a los esperados (todos los elementos de una clase juntos); adjudican las diferencias al término *ruido*. Sin embargo el análisis realizado demuestra que algunas palabras son compartidas por ciertas clases, por lo que algunos datos se proyectan traslapados. Esto no significa que sean ruido o VA, solo son documentos que comparten mucha información y su separación es difícil. Respecto a los VA, encontré ciertas características que los dividen en tres tipos diferentes definidos en el capítulo. El trabajo de calibración es un proceso importante en los experimentos, por lo que consideré importante definir una estrategia para este fin. Al pensar en los puntos a seguir, me dí a la tarea de investigar el comportamiento de los datos dentro del algoritmo, dos aspectos requeridos para asignar valores a los parámetros son las restricciones y características de los parámetros. Describo dos criterios para medir la eficiencia del algoritmo de visualización y los valores esperados de las variables que resultan de aplicar los criterios.

En el Capítulo VI se presentan los resultados de los algoritmos de visualización y separación de clases. La sección de experimentos es una guía para obtener los resultados que demuestran las habilidades de los algoritmos propuestos. En este trabajo se evitó tener experimentos ambiguos. En la primer sección se presentan los resultados obtenidos con las cuatro BD y su comparación con el algoritmo LDA y el GTM con las funciones Gaussiana, Poisson y Mulinomial. En la segunda sección se presentan los resultados del algoritmo de separación de clases; hace uso de los resultados proyectados en 2D de las cuatro BD seleccionadas con los algoritmos VL-ZIP y GTM con función Multinomial. Los resultados del re-entrenamiento del algoritmo de separación de clases solo se realizan para la BD S4.

Finalmente en el Capítulo VII se plasman las conclusiones a las que se llega con el algoritmo de visualización de documentos con una estructura latente y la función ZIP (LV-ZIP). Se demuestra la separación de clases, cada una de ellas concentrada en un área específica y se obtiene suficiente espacio de separación entre clases. La separación obtenida con el algoritmo clasificador propuesto también es visible en las proyecciones.

Capítulo II

Minería de datos

Debido a la cada vez más creciente generación de conocimiento e información, se ha exacerbado el problema de extracción de conocimiento de las fuentes de información. El uso adecuado de dicha información es de gran beneficio y significa ahorro de tiempo; de esta gran necesidad surge la minería de datos. Minar la información es extraer el conocimiento de los datos. Han y Kamber (2000) la definen como la extracción o el minado del conocimiento procedente de grandes cantidades de información. Por otra parte Turaisingham (1999) dice que es el proceso de plantear una búsqueda, extraer la información importante y las tendencias conocidas o desconocidas de grandes cantidades de datos. La importancia de extraer la información relevante de la base de datos reside en la interpretación que se le da al conocimiento obtenido.

A Partir de dicha necesidad de información surgen diferentes tareas y herramientas (Han y Kamber, 2000) para satisfacerlas: clasificación de datos, predicción de tendencias basada en los datos, análisis de conglomerado, reglas de asociación, análisis de valores atípicos, entre otros. La minería de datos está formada por una gama enorme de tareas y herramientas que maneja diferentes tipos de información: Bases de datos relacionales, imágenes, metadatos, documentos (texto en general), secuencias de ADN, entre otras.

El trabajo con documentos es diferente y requiere manipulación específica; además algunas necesidades de información son diferentes. De tal forma que surge la minería de texto; rama de la minería de datos de la que se han desencadenado diferentes áreas de investigación: la recuperación de información, clasificación y análisis de conglomerados, análisis de la bolsa de palabras, técnicas de indexación de palabras, entre otras. Nuestro interés en ésta tesis en el uso de técnicas para visualizar documentos y obtener la separación de las clases en las que se dividen (minería de texto).

La visualización de datos es una herramienta importante en la minería de datos, que permite obtener una representación gráfica de la información que posee alta dimensión. Se proyectan los datos en un espacio de menor dimensión, deseando capturar las diferentes estructuras en el espacio con dimensión proyectado. Sin embargo, cuando la dispersión en los datos es muy grande, las estructuras representadas en la gráfica no son fáciles de distinguir; por lo que su apreciación varía según la percepción de quien la ve. Además, se desconoce la identidad de los elementos que forman a un conglomerado, o de aquellos localizados lejos de su grupo (valores atípicos, VA). Los algoritmos de separación de clases (análisis de conglomerados) proporcionan dicha información requerida.

En algunos trabajos de investigación se ha reportado la dificultad y poca eficiencia de los algoritmos de conglomerado en BD con alta dimensión. Knorr *et al.* (2000), Yanchang y Junde (2001) y Wang *et al.* (2002) reportaron problemas en la obtención de conglomerados en datos con alta dimensión (mayor a cuatro). Esto se debe a que las relaciones de algunos atributos en los registro no siempre se dan entre los mismos objetos. Por ejemplo, el atributo a de un registro puede estar relacionado con otro que se encuentra en el mismo grupo, mientras que el atributo b se relaciona con un registros que pertenece a otro grupo diferente. La factibilidad de trabajar con datos de este tipo aumenta conforme se incrementa el tamaño de la dimensión.

La visualización de documentos permite apreciar la separación de las estructuras formadas, y la dispersión entre los datos del conglomerado (existe una relación entre la similitud de los datos y la dispersión entre ellos). Sin embargo, no se conoce la identidad de los elementos que forman cada conglomerado. Pero, si a estos resultados se les aplica un algoritmo de conglomerado se obtiene dicha información y se evita el problema que se presenta en datos con alta dimensión.

En las siguientes secciones se describen tres áreas de interés para el desarrollo de la investigación. En la primera se describen algunos elementos de la minería de texto; técnicas de recuperación de información (RI), teoría de espacio vectorial, bolsa de palabras, herramientas para la selección de palabras. En la segunda y tercer sección se revisan los trabajos relacionados con la visualización de datos y separación de conglomerados; para establecer los antecedentes de ésta investigación. Finalmente, después de tener las herramientas necesarias, se define el trabajo de investigación de la tesis y los objetivos a alcanzar.

II.1. Minería de texto

El trabajo con documentos requiere de aplicar un tratamiento previo a su uso, con técnicas y herramientas de recuperación de la información (RI). Cada documento se representa como un elemento de un espacio vectorial; el índice representa una palabra del conjunto de palabras (*bag of words*, "*bolsa de palabras*") y el contenido es la frecuencia, es decir el número de veces que se repite la palabra en el documento[1]. Para cada documento de la BD se obtiene su vector correspondiente que es parte de la matriz que representa a todos los documentos (matriz de frecuencia). Esto significa que la dimensión de la matriz es $D \times N$, donde D es la dimensión de la *bolsa de palabras* y N el total de documentos.

II.1.1. Construcción de la matriz de frecuencias

La matriz de frecuencias está formada por N vectores equivalentes al número de documentos; y D renglones, el total de palabras en todos los documentos -bolsa de palabras. Durante este proceso se consideran dos aspectos importantes:

- *Stop words.* Son palabras muy frecuentes que no proporcionan información que indique si el documento pertenece a una clase u otra. Por esta razón no son consideradas en la bolsa de palabras; cuando se eliminan se reduce la dimensión del espacio vectorial. Algunos ejemplos de *stop words* en el idioma inglés son: *the, you, well, fine, ok, and, may, anyone,* entre otras.
- *Algoritmos de Stemming.* Reducen las palabras a la raíz gramatical, existen diferentes algoritmos, aquí se utiliza el algoritmo de Porter (Porter, 1980). Elimina los finales morfológicos comunes e inflexionales de una palabra y remueve los sufijos de las palabras. Por ejemplo, a la palabra *relational* se le cambia la terminación *ational* por *ate*, después de aplicar el algoritmo de Porter nos queda *relate*. Además de reducir la dimensión de la bolsa de palabras el algoritmo realza aquellas palabras útiles para la identificación de clases y evita tener palabras con el mismo significado y efecto.

En el Algoritmo 1 se presentan los pasos para encontrar la representación en espacio vectorial; obtener la matriz de frecuencias y la bolsa de palabras. A partir de esta matriz se puede trabajar con alguna técnica o herramienta de algebra vectorial ó con modelos basados en funciones de probabilidad para describir la contabilización de datos (*data count*).

[1] A este proceso estadístico que se le conoce como contabilización de palabras (*data count*).

Algoritmo 1 Construcción de la matriz de frecuencia.

1. Inicializa $bolsa - palabras$.
2. Inicializa matriz de frecuencia MF.
3. Para cada documento i.
   - Mientras haya palabras en el documento:
     a) $Palabra_1 =$ Leer palabra.
     b) $Importante =$ Busca ($Palabra_1$, ***stop-list***)
     c) Si $Importante =$ "SI"(la palabra no se encontro en stop list)
        1) $Palabra_f =$ algoritmo-Porter($Palabra_1$).
        2) $j =$ Busca-índice-palabra ($Palabra_f$, ***bolsa-palabras***).
        3) Si $j = 0$
           No se encuentra en la bolsa de palabras
           a' j = asignar índice a palabra.
           b' Insertar $Palabra_f$ en ***bolsa-palabras***.
        4) Aumentar frecuencia $MF_{j,i} = MF_{j,i} + 1$.

En la práctica no se utiliza la matriz de frecuencia, sino que se sustituye por una matriz de pesos, donde los datos representan la fuerza que cada palabra tiene en el documento. Este proceso es similar a la normalización de los datos; una de las fórmulas más usadas es

$$Peso_{ji} = \frac{MF_{j,i}}{docF_i}$$

donde $MF_{j,i}$ es la frecuencia de la palabra j en el documento i y $docF_i$ es el total de palabras que posee el documentos i. A partir de aquí se aplican herramientas de algebra vectorial.

En este trabjo se considera el modelo de la matriz de frecuencia por métodos probabilísticos. Una característica de la matriz de frecuencias es que algunos elementos poseen frecuencia 0; porque la palabra no aparece en el documento y luce como la de la Figura 1. La tabla posee un índice de documentos (columnas, índice i) y otro de palabras (renglones, índice j). En la Figura de la derecha se representa la bolsa de palabras, cuyo índice corresponde al de la matriz de frecuencia. Feldman y Sanger (2006) lo llaman "*feature sparsity*", es decir la dispersión de las palabras en la matriz de frecuencia; la cual es común y está presente en la mayoría de las matrices.

Matriz de frecuencias

	Documento							
Palabra	1	2	3	4	...	50	51	...
1	0	5	0	2	...	0	3	...
2	0	0	3	4	...	5	4	...
3	2	0	0	0	...	8	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
101	3	0	3	6	...	1	0	...
102	0	7	1	0	...	0	1	...

Bolsa de palabras

Índice	Palabra
1	bolsa
2	basura
3	árbol
4	vecino
5	serie
⋮	⋮
101	biblioteca
102	libro

Figura 1. Representación de la tabla de frecuencias y la bolsa de palabras. Las columnas indican el número de documento y los renglones el número de palabra que corresponde a la bolsa de palabras.

II.1.2. Selección de palabras, reducción de la dimensión

El preprocesado de la información para minería de texto es tan extenso como se quiera hacer; existen estructuras básicas que solo transforman los documentos al espacio vectorial y eliminan aquellos elementos denominados "*paja*". Los conjuntos de datos con alta dimensión implican una gran dificultad para encontrar relaciones entre las clases. Existe una gran variedad de técnicas que filtran la bolsa de palabras, de tal forma que se obtienen las palabras que mejor representan a cada clase.

La selección de palabras es un área de investigación en la minería de texto, por su gran impacto sobre los resultados de procesos posteriores de minería. Una técnica utilizada para este fin es la categorización; en Feldman y Sanger (2006) se presenta un resumen de diferentes técnicas para obtener una buena selección de palabras. Algunas de ellas: Indexación latente semántica (LSI), aprendizaje de máquina, clasificación probabilística, reglas de edición, entre otras. Otro método que ha tenido gran uso es *Información mutua* (MI), que se basa en el conocimiento a priori de la clase a la que pertenece cada una por medio de un conjunto de entrenamiento (Cover y Thomas, 1991).

Algunas de las técnicas mencionadas requieren un conjunto previo de palabras que indique la clase a la que pertenecen; en aplicaciones en tiempo real el uso de esas técnicas no es factible. Por lo cual se usará una técnica básica, *Resolving Power* (Salton y McGill, 1983) que se basa en el histograma del uso de las palabras en documentos. Utiliza el número de documentos en donde cada elemento de la bolsa de palabras aparece, denominado *fre-*

cuencia por documento (FD). Después de ordenarlos de forma descendente (por el número de documentos donde aparece), se obtiene el rango de las frecuencias. Después se obtiene la gráfica del histograma, en donde se eliminan todas las palabras que aparecen en las orillas; es decir, las de mucho y las de poco uso. Tal como se muestra en el Algoritmo 2; se seleccionan dos límites para obtener el rango de las palabras que se consideran importantes para la BD.

Algoritmo 2 Selección de palabras con *Resolving Power*.

---

1. Para cada palabra de la *bolsa-palabras*.
   a) Obtener freccuencia $FD_j = \sum_i^N TF_{j,i}$.
2. Ordenar frecuencias $FD = sort(FD,$ descendente$)$.
3. Para cada Frecuencia FD calcular el rango.
   a) Calcula frecuencias repetidas $repite = repetidas(FD_j)$.
   b) Si $repite > 1$
      - Calcula rango $rr = j + \frac{j-repite-2}{2}$.
      - Para todos los elementos repetidos asignar el mismo rango (desde j hasta $j + repite - 1$).
        - $rango_j = rr$.
   c) De lo contrario, incrementa índice.
4. Para cada j.
   a) $RF_j = FD_j \times rango_j$.
5. Graficar RF.
6. Indicar Límite superior e inferior.
7. Obtener palabras entre los dos límites.

En Salton y McGill (1983) se dice que los dos extremos de la figura contienen información de tipo *paja* que no contribuye en nada para separar los elementos de una clase y otra. En la Figura 2 se muestra gráficamente ésta técnica; las palabras a utilizar en la matriz de frecuencia están en la sección B de la gráfica. La región A pertenece a aquellas palabras que son de uso muy común, presente en la mayoría de los documentos que solo añade

confusión al método a utilizar. Por el contrario, la región C corresponde a las palabras que pocas veces se mencionan en toda la BD; el impacto que tienen sobre los resultados es reducir la eficiencia porque los documentos no se pueden relacionar usando esas palabras. En la Figura 2 se observa que en la región C existen muchos elementos cuyo producto $frec \times rango$ es prácticamente constante. Esas palabras son de las poco usadas, pero además son completamente atípicas ya que solo son mencionadas en un máximo de 5 documentos. De tal forma que si se dejan en el conjunto de datos el proceso será mas tardado debido a que la dimensión se incrementa y por consecuencia el número de operaciones. Las regiones A y C corresponden a las palabras que Yang y Zhang (1995) y Peters y Koster (2002) consideran ruido, pero como se mencionó anteriormente sólo son palabras irrelevantes que se eliminan.

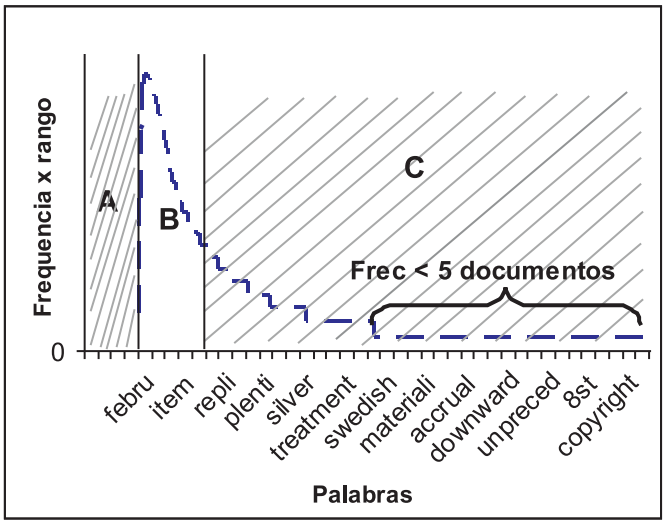


Figura 2. Técnica de selección de palabras con *Resolving power*. Sección A y C se desechan, región B son las palabras seleccionadas.

II.1.3. Ruido y valores atípicos (VA)

Un tema poco abordado en la minería de texto es el concepto de *ruido*, en algunos trabajos se atribuye cuando los resultados son difíciles de explicar, o no son los esperados. Frigui y Nasraoui (2004) en su trabajo de investigación lo relacionan con documentos pequeños que poseen poca información, o con resultados fuera de lo esperado. Por otra

parte, Chakrabarti *et al.* (1999) trabaja con el uso de taxonomía y discriminante de palabras para la búsqueda en Internet, consideran ruido a todas aquellas palabras fuera del contexto de búsqueda. En el trabajo de Yang y Zhang (1995) y Peters y Koster (2002) consideran a las palabras no relevantes o poco significativas ruido, y que los resultados de categorización de documentos se mejoran cuando se eliminan este tipo de palabras. Sin embargo, los métodos propuestos para eliminar dichas palabras consideradas ruido dependen de un criterio subjetivo a un valor establecido. Por lo que el ruido se convierte en una condición relativa; si el criterio disminuye algunas palabras salen del rango especificado para ser considerado ruido, entonces lo que antes era ruido con un nuevo parámetro ya no lo es. Y considero que el ruido en cualquier condición posee las características que lo identifican como ruido. Las características de las palabras que Yang y Zhang (1995) y Peters y Koster (2002) consideran ruido, en esta tesis las considero no relevantes (*paja*) y se explican en la siguiente sección. Por otra parte Castellanos (2004) maneja el concepto de texto sucio. Por medio de un análisis realizado con datos obtenidos de una BD de correos electrónicos descubre que los errores producidos están relacionados con palabras mal escritas, o mal empleadas, abreviaciones, símbolos especiales, uso incorrecto de la gramática, ambigüedades en la redacción, entre otras. Ella define una técnica de pre-procesamiento de documentos para eliminar la mayor cantidad de ruido.

Algunos investigadores hablan de la gran relación y/o confusión que existe entre el ruido y VA (Barnett y Toby, 1991; Hampel *et al.*, 1986). La característica que identifica a un dato como ruido o VA depende de la procedencia de los datos:

- Los VA pueden ser elementos distantes de la distribución como se muestra en las Figuras 3a y 3b. En la Figura 3a todos los datos son considerados como parte de la distribución *F*, donde el primer y el último son elementos extraídos de la muestra. En la Figura 3b el último dato es considerado VA porque no está dentro de esa distribución.

- También se dice que un VA es un dato extraño que pertenece a una distribución diferente *G* (Figura 3c y 3d), de la que se pretende encontrar la distribución de procedencia. Las dos distribuciones de la Figura 3c contienen algunos datos en marcados con círculos; desde un punto de vista pueden ser considerados VA porque están afuera de la distribución *F* o *G*. Pero, vistos desde otra perspectiva no están afuera de *F* ó *G*, si no que son parte de ellos localizados en la mitad de *F* y *G* por lo que no son considerados VA. Lo mismo sucede en la Figura 3d.

En estadística el ruido se define como una variable aleatoria que afecta los resultados;

mientras que los VA son observaciones cuya apariencia es inconsistente con el resto del conjunto de datos. Aunque los VA son diferentes al promedio no significa que sean errores de medición; algunos son datos poco representativos de la muestra que pueden indicar la presencia de un fenómeno (Liu *et al.*, 2002) o son datos generados por otro mecanismo (Barnett y Toby, 1991). Por otra parte, Hawkins (1980) dice que el ruido proviene de una distribución contaminada; también puede ser explicado como error o interferencia en la toma de la muestra.

Después de analizar algunos conceptos y teorías sobre el ruido, los VA y su procedencia, el ruido en documentos puede definirse como Castellanos (2004); errores de escritura, gramática, abreviaciones, palabras mal escritas, etc. Con base en lo anterior un VA es un documento con estilo de redacción diferente y/o el uso de palabras clave diferentes a las utilizadas en los documentos de la misma clase.

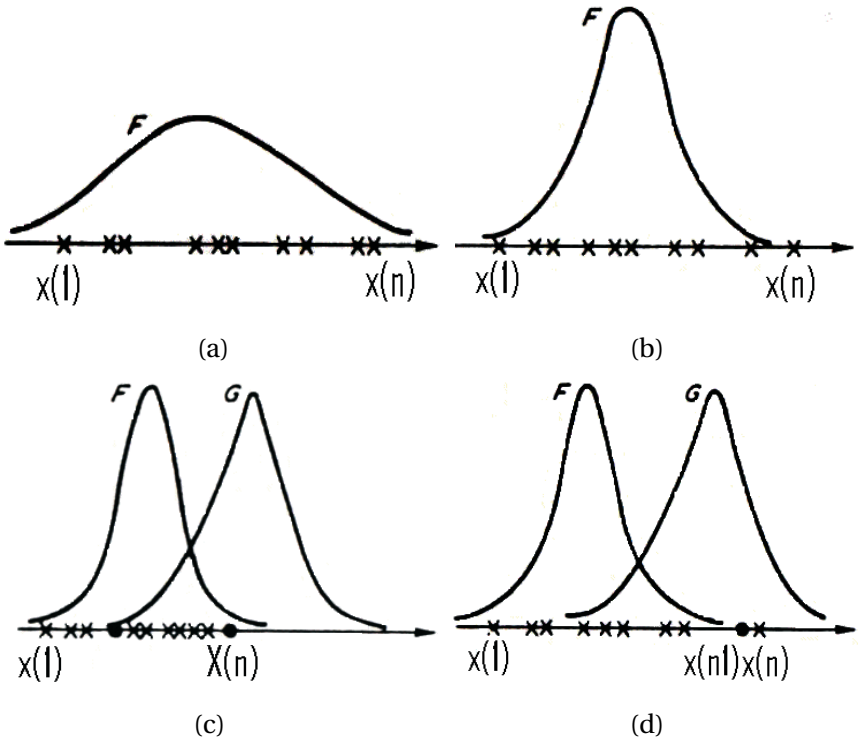


Figura 3. Identificación de VA o ruido según la procedencia de los datos; a y b) el dato puede ser considerado como VA a los demás; c y d) los círculos indican a elementos extremos o que son miembros de otra distribución.

II.2. Modelos de visualización de documentos

En un conjunto de documentos existen diferente tipo de información, algunos datos se pueden relacionar más entre sí para formar estructuras de agrupación. Pero también existen otros documentos que no se relacionan fácilmente con otros, los cuales son considerados valores atípicos (VA). Cuando la dimensión de la BD es alta es difícil identificar las estructuras formadas (conglomerados, VA y subgrupos de VA). Una herramienta útil para observar estas estructuras es la visualización de datos, que proyecta los datos con alta dimensión a un plano de menor dimensión y captura las características importantes de los datos.

Visualizar los datos en un plano de 2D proporciona ciertas ventajas que los modelos de clasificación no poseen. Se puede identificar la similitud de los datos por medio de la cercanía de proyección, ó apreciar aquellos que son parte de un grupo o identificar que tan lejanos están los VA o pequeños grupos de VA. Cuando el conjunto de datos son documentos, la información antes mencionada sirve cuando se busca información diferente a la que se sabe que prevalece en un grupo. Por ejemplo, si se busca información muy específica diferente al común denominador de la que se relaciona con cierto tema, es posible encontrarla visualmente en la proyección de los documentos.

Existen diferentes métodos disponibles para proyección de datos, uno de los más representativos de proyección lineal es el análisis de componentes principales (PCA), que se limita a encontrar una correlación lineal en los datos. Diversos métodos de proyección no lineales han sido propuestos, basados en diferentes estructuras y teorías. Como los basados en redes neuronales (Mao y Jain, 1993), mapas autorganizados (SOM) (Kohonen, 1988), escalamiento multidimensional (Kruskal, 1964), y modelos probabilísticos como el de mapas topográficos (GTM) (Bishop *et al.*, 1998). Este último esta basado en un espacio de variables latentes, que busca una representación de los datos originales en alta dimensión a un espacio latente. Por otra parte, el SOM ha sido ampliamente usado en diferentes investigaciones, por lo que existen diferentes adaptaciones para texto (Ritter y Kohonen, 1989), (Miikkulainen, 1993). El WEBSOM (Honkela *et al.*, 1996; Kaski *et al.*, 1996; Kohonen *et al.*, 1996; Lagus *et al.*, 1996) es una implementación del SOM para trabajar con colecciones de documentos.

El GTM es una de las primeras alternativas en nuestro trabajo, el cuál originalmente fue propuesto considerando una distribución con ruido Gaussiano. Debido a que el algoritmo

original está basado en la función de distribución Gaussiana, no se obtienen buenos resultados cuando los datos son texto. Basado en el GTM Girolami (2001) formuló un algoritmo para datos binarios modelados con una distribución multidimensional Bernoulli. Kabán y Girolami (2001) adaptaron el algoritmo GTM para la familia de distribución exponencial. Otras aplicaciones para visualización de conglomerados con el algoritmo GTM se han publicado en: Bishop *et al.* (1998), Tinǒ y Nabney (2002) y Yang y Zhang (2001).

La implementación de los modelos multinomial y Poisson por Kabán y Girolami (2001) ha representado mejor los documentos; pero la proyección latente no proporciona una clara separación de clases. Por otra parte, Vellido y Lisboa (2006) desarrollaron un algoritmo de variables latentes, en el que consideraron la identificación de conglomerados y VA en los datos. Como principal antecedente del algoritmo de visualización de datos en la siguiente sección se describe el algoritmo GTM.

A continuación se describen brevemente tres de los algoritmos de visualización más representativos del trabajo a realizar en la tesis. De los trabajos revisados anteriormente, se considera que nuestro algoritmo base es el GTM.

II.2.1. Modelo Generativo de Mapas Topográficos (GTM)

Es un modelo basado en la densidad de probabilidad $p(t)$ que describe la distribución de los datos, supone un conjunto de N datos t con dimensión D, representado por K variables latentes x de dimensión L. El objetivo del modelo es encontrar una representación para la distribución $p(t)$ por medio de la función de transformación ("mapping") que transporta los datos con dimensión D al espacio latente de dimensión L ($\mathbb{R}^D \Rightarrow \mathbb{R}^L$) (Svensen, 1998). Las variables latentes son auxiliares en la función de transformación ($y(x;W)$) regida por el parámetro W. Las dos etapas del algoritmo GTM son:

- Entrenamiento. Aplicar el algoritmo EM para obtener los parámetros de la función de densidad que representa la relación entre los datos y las variables latentes. Los datos son modelados por medio de una función de la cuál se supone que provienen los datos, en la que se incluye una relación de transformación del espacio latente.

- Visualización. Se representan gráficamente los datos proyectados en el espacio latente. En este proceso final se pueden observar las relaciones de conglomerado entre los datos y los VA.

Entrenamiento del GTM.

Se define una función de densidad que modela los datos, Bishop *et al.* (1998) la define como una mezcla de *K* funciones Gaussianas (que interactúan con el espacio latente). El espacio latente está formado por *K* variables latentes y *M* centros; relacionados por funciones bases radial (FBR). Dichas funciones son estructuras Gaussianas que transforman la dimensión de los datos en el espacio latente. La densidad condicional de los datos en el espacio latente está dada por:

$$p(t|x,W,\beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\sum_{j=1}^{D}\Big(t_j - y_j(x;W)\Big)^2\right\} \tag{1}$$

donde t son los datos de entrada, β es la varianza del ruido, $y(x;W)$ es un modelo generalizado de regresión lineal que transforma las variables latentes en el espacio de datos correspondiente:

$$y_j(x,W) = \sum_{m}^{M} \phi_m(x) w_{mj}, \tag{2}$$

donde W es una matriz de $D \times M$ con los parámetros desconocidos w_{mj}, y ϕ son las FBR evaluadas en las variables latentes x. Las FBR son definidas por:

$$\phi_m(x) = \exp\left\{-\frac{||x-\mu_m||^2}{2\sigma^2}\right\}, \tag{3}$$

con σ un parámetro desconocido. Integrado sobre las variables latentes, la distribución de probabilidad de los datos se expresa en función de los parámetros β y W:

$$p(t|W,\beta) = \int p(t|x,W,\beta)p(x)dx.$$

Para estimar los parámetros W y β dado t_n con el espacio latente descrito por ϕ, primero se define la función de verosimilitud considerando (1) y suponiendo que la distribución de las variables latentes corresponde a una función delta colocada en cada punto de la malla:

$$\mathcal{L} = \prod_{n}^{N}\Big[\frac{1}{K}\sum_{k}^{K} p(t_n|x_k,W,\beta)\Big]. \tag{4}$$

El algoritmo iterativo EM (Dempster *et al.*, 1976) estima los parámetros W y β del logaritmo de la verosimilitud. El nombre del algoritmo proviene de las iniciales de los dos pasos que lo forman: *Expectation* (Esperanza, Paso-E) y *Maximization* (Maximización, paso-M). La esperanza del logaritmo de la verosimilitud relativa (4) con los parámetros antiguos y nuevos está dada por:

$$
\begin{aligned}
E(\log \mathcal{L}) &= \sum_{n}^{N}\sum_{k}^{K} p^{ant}(W,\beta|t_n)\log\Big(\frac{1}{K}p_{nk}(t_n|x_k,W,\beta)\Big) \\
&= \sum_{n}^{N}\sum_{k}^{K} \theta_{nk}\log p_{nk}(t_n|W,\beta), \qquad (5)
\end{aligned}
$$

donde, θ_{nk} es la esperanza que relaciona los parámetros en la iteración anterior con los actuales, es decir, la probabilidad de que t_n pertenezca a la mezcla del componente k. Se calcula con la regla de *Bayes*,

$$
\theta_{kn} = \frac{p(t_n|x_k,W,\beta)p(x_k)}{\sum_{k'} p(t_n|x_{k'},W^{ant},\beta^{ant})p(x_{k'})} \qquad (6)
$$

En el paso-M se maximiza (5) para encontrar los parámetros (W y β). Cuando el algoritmo converge a un máximo, se obtiene la reducción de la dimensión de los datos realizando el productoΘX y se grafica para visualizar la proyección.

II.2.2. Combinación de clases latentes para la visualización de datos discretos

Retomando el tema de la organización topográfica para la búsqueda de estructuras de conglomerado, Kabán y Girolami (2001) presentan un trabajo enfocado al buen funcionamiento con datos discretos. Definen al conjunto $D = (d_{tn})_{t=1,\ldots,T;n=1,\ldots,N}$ como los datos observados; donde los datos son la expansión ruidosa de variables generativas ocultas.

Como un modelo generalizado, consideran la familia de distribución exponencial para modelar el ruido en la generación de la siguiente forma,

$$
p_G(x|\theta) = exp\{\theta x - G(\theta)\}p_0(x), \qquad (7)
$$

donde θ es un parámetro canónico de la familia de distribución exponencial, $G(\theta) = \ln\Big(\int exp(\theta x)\ p_0(x)dx\Big)$ es la función acumulativa y es estrictamente convexa respecto de θ, $p_0(x)$ es un factor independiente de los parámetros.

El ruido es modelado por la probabilidad condicional, con componentes latentes c_k y el parámetro canónico formado como una mezcla lineal $\theta_k = Ac_k$ de la siguiente forma,

$$
p_G(d_n|c_k,A) = exp\{Ac_k d_n - G(Ac_k)\}p_0(d_n). \qquad (8)
$$

El modelo es similar al GTM (Bishop *et al.*, 1998), con una estructura latente en 2D formada por una malla de puntos uniformes X de $M \times K$, (M = 2), transformada por un conjunto de L vectores bases lineales Φ_l, de la siguiente forma,

$$C = \Phi(X) \tag{9}$$

donde C es una matriz de dimensión $L \times K$.

De forma general, definen el modelo con ruido de la siguiente manera

$$d_n = g(Ac_n) + n, \tag{10}$$

donde c_n son las variables latentes no observadas, n es el ruido modelado y $g(.)$ es el parámetro esperado del vector gradiente de la función acumulativa, representado por,

$$m_k = g(Ac_k) = \nabla_{\theta_k} G(Ac_k), \tag{11}$$

∇ es el operador gradiente y $g(.)$ la función de liga ó enlace.

A partir del modelo definido en (7), se define la función logaritmo de la verosimilitud como,

$$L = \sum_{n=1}^{N} \log\{p(d_n)\} = \sum_{n=1}^{N} \log\Big\{ \sum_{k=1}^{K} p(d_n|c_k)P(c_k) \Big\} \tag{12}$$

Al igual que con el GTM, se aplica el modelo EM para encontrar los parámetros que modelan la función; donde la relación entre los parámetros anteriores y los nuevos está dada por,

$$Q = \sum_{n=1}^{N} \sum_{k=1}^{K} p^{anterior}(c_k|d_n) \log\{p^{nvo}(d_n|c_k)P^{nvo}(c_k)\}, \tag{13}$$

donde,

$$p^{anterior}(c_k|d_n) = \frac{p(d_n|c_k)P(c_k)}{\sum_{k=1}^{K} p(d_n|c_{k'})P(c_{k'})} \tag{14}$$

donde $p^{anterior}(c_k|d_n)$ se describe con la regla de Bayes; se estima en el paso E del algoritmo EM y es considerada constante en la fase de M que se explica con mayor detalle en la sección III.2.

II.3. Algoritmos de separación de clases

Una forma de explorar la información y presentar un resumen de las relaciones encontradas en ella, es con el análisis de conglomerado. La información obtenida por dicho análisis refleja las relaciones encontradas. Jain y Dubes (1988) lo define como el proceso de clasificar objetos en subconjuntos con el mismo significado en el contexto de un problema particular. Según Kaufman y Rousseeuw (1990) el análisis de conglomerado se refiere a una herramienta exploratoria sin conocimiento previo; no se requiere comprobar ninguna hipótesis de trabajo, solo ver la información que ofrecen los datos.

Dos métodos muy importantes para encontrar conglomerados son partición y jerárquico. El primero separa los datos en k grupos seleccionados por la similitud entre ellos; que es medida con una función de distancia o regla que indica si la información pertenece o no al grupo, un ejemplo es el algoritmo *k*-medias. El jerárquico agrupa los datos al medir la similitud entre ellos y los acomoda de forma jerárquica de mayor a menor similitud (Jain y Dubes, 1988), un ejemplo es el método de la liga simple.

Kaufman y Rousseeuw (1990) presentan dos algoritmos basados en el método de partición; el primero es el PAM ("Partitioning Around Medoids") para BD pequeñas. PAM busca *k* centroides y aproxima los datos cercanos a ellos de acuerdo a ciertas reglas de pertenencia. El segundo algoritmo propuesto fue CLARA; creado para BD muy grandes, es una extensión del PAM y disminuye el costo computacional. Secciona la BD en M diferentes subconjuntos que son re-entrenados con el algoritmo PAM. Finalmente, compara los M resultados y se queda con los mejores centros encontrados.

Otro tipo de algoritmo para generar conglomerados se forma con la mezcla del algoritmo jerárquico y el particional que representan la información en un plano. Tal es el caso del algoritmo CURE (Guha *et al.*, 1998) que encuentra *k* grupos y los redefine con una estructura jerárquica. Esto por medio de un algoritmo que compacta los datos y los representa como estructuras elípticas o circulares. Por otra parte, Qian *et al.* (2002) propone una versión mejorada llamada CURE-NS para conglomerados que, cuando se proyectan poseen diferentes figuras geométricas. Siguiendo con los algoritmos basados en figuras, el trabajo de Calafiore (2002) aproxima los datos a una estructura elíptica. El trabajo de Yanchang y Junde (2001) encuentra conglomerados de datos proyectados en un plano 2D, propone un algoritmo basado en isolíneas para separar los datos en conglomerados; identifica las regiones del plano que poseen mayor densidad y por medio de un umbral identifica cuales

elementos están dentro de una figura u otra; es decir, quienes son parte de cada conglomerado.

Con estructuras más sencillas, se encuentran los algoritmos basados en aproximación geométrica (primitivas geométricas), usados principalmente en problemas de graficación. Algunos de ellos hacen uso de las estructuras básicas para encontrar datos proyectados en un plano 2D o 3D, las primitivas geométricas que forman líneas abiertas o cerradas (formando polígonos) para determinar un área. El uso de algoritmos de visualización de datos encuentra conglomerados, cuando reduce los datos de alta dimensión a 2D, visualmente se pueden encontrar éstas estructuras. Sin embargo, por si sola la proyección no presenta la información separada por cúmulos. Por lo que se requiere aplicar un algoritmo de separación de datos en un plano de 2D. A continuación se presentan los algoritmos que separan los datos en conglomerados. Funcionan como complemento para un algoritmo de proyección de datos en 2D. Se describe rápidamente la estructura básica ”poli-línea”de primitivas geométricas, y el algoritmo de Yanchang y Junde (2001) basado en la densidad de los datos delimitados por isolíneas para encontrar conglomerados.

II.3.1. Primitivas geométricas

Las primeras herramientas de primitivas geométricas eran de propósito general. Poco a poco se han convertido en herramientas básicas en la graficación (figuras esenciales para identificar objetos). Por medio de líneas o curvas se comienza a trazar el área de una imagen hasta formar la figura correspondiente. En Rockwood (1997) se revisan diferentes tipos de elementos geométricos: líneas y poli-líneas, arcos, elipses y polígonos.

- ***Líneas y poli-líneas.*** La línea es la estructura más frecuente y básica formada por dos movimientos (posicionar coordenadas y trazar línea). Las poli-líneas son la extensión de las líneas que marcan una trayectoria.

- ***Arcos y elipses.*** Los arcos y elipses requieren las coordenadas del centro, radio (mayor y menor en elipses) y el ángulo de inicio y final. Y a partir de esto se especifica la trayectoria en el plano buscando la figura.

- ***Polígonos.*** Encierran un conjunto de poli-líneas, la última línea termina donde comienza la primera hasta formar un polígono. Ésta es una de las estructuras más usadas que se asocia a una superficie.

- ***Triángulo.*** Es el polígono más popular, rápido y utilizado en la mayoría de los métodos de graficación.

La mayoría de las estructuras descritas identifican el área y delimitan el objeto por medio de los parámetros requeridos.

II.3.2. Algoritmo de conglomerado GDILC

Encontrar conglomerados en un conjunto de datos consiste en maximizar la distancia entre los conglomerados y minimiza la que existe dentro cada uno. Yanchang y Junde (2001) visualizan los conglomerados como: el descubrimiento de las regiones más densas en el espacio de datos, donde cada región densa es un conglomerado. Por lo que diseñan una función de densidad para calcular grupos; redefinen el algoritmo llamado GDILC (Grid-Based Density Isoline Clustering). Que se aplica a datos normalizados en el rango $[0, 1]$ como requerimiento de la función que calcula la distancia. El algoritmo funciona con dos procesos: calcular la densidad de cada dato y combinar las regiones vecinas más densas para formar los conglomerados.

La idea es encontrar los conglomerados usando isolíneas de densidad para identificar el contorno de las figuras (Figura 4). Una ventaja del algoritmo es que ahorra tiempo computacional, porque calcula la función de densidad solo para los datos vecinos. El algoritmo para identificar conglomerados se define por los siguientes pasos:

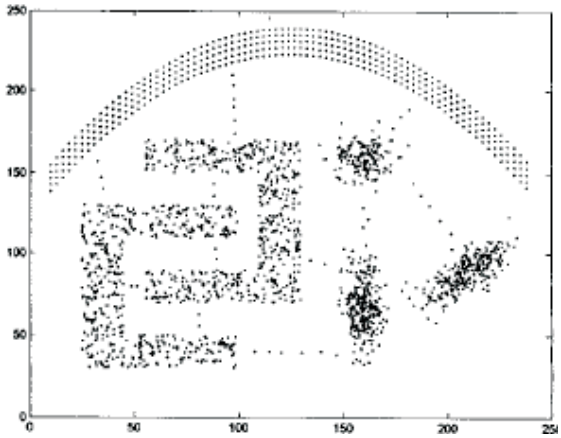


Figura 4. Separación de conglomerado con el algoritmo CURE-NS basado en isolíneas de densidad.

- Inicializar celdas.
  Dividir el espacio de proyección en m celdas con el mismo intervalo, de acuerdo a las coordenadas. Cada dimensión del conjunto de datos se divide en el mismo número de celdas (vecindario). Por ejemplo si los datos están en un espacio 2D, se generan m celdas para las coordenas x e y.

- Calcular la distancia umbral RT.
  Para cada dato (α), calcular la distancia que hay entre ella y cada dato del vecindario al que pertenece ($C\alpha$). Obtener la distancia de cada elemento y la promedio.

- Calcular el vector de densidad y el umbral de densidad DT. Para cada elemento α contabilizar el número de elementos vecinos y establecerlo como su umbral de densidad. Calcular la densidad umbral DT como la densidad promedio de todos los datos.

- Agrupar automáticamente.
  Primero considerar que cada elemento α con densidad mayor a DT es un conglomerado. Revisar los elementos del vecindario $C\alpha$ y juntar con aquellos cuya distancia es menor a RT. Continuar juntando los conglomerados cuando la densidad sea mayor a DT y la distancia menor a RT.

- Remover ruido.
  Remover los conglomerados menores a cierta cantidad indicada.

La función de densidad está definida por:

$$Density(\alpha) = tamaño(\{\beta | Dist(\alpha, \beta) \leq T\}) \tag{15}$$

Donde T es el umbral definido, y $dist(\alpha, \beta)$ es la función que mide la similitud entre dos elementos α y β definida como,

$$Dist_\lambda(\alpha, \beta) = \Big[\sum_{i=1}^{d} |\alpha_i - \beta_i|^\lambda \Big]^{1/\lambda} \tag{16}$$

λ es un entero positivo, y cuando es igual a dos la distancia es la *Euclideana*.

La isolínea de densidad óptima requiere de un proceso, en donde el cambio de la distancia umbral interviene; existen diferentes condiciones extremas que pueden producir resultados no deseados. Por ejemplo que la distancia mayor sea igual a 1, entonces todos los subgrupos se agruparán en un mismo conglomerado, o que existan distancias mínimas que permiten se generen demasiados conglomerados. Por esto la distancia se obtiene bajo la siguiente condición,

$$\text{mín}(Dist) < RT < max(Dist) \tag{17}$$

Entonces el cambio de la distancia RT delimitará la distribución de la densidad tan amplia y separada como sea posible. Así, si la clase es muy densa la isolínea de densidad describirá la distribución de forma adecuada. Los parámetros RT y DT se calculan dinámicamente de acuerdo al número de elementos en el conglomerado. La distancia umbral se calcula con

$$RT = \frac{media(Dist)}{d \times coefFT}$$

$Dist$ es el vector de todas las distancias, d es la dimensión de los datos y $coefRT$ es un coeficiente ajustable. Cuando el resultado de conglomerados no es satisfactorio el $CoefFT$ se ajusta para obtener mejores resultados. Otro parámetro importante en la formación de conglomerados es la densidad umbral (DT). Si es muy pequeña, los conglomerados de un vecindario pueden ser considerados como un solo grupo. Por el contrario, si el valor es muy grande los elementos que se deberían de juntar los va a separar. Por lo que se definió un criterio,

$$DT = \begin{cases} 2 & n < 1000, \\ \dfrac{media(Densidad)}{\log_{10}(n)} \times coefDT & n \geq 1000. \end{cases}$$

II.4. Definición del problema

Una tarea común en los repositorios de documentos es la separación del texto por áreas temáticas; una forma de clasificar documentos es accediendo a las palabras claves y/o la categorización que el autor propone. Sin estas características la tarea de clasificación se hace difícil y tardada, por lo que, la automatización es una condición ideal.

El trabajo de investigación de esta tesis es en el área de minería de texto para visualizar la agrupación y características de dispersión de un conjunto de documentos de diferentes tópicos. Sin embargo por si sola la proyección de datos no presenta la información separada por cúmulos. Por lo que se requiere aplicar un algoritmo de separación de datos en un plano de 2D. Nuestra propuesta de tesis consiste en generar dos algoritmos, uno para obtener las estructuras de relación de documentos (visualizarlas) y otro para separarlas de forma automatizada en una lista que también identifique los VA.

El algoritmo de visualización de datos debe reducir la dimensión de los mismos y reflejar la separación de las clases y la dispersión de los documentos en ellas (proporciona una

idea sobre la similitud de los documentos). Además debe enfatizar las diferentes estructuras que forman a un conglomerado; por ejemplo, subgrupos, VA o pares de documentos separados de la mayor concentración de documentos.

El trabajo con documentos se realiza transformando el texto a un espacio vectorial; por un proceso de contabilización de datos que ha sido estudiado en diferentes aplicaciones estadísticas. Kabán y Girolami (2001) lo consideran en su trabajo de visualización de documentos en el que obtienen una representación para los documentos usando la familia de distribución exponencial. Considerando el manejo especial de la contabilización de ceros en la teoría de ZIP, se propone su uso en un algoritmo de mezcla probabilística de estructuras latentes.

Función de distribución

Antes de desarrollar el modelo, se presenta el resultado de un análisis previo sobre la distribución de los datos. De forma empírica se analiza el comportamiento de los datos y algunas funciones de distribución, para saber si la selección de la función de densidad ZIP representa los datos. Se obtuvo la distribución de probabilidad empírica de una palabra (renglón i de la matriz de frecuencia) del conjunto de datos 20-Newsgroup, formado por 300 documentos de las clases com.sus.ibm.pc.hardware, comp.sys.mac.pc.hardware y sci.med, que se explicará a mayor profundidad en el capítulo de diseño de experimentos.

El experimento consistió en obtener la función de densidad en el vector de frecuencias de los 300 documentos para la palabra "*system*". Se consideró trabajar con tres funciones de distribución de contabilización de datos: Binomial, Poisson y ZIP; los parámetros de cada una fueron estimados por la función de máxima verosimilitud. El modelo Binomial es seleccionado como alternativa del modelo Multinomial, ya que se aplicará la función para un conjunto de datos a un vector. En la Figura 5 se presenta el resultado de la función de densidad empírica; donde el eje de las abscisas representa el número de veces que se repite la palabra en los documentos y el eje de las ordenadas representa el valor de la función de densidad. Como se observa ZIP logra ajustar los datos de forma adecuada, los documentos cuya frecuencia es cero son aproximados adecuadamente por lo que se cree que con esta función se describe mejor a los documentos. A continuación se define la estructura de la mezcla ZIP.

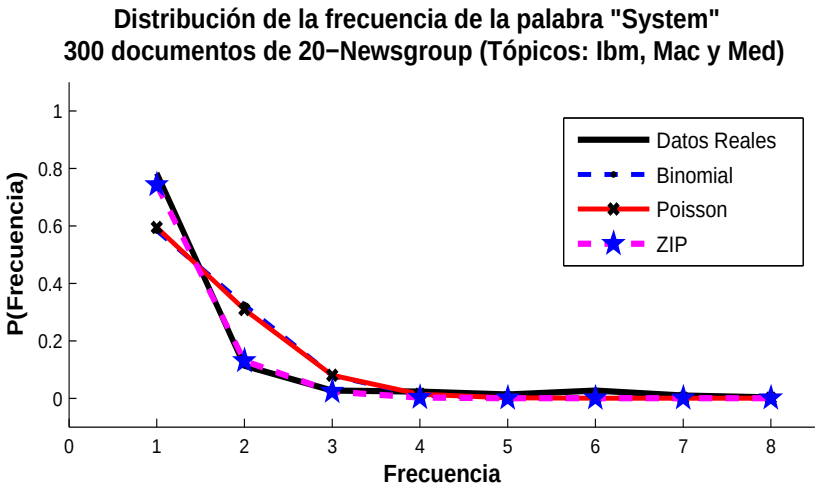


Figura 5. Distribución empírica de una palabra en 300 documentos extraídos de la BD 20-Newsgroup con tópicos: comp.sys.mac.pc.hardware, comp.sys.ibm. pc.hardware y sci.med.

Modelo Zero Inflated Poisson (ZIP)

La distribución de Poisson está diseñada para procesos de contabilización de ocurrencias de un evento, Li y Zha (2004) estiman la distribución de un vector documento con una mezcla de Poisson para clasificar documentos y palabras. Lambert (1992) propone una adaptación de Poisson para modelar los datos considerando el exceso de ceros, y lo llamó ZIP. El modelo supone que el parámetro de probabilidad p modela el exceso de ceros; y con probabilidad $1-p$ cuando provienen de un proceso Poisson. El modelo de regresión ZIP supone que los datos $T = \{t_1, t_2, ..., t_n\}$ son independientes y que,

$$T_n = \begin{cases} 0 \quad \text{con probabilidad} \quad p_n + (1-p_n)e^{-\lambda_n}, & (18) \\ t_n \quad \text{con probabilidad} \quad (1-p_n)\dfrac{e^{-\lambda_n}\lambda_n^{t_n}}{t_n!}, & (19) \end{cases}$$

donde λ_n es la media de la parte de Poisson y p_n es el parámetro de la parte del modelo que controla el exceso de ceros. Ambos parámetros se relacionan con funciones de enlace que cumplen las siguientes condiciones,

$$\begin{aligned} \log(\lambda) &= \mathrm{B}\beta \qquad \mathrm{y} \\ \text{logit}(p) &= \log\left(\frac{p}{1-p}\right) = \mathrm{G}\gamma \end{aligned} \tag{20}$$

donde B y G son matrices de covarianza, y logit es un modelo de regresión logística (McCullagh y Nelder, 1989).

II.4.1. Objetivos

Con base en la característica particular de la matriz de frecuencias; no todas las palabras se encuentran en todos los documentos, y se genera la siguiente hipótesis de trabajo:

"*Los documentos (matriz de frecuencia) se ajustan a la mezcla de probabilidad ZIP, y el algoritmo de visualización GTM con distribución ZIP obtiene una representación de los datos en 2D que muestra las proyecciones en conglomerados.*"

A partir de esto se genera el objetivo general:

Obtener un algoritmo para visualizar los datos en un espacio 2D que permita identificar y separar los conglomerados y VA en bases de datos de gran tamaño y alta dimensionalidad.
Con objetivos particulares:

- Analizar los diferentes algoritmos para visualizar conglomerados e identificar las características de las BD, para encontrar un algoritmo que obtenga una representación de los datos.
- Identificar las características de los métodos robustos para describir bajo qué situaciones y cómo funciona cada uno de ellos en la creación de conglomerados con datos difíciles de manejar (VA).
- Encontrar las condiciones bajo las cuales existe problema en la creación de conglomerados con VA.
- Encontrar una función o estructura de apoyo para manipular los datos difíciles.
- Proponer un algoritmo con capacidad robusta para manipular datos difíciles de separar.
- Proponer un algoritmo de separación de clases e identificar aquellos documentos en los que no se distinga la clase a la que pertenecen.
- Definir criterios de evaluación de los algoritmos propuestos.

- Evaluar el desempeño de los algoritmos propuestos y comparar con modelos del estado del arte.

Para cumplir con los objetivos se propone una metodología para el análisis de conglomerado en documentos y su evaluación por medio de una serie de experimentos que ayudan a aceptar ó rechazar la hipótesis de trabajo.

Capítulo III

Modelo de visualización VL-ZIP

En la representación de documentos (bolsa de palabras), un conjunto de documentos es representado por una matriz de frecuencias T. En la formulación del algoritmo propuesto se utiliza la versión transpuesta con respecto a su definición (ver sección II.1.1). Un elemento de la matriz es representado como t_{nj}, refiriéndose al número de ocurrencias de la palabra j en el documento n. En la formulación del modelo se utiliza la siguiente terminología como reducción de la notación; $\mathbf{t}_n$ es el *n-ésimo* renglón documento (transpuesto) de la matriz T y $\mathbf{t}_j$ es la *j-ésima* columna (*j-ésima* palabra). El vocabulario del conjunto de documentos después del preprocesado de palabras y la selección del conjunto que mejor representa los datos es de dimensión D.

En la bolsa de palabras existe gran cantidad de palabras que no son utilizadas en todos los documentos, por lo que la presencia excesiva de ceros en la matriz de frecuencias es muy común. Lambert (1992) propuso una modificación a la función de Poisson con el modelo ZIP considerando el exceso de ceros en el vector de datos. Considerando el manejo de la contabilización de ceros en la teoría de ZIP, se propone su uso en el modelo de mezclas probabilísticas de clases latentes (algoritmo VL-ZIP). El modelo de visualización de documentos está formado por dos tareas básicas como se muestra en el diagrama de la Figura 6; la primera corresponde a la definición del modelo probabilístico, y la segunda a la visualización de los datos. En la siguiente sección se describe la parte teórica y matemática del modelo probabilístico. Dos aportaciones al modelo son propuestas, la introducción de la función de probabilidad ZIP para representar a los datos y la definición de una nueva estructura latente con características robustas. En la última sección se describen los pasos del algoritmo VL-ZIP.

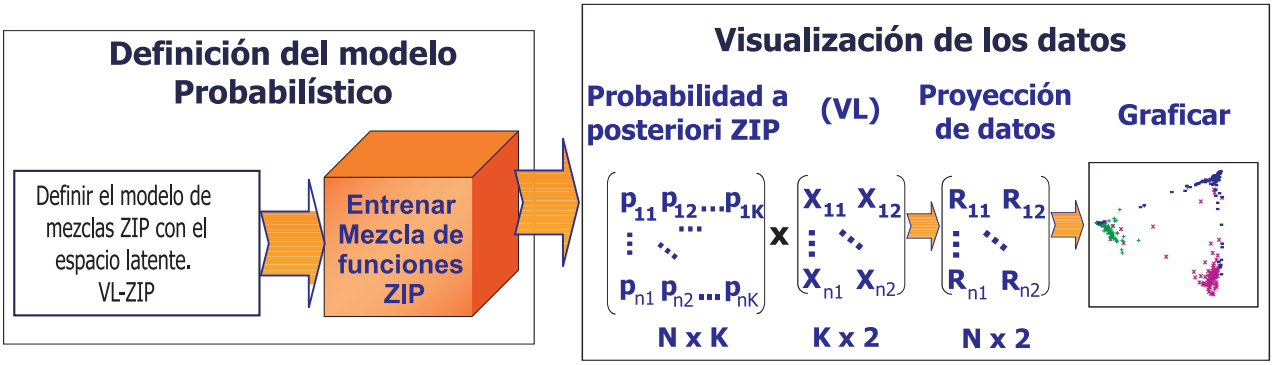


Figura 6. Esquema del modelo de visualización de documentos VL-ZIP, formado por la definición del modelado probabilístico y la visualización de los datos.

III.1. Definición del modelo probabilístico VL-ZIP

En esta etapa se definen la función de probabilidad que modela los datos, a partir de ella se define la estructura probabilística. La representación del modelo de regresión ZIP para el caso multivariado en el proceso de contabilización de datos para t_{nj} se plantea como

$$T_{nj} = \begin{cases} 0 \quad \text{con probabilidad} \quad p_{nj} + (1 - p_{nj})e^{-\lambda_j}, & (21) \\ \mathbf{t}_{nj} \quad \text{con probabilidad} \quad (1 - p_{nj})\dfrac{e^{-\lambda_j}\lambda_j^{t_{nj}}}{t_{nj}!}, & (22) \end{cases}$$

donde λ_j es la media de los elementos de la columna *j-ésima* y p_{nj} es el parámetro del modelo para el exceso de ceros. La representación supone que existe una relación entre λ y las funciones base ϕ_m, evaluadas en las variables latentes (VL) x, a través de la función de enlace $\log(\lambda)$

$$\log(\lambda_j) = \beta + \sum_{m=1}^{M} \phi_m w_{mj}, \tag{23}$$

donde β es el parámetro bias, w_{mj} es un peso aplicado a cada función. Se supone que existen M funciones base, especificadas en la inicialización. Se considera además una función de enlace logística para evaluar p_{nj} con matrices de covarianza G

$$\text{logit}(p_{nj}) = \log \frac{p_{nj}}{1 - p_{nj}} = G_{nj}\gamma_j. \tag{24}$$

Wedel *et al.* (1993) consideran un modelo de regresión latente para la contabilización heterogénea de datos. Suponen que las observaciones fueron generadas por una mezcla

finita de distribuciones Poisson con distribuciones diferidas en las intersecciones y coeficientes de variables explicadas en los componentes de regresión del modelo. Se puede recuperar el modelo de mezcla de Wedel *et al.* (1993) suponiendo como (Bishop *et al.*, 1998) que las variables explicables (latentes) están distribuidas como funciones delta centradas en un conjunto de puntos conocidos (x_k) en una malla de dos dimensiones, relacionadas con el logaritmo de la media (λ) por medio de los parámetros w. Integrando sobre las variables latentes, la distribución de probabilidad para el *n-ésimo* renglón se considera como

$$P_n(\mathbf{t}_n|w,\gamma) = \sum_k \alpha_k f_{nk}(\ \mathbf{t}_n|w,x_k,\gamma), \tag{25}$$

una función para los parámetros w y γ; el modelo está formado por K mezclas Poisson, ZIP donde f_{nk} es la función de densidad de la probabilidad ZIP para $\mathbf{t}_n$ y α_k es del parámetro coeficiente de la mezcla, que se puede interpretar como la probabilidad incondicional de que un individuo pertenezca a la clase k, con la restricción

$$\sum_{k=1}^{K} \alpha_k = 1, \qquad\qquad 0 < \alpha_k < 1. \tag{26}$$

Ahora, suponiendo el modelo de mezclas para datos tipo documento, se considera el espacio latente como en GTM. En la siguiente sub-sección se describe la estructura latente, y después el modelo de mezclas propuesto; posteriormente se describe el proceso de entrenamiento para estimar los parámetros que modelan la función de mezclas.

III.1.1. Modelo de espacio latente

El espacio latente en el GTM es un modelo de regresión lineal generalizado que transforma las variables latentes x a su espacio de datos correspondiente (Bishop *et al.*, 1998):

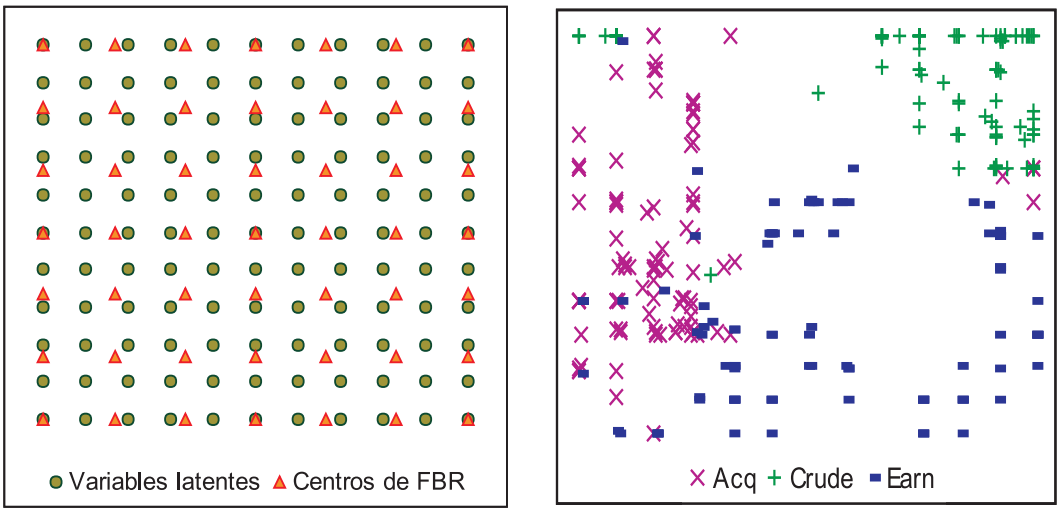
$$\mathbf{t}(\mathbf{x};\mathbf{W}) = \mathbf{W}\Phi(x) + \mathbf{n}, \tag{27}$$

donde **W** es una matriz de $D \times M$ con parámetros desconocidos w_{mj} y Φ es una matriz con la evaluación de funciones base radial en las variables latentes **x**, con ruido **n**. El modelo de funciones base radial (FBR) del GTM es considerado como

$$\phi(\mathbf{x}) = \exp\Big(-\frac{\big\|x-c\big\|^2}{2\sigma^2}\Big),$$

con c centros de FBR y σ es el parámetro de anchura. En la Figura (7) se muestra la representación gráfica del espacio latente, en la Figura 7(a) se presentan las variables latentes con la distribución usualmente empleada en GTM. En la Figura 7(b) se observan las

proyecciones obtenidas con el algoritmo original GTM de la BD Reuters. En la Figura 7(b) se observa que en la proyección del espacio latente obtenida con el GTM, las diferentes clases no pueden ser representadas de forma separada. En el algoritmo que se propone supone un espacio latente en 2D con K puntos latentes $x_1, ..., x_K$, con una distribución latente como la mezcla de funciones delta localizadas en los puntos latentes. Las variables latentes en el esquema propuesto aparecen en función $\log \lambda$ en el modelo ZIP.



(a) (b)

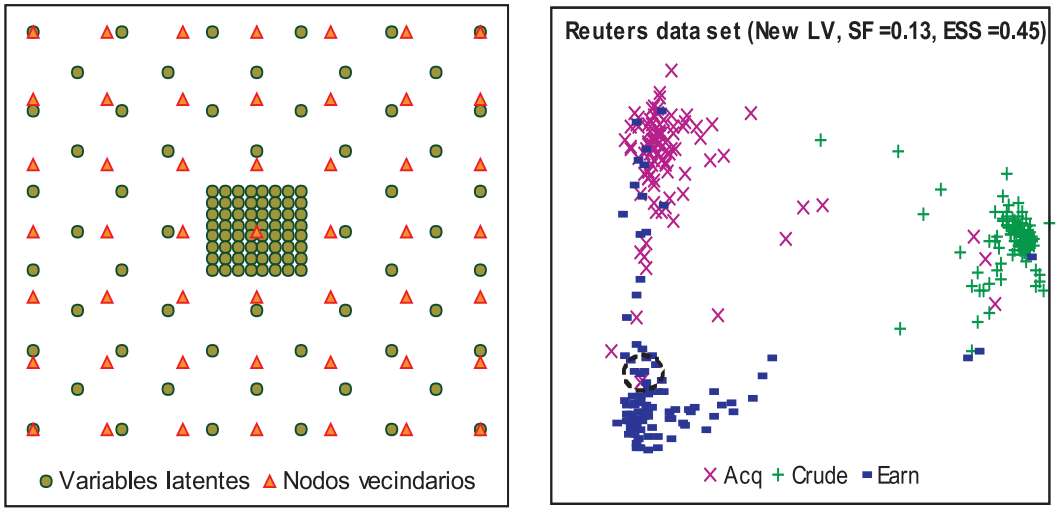
Figura 7. Esquema latente; a) Representación de variables latentes; b) Proyección de datos en el espacio de variables latentes con un subconjunto de la BD Reuters.

Continuando con la representación de los datos, se propone extender el área de proyección de las variables latentes. En la que se consideran dos áreas diferentes: una con distribución uniforme sobre el intervalo $[-1, 1]$ y la otra con mayor dispersión entre los puntos y límites de $[-5, 5]$ (ver Figura 8a). Además de ampliar el espacio de proyección, se considera el uso de la función tangente hiperbólica de la siguiente forma,

$$\phi(x) = \tanh(\mathbf{x}_k^t \mathbf{c}_m). \tag{28}$$

con $\mathbf{x}^t$ como el vector transpuesto $\mathbf{x}$. En lugar de utilizar centros de FBR se incluyen nodos vecindarios c como parámetros base. La relación entre las variables latentes y los nodos vecindarios es la función de producto punto, que indica la similitud entre ellos. El objetivo de la dispersión propuesta es obtener regiones con alta probabilidad de ubicar una clase, se obtuvo como resultado de la búsqueda de mayor separación entre las clases para evitar tener objetos en el centro del plano. Los conglomerados deben aparecer cerca de

los bordes de la malla latente. Con un valor negativo la función indica que un nodo vecindario y una variable latente se encuentran en posición opuesta, con ángulo mayor a $90°$ (se considera que esta condición es un aspecto de penalización natural). En la Figura 8(a) se muestra la nueva representación de las variables latentes. En la Figura 8(c) se ve la proyección obtenida con el algoritmo VL-ZIP y las variables latentes propuestas y los mismos datos, mostrando una mejor representación en conglomerados.



(a) (b)

Figura 8. Nuevas variables latentes y distribución de funciones base, a) distribución propuesta para las variables latentes, con el modelo propuesto VL-ZIP, para los datos Reuters.

III.1.2. Definición del modelo de mezclas VL-ZIP

La función de distribución de Poisson puede tener una cola larga, o la media puede ser muy cercana al cero, proporcionando cierto tipo de robustez. El modelo VL-ZIP tiene esta característica especial de la distribución de Poisson y también proporciona la habilidad de trabajar con la contabilización del cero como una porción de la función. En el marco de trabajo se considera el uso de clases latentes con el modelo de regresión de Poisson (Wedel *et al.*, 1993), formulando el modelo ZIP en lugar de Poisson. La probabilidad de que ocurra una palabra en ZIP está dada por el estado cero (21) ó el estado Poisson sin ceros (22). La mezcla del modelo ZIP será considerada como (25), donde el modelo del espacio latente es representado por la función de liga (23).

A partir del modelo definido la proyección se obtiene con la relación de la respuesta de

la función de probabilidad y el espacio latente. Para evaluar la función se estiman los parámetros W, γ y β con los datos $\mathbf{t}_n$ y el espacio latente descrito por ϕ. Primero se define la función de verosimilitud completa con (25),

$$\mathcal{L}(W,\gamma,\beta) = \prod_{n=1}^{N}\left(\sum_{k=1}^{K}\alpha_k f_{nk}(\,\mathbf{t}_n|\mathbf{W},\gamma,\beta)\right), \tag{29}$$

y se resuelve para estimar los parámetros que maximizan el logaritmo de la verosimilitud, con el algoritmo EM (Dempster *et al.*, 1976). El primer paso es calcular la Esperanza de la verosimilitud relativa (29) de la relación entre los parámetros anteriores y los nuevos, de la siguiente forma:

$$\begin{aligned} E(\log\mathcal{L}) &= \sum_{n=1}^{N}\sum_{k=1}^{K} f^{ant}(\mathbf{W},\alpha_k,\gamma,\beta|\mathbf{t}_n)\log\left[\alpha_k f_{nk}(\,\mathbf{t}_n|\mathbf{W},\gamma,\beta)\right] \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\theta_{nk}\log f_{nk}(\,\mathbf{t}_n|\mathbf{W},\gamma,\beta) + \sum_{n=1}^{N}\sum_{k=1}^{K}\theta_{nk}\log\alpha_k \end{aligned} \tag{30}$$

donde θ_{nk} es la esperanza actual (relación con los parámetros anteriores $\{^{ant}\}$), la probabilidad de que $\mathbf{t}_n$ pertenezca al componente k de la mezcla. Es un componente de Θ y se evalúa por medio de la regla de Bayes,

$$\theta_{nk} = f^{ant}(\mathbf{W},\alpha_k,\gamma,\beta|\mathbf{t}_n) = \frac{\alpha_k f_{nk}(\,\mathbf{t}_n|\mathbf{W},\gamma)}{\sum_{k=1}^{K}\alpha_k f_{nk}(\,\mathbf{t}_n|\mathbf{W},\gamma)}. \tag{31}$$

Considerando todos los elementos del renglón $\mathbf{t}_n$, la función esperanza a maximizar con respecto a los parámetros γ, β y w es;

$$\begin{aligned} E(\log\mathcal{L}) &= \sum_{\substack{n,k,j\\ t_{nj}=0}}^{NKD}\theta_{nk}\log(e^{G_{nj}\gamma_j}+e^{-\lambda_{kj}}) - \sum_{\substack{n,k,j\\ t_{nj}=0}}^{NKD}\theta_{nk}\log(1+e^{G_{nj}\gamma_j}) \\ &\quad - \sum_{\substack{n,k,j\\ t_{nj}>0}}^{N,K,D}\theta_{nk}\log(1+e^{G_{nj}\gamma_j}) + \sum_{\substack{n,k,j\\ t_{nj}>0}}^{N,K,D}\theta_{nk}\left(t_{nj}\log(\lambda_{kj})-\lambda_{kj}\right) \\ &\quad - \sum_{\substack{n,k,j\\ t_{nj}>0}}^{N,K,D}\theta_{nk}\log(t_{nj}!) + D\sum_{n,k}^{N,K}\theta_{nk}\log(\alpha_k). \end{aligned} \tag{32}$$

con $\log(\lambda_{kj}) = \beta_k + \sum_{m=1}^{M} w_{mj}\phi_m(x_k)$.

Lambert (1992) propuso una mejor representación que se puede obtener considerando utilizar una variable indicadora z; tal que, $z_{nj} = 1$ cuando $\mathbf{t}_n$ proviene de cero-inflado, y

$z_{nj} = 0$ cuando proviene de una mezcla de Poisson. Al añadir la variable, el logaritmo de la verosimilitud está dado por,

$$\begin{aligned} E(\log \mathcal{L}) &= \sum_{n,k,j}^{N,K,D} z_{nj}\theta_{nk}\mathbf{G}_{nj}\gamma_j - \sum_{n,k,j}^{N,K,D} \theta_{nk}\mathbf{log}(1+e^{G_{nj}\gamma_j}) \\ &+ \sum_{n,k,j}^{N,K,D} (1-z_{nj})\theta_{nk}\Big(t_{nj}\log(\lambda_{kj}) - \lambda_{kj}\Big) \\ &- \sum_{n,k,j}^{N,K,D} (1-z_{nj})\theta_{nk}\ \log(t_{nj}!) + D\sum_{n,k}^{N,K}\theta_{nk}\log(\alpha_k). \end{aligned} \tag{33}$$

La esperanza de la variable indicadora z se estima con la media condicional posterior en $\mathbf{t}_{nj}$ (Dobson, 2002) como,

$$z_{nj} = \begin{cases} (1+e^{-G_{nj}\gamma_j}e^{-\lambda_{kj}})^{-1} & \text{Si } \mathbf{t}_{nj} = 0 \text{ (estado cero)} \\ 0 & \text{Si } \mathbf{t}_{nj} > 0 \end{cases} \tag{34}$$

La solución aproxima los parámetros desconocidos γ, β, **W** y α con el paso M, que corresponde a maximizar (33).

III.1.3. Entrenamiento: Paso M

En Bishop *et al.* (1998) se proponen dos pasos básicos para proyectar los datos en un plano de 2D, como se muestra en la Figura 6. Después de definir el modelo probabilístico, en esta sub-sección se describe el entrenamiento que le corresponde al paso M del algoritmo EM. En las siguientes líneas se presenta el desarrollo de maximizar la ecuación (33) con respecto a γ, β, **W** y α.

Estimación de γ

A partir de la esperanza de $\mathcal{L}(\gamma, \mathbf{t}, Z)$ (33), se maximiza con respecto al parámetro γ aplicando el algoritmo EM con un procedimiento de regresión logística (McCullagh y Nelder, 1989). La primera y segunda derivada son:

$$\begin{aligned} \frac{\partial E(\log \mathcal{L}_c)}{\partial \gamma_{j'}} &= \sum_n \sum_k \theta_{nk} G_{nj'}(z_{nj'} - p_{nj'}) \\ \frac{\partial^2 E(\log \mathcal{L}_c)}{\partial \gamma_{j'}^2} &= \sum_n \sum_k \theta_{nk} G_{nj'}^2(p_{nj'}[1-p_{nj'}]) \end{aligned}$$

Observando que las derivadas cruzadas son iguales a cero. Se obtienen los parámetros por medio de un proceso iterativo con el algoritmo de Newton-Raphson:

$$\gamma_k^{nvo} = \gamma_k^{ant} - \eta_1 \frac{\partial E}{\partial \gamma_k} / \frac{\partial^2 E}{\partial \gamma_k^2}, \tag{35}$$

Con η_1 pequeño, y compromete la rapidez de la convergencia con la eficiencia de encontrar un máximo.

Cálculo de los parámetros β y los elementos de W

Estos parámetros son calculados de forma iterativa con el algoritmo de Newton-Raphson, de la siguiente forma,

$$w_{jk}^{nvo} = w_{jk}^{ant} - \eta_2 \frac{\partial E}{\partial w_{jk}} / \frac{\partial^2 E}{\partial w_{jk}^2}, \tag{36}$$

y

$$\beta_k^{nvo} = \beta_k^{ant} - \eta_2 \frac{\partial E}{\partial \beta_k} / \frac{\partial^2 E}{\partial \beta_k^2}, \tag{37}$$

El parámetro η_2 limita la rapidez de la convergencia para favorecer un valor óptimo. La primer y segunda derivada para calcular w son:

$$\begin{aligned} \frac{\partial E(\log \mathcal{L})}{\partial w_{m'j'}} &= \sum_n (1 - z_{nj'}) \sum_k \phi_{km'} \theta_{nk} \Big(t_{nj'} - \lambda_{kj'} \Big) \\ \frac{\partial^2 E(\log \mathcal{L})}{\partial w_{m'j'}^2} &= -\sum_n (1 - z_{nj'}) \sum_k \phi_{km'}^2 \theta_{nk} \lambda_{kj'}. \end{aligned}$$

Y para β las derivadas son:

$$\begin{aligned} \frac{\partial E(\log \mathcal{L})}{\partial \beta_{k'}} &= \sum_{nj} (1 - z_{nj}) \theta_{nk'} \Big(t_{nj} - \lambda_{k'j} \Big) \\ \frac{\partial^2 E(\log \mathcal{L})}{\partial \beta_{k'}^2} &= -\sum_{nj} (1 - z_{nj}) \theta_{nk'} \lambda_{k'j}. \end{aligned}$$

La actualización de los parámetros anteriores se realiza en un conjunto de iteraciones. Pero el primer valor de **W** antes de entrar al ciclo de entrenamiento se obtiene con la esperanza del logaritmo de la verosimilitud de $\mathcal{L}_+$, considerando la parte positiva que le corresponde a Poisson de la siguiente forma:

$$\begin{aligned} E(\log \mathcal{L}_+(\mathbf{W}, \beta)) &= \sum_{n,k,j}^{N,K,D} \theta_{nk} \left(t_{nj} \left(\beta_k + \sum_m^M \phi_{km} w_{mj} \right) - \lambda_{kj} \right) \\ &\quad - \sum_{n,k,j}^{N,K,D} \theta_{nk} log(1 - e^{-\lambda_{kj}}) - \sum_{n,k,j}^{N,K,D} \theta_{nk} \log(t_{nj}!). \end{aligned} \tag{38}$$

La inicialización de w_{mj} utiliza el algoritmo de Newton-Raphson, donde la primera y segunda derivadas son:

$$\begin{aligned}\frac{\partial E(\log \mathcal{L}_+)}{\partial w_{m'j'}} &= \sum_{nk} \phi_{km'}\theta_{nk}\left(t_{nj'} - \frac{\lambda_{kj'}}{1-e^{-\lambda_{kj'}}}\right)\\ \frac{\partial^2 E(\log \mathcal{L}_+)}{\partial w^2_{m'j'}} &= -\sum_{nk} \phi^2_{km'}\theta_{nk}\lambda_{kj'}\frac{[1-e^{-\lambda_{kj'}}(1+\lambda_{kj'})]}{(1-e^{-\lambda_{kj'}})^2}\end{aligned}$$

Observese que, las derivadas cruzadas también son cero en este caso. La evaluación con el algoritmo de Newton-Raphson está dada por,

$$w^{nvo}_{jk} = w^{ant}_{jk} - \eta_2 \frac{\partial E(\log \mathcal{L}_+)}{\partial w_{mj}} \Big/ \frac{\partial^2 E(\log \mathcal{L}_+)}{\partial w^2_{mj}}. \tag{39}$$

La actualización del parámetro β es similar a la de w_{mj}, con

$$\begin{aligned}\frac{\partial E(\log \mathcal{L}_+)}{\partial \beta_{k'}} &= \sum_{nj} \theta_{nk'}\left(t_{nj} - \frac{\lambda_{k'j}}{1-e^{-\lambda_{k'j}}}\right)\\ \frac{\partial^2 E(\log \mathcal{L}_+)}{\partial \beta^2_{k'}} &= -\sum_{nj} \theta_{nk'}\frac{\lambda_{k'j}[1-e^{-\lambda_{k'j}}(1+\lambda_{k'j})]}{(1-e^{-\lambda_{k'j}})^2}\end{aligned}$$

y la actualización con Newton-Raphson esta dada por:

$$\beta^{nvo}_{k} = \beta^{ant}_{k} - \eta_2 \frac{\partial E(\log \mathcal{L}_+)}{\partial \beta_{k}} \Big/ \frac{\partial^2 E(\log \mathcal{L}_+)}{\partial \beta^2_{k}}, \tag{40}$$

con valor η_2 igual que para w.

Cálculo de α

Como la suma de α_k debe de ser igual a 1, se maximiza (33) con respecto a α_k como la función aumentada $\sum_{n,k} \theta_{nk} \log \alpha_k - \mu(\sum_k \alpha_k - 1)$, donde μ es un multiplicador Lagrangiano. Entonces la actualización se define como,

$$\alpha^{nvo}_{k} = \frac{1}{ND}\sum_{n} \theta_{nk}, \tag{41}$$

donde N es el número de documentos y D su dimensión.

III.2. Algoritmo computacional

La construcción del algoritmo computacional de visualización de documentos (VL-ZIP) se divide en tres etapas: inicializar variables, entrenar y proyectar los datos, detallados en el Algoritmo 3.

Algoritmo 3 Algoritmo de variables latentes VL-ZIP.

1. **Inicializar variables.**

   - Valores fijos.
     - Calcular $\mathbf{W^{(0)}}$ como en (39).
     - $G_{nj} = \begin{cases} \frac{1}{D-n_0} & \text{Si } t_{nj} = 0 \\ \frac{n_+}{D-n_+} & \text{de cualquier otra forma.} \end{cases}$
     - $\gamma_j^{(0)} = \mathcal{N}(0, 0{,}1)$.
     - $\alpha_k^{(0)} = 1/K$.
     - Calcular $\beta^{(0)}$ como en (40).
   - Valores a Calibrar.
     - X = nuevas variables latentes con dispersión $[-5, 5]$ (descritas en III.1.1).
     - c = vector en dos dimensiónes (coordenadas).
     - η_1
     - η_2

2. **Entrenamiento, iterar hasta que converja:**

   - Paso E:
     - Calcular Θ de (31).
   - Paso M: Actualización de parámetros.
     - Calcular γ^{nvo} con (35)
     - Calcular w^{nvo} con (36)
     - Calcular β^{nvo} usando (37)
     - Calcular α^{nvo} de (41)

3. **Proyección de datos.**

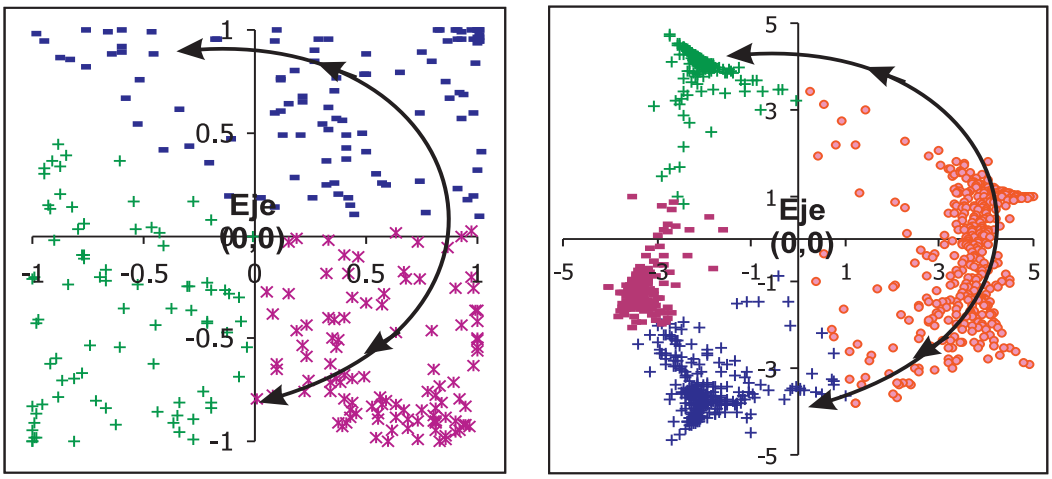
   - Calcular $R = \Theta X$
   - Graficar R

El segundo paso del Algoritmo 3 corresponde al primer módulo del diagrama de la Figura 6 y la tercer etapa corresponde al segundo módulo (visualización de los datos). Donde n_+ es el número de elementos $\mathbf{t}_{nj}$ diferentes de cero, y n_0 es el número de elementos iguales a cero. **W** se inicializa con (39) y β con (40). El número de iteraciones lo determina el cambio en $E(\mathcal{L})$ cuando es más bajo que cierto valor predeterminado. La visualización de documentos en 2D a considerar es la estructura latente, que presenta un mapa con la reducción $R = \Theta X$, donde Θ es la matriz de probabilidad a posteriori de cada documento (31) y X son las variables latentes inherentes en la formulación ZIP (25). Aunque se utiliza la contabilización de ceros, la matriz de datos es el resultado del proceso de reducción de la dimensión a través del criterio de selección de palabras descrito en el capítulo de minería de texto.

Capítulo IV

Algoritmo de separación de clases

En este capítulo se propone un algoritmo simple para separar los datos proyectados en un plano de 2D acceder a ellos por medio de una lista. En la mayoría de los resultados de proyección se encuentra una característica en la formación de conglomerados. Los datos obtenidos con el algoritmo de visualización Multinomial (Kabán y Girolami, 2001) y el LV-ZIP definido en el capitulo anterior distribuyen los datos alrededor del origen del plano cartesiano como se muestra en la Figura 9.



(a) (b)

Figura 9. Representación gráfica de los conglomerados alrededor de las coordenadas $(0, 0)$ del plano cartesiano. Las líneas indican la posible trayectoria de los datos.

A partir de esta observación, se desarrolló un algoritmo que separa los datos por medio de primitivas geométricas (líneas) y un parámetro de densidad que indica el centro de un

conglomerado. La estructura es simple, primero se marcan líneas que inician en $(0,5)$ y se enumeran a favor de las manecillas del reloj. Entre dos líneas existen $g°$ de separación, con $g = [3,4,5]$ y magnitud de 5 como se muestra en la Figura 10. Al espacio entre cada línea se llama $Rango_n$, la estructura es similar a la mostrada en la Tabla I. A continuación, para cada dato se calcula el ángulo con respecto al vector inicio (coordenada(0,5)). Se calcula la densidad de cada región ($g°$), actualmente se contabiliza el número de documentos que se encuentran en cada intervalo (en un futuro se implementará una función de densidad).

Tabla I. Estructura de la matriz $Rango$.

Angulo		
Inicio	Fin	Densidad
0	5	5
5	10	70
...	...	...
355	360	1

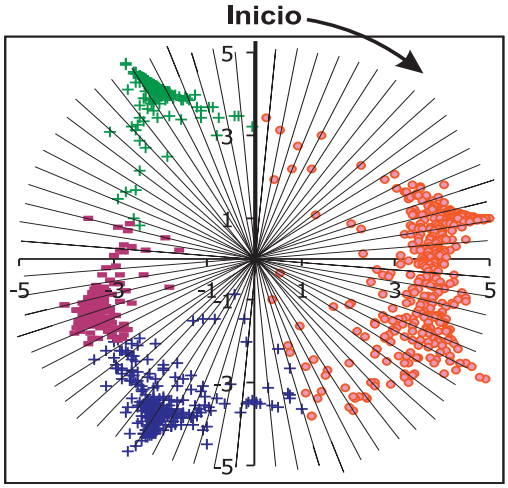


Figura 10. Proyección de líneas de separación (origen en $(0,0)$) y $g°$ de separación entre ellas; en el eje positivo de las ordenadas inicia la secuencia.

Se identifican las regiones con mayor y menor densidad; es decir los centros de los K conglomerados $\{Rango\_c1, Rango\_c2, ..., Rango\_cK\}$ iniciales (Tabla II) y las Q separaciones iniciales $\{Rang\_s1, Rango\_s2, ..., Rango\_sQ\}$ (Tabla II). En este punto se encuentran los primeros K posibles conglomerados y las Q separaciones (Figura 11). Se considera que un

$Rango_n$ es mayor si el número de elementos (frecuencia) en él es mayor a $Dmax$ y es menor si contiene menos de $Dmin$ documentos,

$$Dmax = max(Rango\_n) * pmax, \tag{42}$$

$$Dmin = max(Rango\_n) * pmin, \tag{43}$$

$$Dmin2 = max(Rango\_n) * pmin2, \tag{44}$$

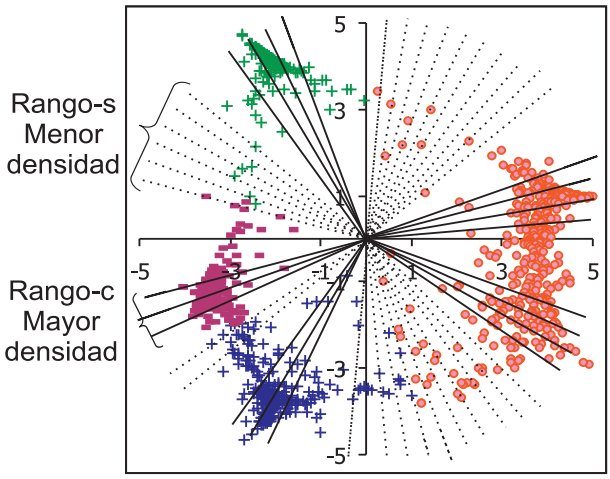


Figura 11. Rangos con mayor ($Rango\_c$) y menor ($Rango\_s$) densidad; el primero identifica un posible centro de conglomerado y el segundo la separación entre clases.

donde $max(.)$ es la función que encuentra el valor máximo, $pmax$ y $pmin$ son la máxima/mínima porción del valor máximo. La proporción de $pmax$ oscila entre $[60, 70]\,\%$ del valor máximo de $Rango_n$ y $pmin$ entre $[15, 17]\,\%$ y $pmin2$ entre $[18, 19]\,\%$.

Tabla II. Estructura del elemento $Rango\_c$.

	Angulo	
Conglomerado	Inicio	Fin
1	70	85
2	110	125
3	205	215
4	245	255
5	325	340

Una vez que se obtuvieron los primeros centros y separaciones se reajustan los espacios de separación. Si existen dos centros consecutivos con separación mayor a $15°$ ($Rango\_c_{i+1}\{inicio\} - Rango\_c_i\{fin\} > 15$) y no hay separación en ese $Rango$, se identifica uno cuya densidad sea $< Dmin2$ y se agrega una nueva separación. En este punto el algoritmo revisa el espacio entre dos centros y, si existe una región menos densa $< Dmin2$ crea una nueva.

Tabla III. Estructura del elemento $Rango\_s$.

	Angulo			
Separación	Inicio	Fin	Inicio2	Fin2
1	345	360	0	70
2	145	190		
3	265	325		

El ejemplo numérico se representa en las Tablas (II y III), donde los $Rango\_c$ con índice 2 y 3 poseen suficiente separación como para ser considerados dos conglomerados y un elemento $Rango\_c_i\{fin\} < Rango_n < Rango\_c_{i+1}\{inicio\}$ posee menor densidad a $Dmin2$; un ejemplo de los supuestos valores para $Rango\_s$ se presenta en la Tabla IV. En este punto los espacios de separación marcan el inicio y fin de los conglomerados.

Tabla IV. Estructura del elemento $Rango\_s$ después del reajuste de separación.

	Angulo			
Separación	Inicio	Fin	Inicio2	Fin2
1	345	360	0	70
2	145	190		
3	230	235		
4	265	325		

A continuación se forma la matriz de rangos de conglomerados: $Rango\_c_k\{inicio\} = Rango\_s_k\{fin\}$ y $Rango\_c_k\{fin\} = Rango\_s_{k+1}\{inicio\}$ Tabla V); se ajustan los índices del inicio y el final de la matriz. A partir de la matriz $Rango\_c$ se forman los conglomerados con los documentos cuyos ángulos pertenecen al rango marcado ($doc_i \in clase\_k$; si y solo si $Rango\_c_k\{inicio\} < ang\_doc_i < Rango\_c_k\{fin\}$). Todos los documentos con ángulo dentro de $Rango\_s$ son considerados VA; para graficar se pueden añadir a un conglomerado o

Tabla V. Separación de conglomerados, $Rango\_c$.

	Angulo	
Separación	Inicio	Fin
1	325	345
2	70	145
3	190	230
4	235	265

identificar como VA.

El resumen se presenta en el Algoritmo 4.

El algoritmo de separación de clases se complementa con una fase que se llama *re-entrenamiento*, vuelve a aplicar el algoritmo de visualización y las clases encontradas con la separación. A esas clases se les vuelve a aplicar el algoritmo de separación para identificar otras estructuras de conglomerados. Si en el primer conjunto de datos proyectado hubo algunas clases traslapadas, el proceso de re-entrenamiento las reubica en su clase; se cree que al eliminar documentos de diferente clase que poseen palabras similares a las de los documentos traslapados se puede lograr la separación de éstos.

Algunos conceptos utilizados se definen a continuación; los primeros resultados del algoritmo de separación se llaman *Clases* y equivalen a un conglomerado. Tentativamente los elementos que se encuentran en el espacio de separación de clase se denominan VA. El volver a aplicar el algoritmo de visualización con las clases encontradas se denomina *re-entrenamiento*. Los resultados de la separación de los datos re-entrenados se llaman *S-clase*. El re-entrenamiento de los datos puede ser de dos formas,

- *Cada dos clases*. Con esta modalidad se quiere ubicar pequeñas aglomeraciones localizadas entre dos clases, de tal forma que el algoritmo se concentra sólo en los documentos de dos clases. El re-entrenamiento se realiza con pares de clases encontradas, {Clase 1, Clase 2, . . ., Clase K} y los elementos entre las clases (VA). Los pares se forman de la siguiente forma: {Clase 1, Clase 2}, {Clase 2, Clase 3} y el último par de clases corresponde a {Clase 1, Clase K}; esto es para unir las clases como una estructura cerrada (elipse). Después, con los datos proyectados vuelve a aplicar el algoritmo de separación para encontrar las estructuras fuertes de la dos clases (S-clase1,

Algoritmo 4 Algoritmo de separación de clases.

1. Definir g.
2. Formar matriz $Rango$.
3. Para cada documento,
   a) Calcular ángulo $ang\_doc_i = 360 - acos\left(\frac{doc_i \times inicio}{||doc_i|| ||inicio||}\right)$
4. $N = 360/g$,
5. Para cada rango de separación (de $n = 1, ..., N$),
   a) Calcula Densidad $Rango_i = Densidad(ang\_dc, rr, rr + g)$.
6. Calcula $Dmin$, $Dmin2$ y $Dmax$.
7. Identificar regiones con mayor densidad,
   - Calcula $menores = Calcula\_menores(Rango, n, Dmin)$.
   - Calcula $mayores = Calcula\_mayores(Rango, n, Dmax)$.
   - Si $menores > 0$,
     a) Calcula $ang\_inicio = Limite1(Rango, n, Dmin)$.
     b) Calcula $ang\_fin = Limite2(Rango, n, Dmin)$.
     c) $Rango\_s_q\{inicio\} = ang\_inicio$.
     d) $Rango\_s_q\{fin\} = ang\_fin$.
   - De lo contrario, si $mayores > 0$,
     a) Calcula $ang\_inicio = Limite1(Rango, n, Dmax)$.
     b) Calcula $ang\_fin = Limite2(Rango, n, Dmax)$.
     c) $Rango\_c_k\{inicio\} = ang\_inicio$.
     d) $Rango\_c_k\{fin\} = ang\_fin$.
8. Para cada $Rango\_c$
   - Si $Rango\_c_{k+1}\{inicio\} - Rango\_c_k\{fin\} > 15$ y No hay $Rango\_s$ y algún $Rango_n < Dmin2$
     a) $nva_separacion = Busca(Rango, Rango\_c\{inicio\}, Rango, Rango\_c\{fin\}, Dmin2)$.
     b) Si $nva_separacion > 0$
        - Crear nuevo renglón para $Rango\_s$.
        - Actualiza Q.
9. Para cada rango de separación
   a) Ajustar índice q para el primer y último elemento.
   b) $Rango\_c_q\{inicio\} = Rango\_s_q\{fin\}$.
   c) $Rango\_c_q\{fin\} = Rango\_s_{q+1}\{inicio\}$.

10. para cada documento i

    a) Formar clases $doc_i \in clase\_k$ si $Rango\_c_k\{inicio\} < ang\_doc_i < Rango\_c_k\{fin\}$.

    b) Formar VA $doc_i \in VA\_k$ si $Rango\_s_k\{inicio\} < ang\_doc_i < Rango\_s_k\{fin\}$.

11. Graficar separación de clases.

S-clase2, . . ., S-clase K).

- *Solo una clase.* Re-entrena los datos de una clase y los VA en ambos costados; separa los elementos traslapados. En un conjunto grande, las palabras compartidas tienen gran impacto, si se eliminan documentos que las poseen e indirectamente se relacionan con la clase a re-entrenar se pueden identificar mejores estructuras y similitudes.

Capítulo V

Diseño de experimentos

Antes de realizar cualquier experimento se definen algunos aspectos a considerar; en primer instancia se establecen los objetivos y alcance de los experimentos. El *objetivo* del conjunto de experimentos es demostrar la eficiencia del modelo propuesto, de tal forma que con una adecuada selección de datos se cubra la mayor parte de puntos críticos para encontrar las bondades y debilidades de nuestra propuesta (alcance). Con el término experimento se refiere a la calibración del modelo con un conjunto de datos determinado; donde la calibración del modelo es el proceso de prueba y error para ajustar los parámetros del algoritmo: η_1, η_2, inicialización de parámetros (W, γ, β), determinar el número adecuado de variables latentes y nodos vecindarios con los que el algoritmo trabaja adecuadamente para un determinado conjunto de datos. En la etapa de calibración del algoritmo se consideran dos aspectos, la dispersión de los datos y la separación de cúmulos en forma de clases separadas. La dispersión entre los documentos varía según el número de variables latentes y nodos vecinos.

Antes de realizar los experimentos se preprocesan los documentos que forman cada base de datos, y se obtiene la matriz. El diseño general de un experimento está formado por cuatro pasos generales:

- ***Preprocesar los documentos.*** Este paso se realiza una sola vez por cada conjunto de datos (base de datos). Cada documento se expresa como un elemento de un espacio vectorial por medio de RI (Salton y McGill, 1983) como se comento en el Capítulo III. Al final se obtiene la matriz de datos y el diccionario de palabras de cada base de datos en particular.

- ***Selección de palabras.*** Éste proceso es de gran impacto en la visualización y sepa-

ración de clases, ya que con una buena selección de palabras los resultados mejoran considerablemente. De las diversas técnicas existentes en este trabajo de tesis se usa la de *frecuencia inversa de documentos* (IDF por sus siglas en ingles) (Salton y McGill, 1983). Uno de los motivos para utilizar esta técnica es su facilidad, rapidez y que no necesita conocimiento a priori de tal forma que emula un escenario real en el que no existe información para obtener conocimiento a priori. Por sí sola esta etapa es un área de investigación por su gran complejidad. En los experimentos solo se realizan entre dos y tres procesos de selección, para determinar el límite inferior.

- ***Simulación.*** Es el proceso por el cual se obtiene la visualización de los datos y separación de clases. En esta etapa se calibra el modelo a partir de la ejecución del programa, ajustando los parámetros para obtener una mejor visualización de datos. Dicha calibración incrementa o disminuye los parámetros para obtener datos con menor dispersión y mayor separación. Cada experimento está formado por diversas simulaciones de tal forma que los parámetros de calibración varían para encontrar una generalización. En el próximo capítulo, donde se presentan los resultados se explicará ampliamente de las generalizaciones encontradas, especificando la calibración del modelo según los diferentes casos estudiados.

- ***Evaluación.*** En el área de visualización de datos, existen pocas herramientas para evaluar numéricamente el resultado visual. En este trabajo aplicamos dos criterios: el FS (Fisher stress), que mide la dispersión entre los datos y las clases y el ESS (Error Sammon Stress) que mide que tanto se preservó la topología de los datos reales con respecto a la nueva proyección en un plano de menor dimensión.

En las siguientes sub-secciones se describe el proceso de experimentación, así como la descripción y características de los datos.

V.1. Descripción de los datos

Para demostrar la eficiencia del modelo de visualización se propone evaluar la representación de los datos, de tal forma que, cada clase sea fácilmente identificable y separable. Con el fin de obtener una generalización de la eficiencia del modelo en diversas situaciones y detectar el alcance del mismo, seleccionamos cuidadosamente las BD, de tal forma que los resultados del modelo sean eficientes en cualquier escenario; desde el más sencillo hasta

el más complicado. Seleccionamos cuatro conjuntos de datos, tres de ellos constan de tres clases cada uno y poseen diferente complejidad. El cuarto conjunto de datos corresponde a 5 clases, de tal forma que se compruebe si el modelo funciona para más clases. A cada conjunto de datos le asignamos un nombre para diferenciarlo de los demás, el cual comienza con la letra ”S”seguida de un número consecutivo. La descripción de los conjuntos es la siguiente:

- ***S1.*** Está formado por tres clases de la colección de datos *Reuters-21578*, distribución 1,0 (disponible en http://kdd.ics.uci.edu/). Se seleccionan 300 documentos en forma aleatoria de las clases (tópicos): *Acquisition* (Acq), *Crude* y *Earn* (100 documentos por cada clase). La temática de ésta colección son noticias internacionales en diferentes áreas: política, negocios, finanzas, entre otras. Y cada documento es el resumen de una noticia (corta o larga), escrito con estilo formal (léxico y gramático) en el idioma inglés.

- ***S2.*** Base de datos formada por documentos de la colección *20-Newsgroup* (http://www.cs.cmu. edu/textlearning). Este repositorio está formado por correos electrónicos de diversas comunidades (electrónica, automóviles, religión, política, computación, entre otras). La redacción de cada documento varia léxica y gramaticalmente, en ocasiones se utiliza algún tipo de caló o lenguaje no formal propio de cada área (en idioma inglés). De ésta BD se seleccionaron 100 documentos de tres clases diferentes: *comp.sys.ibm.pc.hardware*, *comp.sys.mac.hardware* y *sys.med*.

- ***S3.*** Se obtiene a partir de tres bases de datos diferentes CISI (resumen de información científica), Cranfield (resumen de artículos de aerodinámica) y Medlars (resumen de información médica). Al igual que en el conjunto *S1* la redacción de los documentos es formal en el idioma inglés. Se seleccionaron de forma aleatoria 100 documentos de cada clase.

- ***S4.*** Este conjunto de datos es diferente a los tres anteriores porque posee 5 clases y mas documentos por cada una, ya que se pretende evaluar la eficiencia del modelo para mas datos y clases. Al igual que *S2* los documentos provienen del repositorio *20-Newsgroup* con las clases: *rec.sport.baseball* (baseball), *soc.religion.christian* (religion), *talk.religion.guns* (guns), *comp.graphics* (graph) y *rec.autos* (autos). Para cada clase se seleccionaron 200 documentos de forma aleatoria.

Cada BD fue seleccionada considerando que las diferencias entre clases pueden servir para separarlas. En la Tabla VI se resume el análisis de cada BD; donde la primer columna se

refiere al nombre de la base de datos, la segunda es el nombre de la clase o las clases compartidas, la tercera es el porcentaje de palabras exclusivas de la clase o clases antes de seleccionar las palabras y la cuarta es después de la selección de las palabras.

Tabla VI: Análisis de las BD por palabras exclusivas para cada clase.

BD	**Clase**	**Porcentaje**	
		Datos completos	**Después de RP**
S1	Acq	10.61	9.05
	Crude	35.61	18.59
	Earn	9.16	13.07
	Todas las clases	18.46	24.12
	Acq y Crude	10.17	15.58
	Acq y Earn	6.25	5.03
	Crude y Earn	9.74	14.57
S2	IBM	4.5	4.48
	Mac	1.99	1.99
	Med	15.4	15.92
	Todas las clases	40.3	34.83
	IBM y Mac	25	25.37
	IBM y Med	6	5.97
	Mac y Med	5.9	10.95
S3	Cranfield	11.71	6.97
	CISI	13.51	10.66
	Medlar	24.47	23.77
	Todas las clases	18.02	28.69
	Cranfiel y CISI	8.26	6.97
	Cranfield y Medlar	9.76	10.66
	CISI y Medlar	14.26	12.3
	Baseball	4.48	8.42
	Religion	1.95	6.55
	Guns	2.87	6.38
	Graphics	3.33	6.47
	Autos	1.72	6.21

Tabla VI: Continuación.

BD	Clase	Porcentaje	
		Datos completos	Después de RP
S4	Todas las clases	3.56	3.49
	Baseball y Religion	2.87	3.36
	Baseball y Guns	3.45	2.47
	Baseball y Graphics	0.92	0.85
	Baseball y Autos	2.41	1.79
	Baseball y Religion y Guns	5.4	5.36
	Baseball y Religion y Graphics	1.61	1.53
	Baseball y Religion y Autos	1.95	2.38
	Baseball y Guns y Graphics	1.72	1.15
	Baseball y Guns y Autos	4.94	4.30
	Baseball y Graphics y Autos	2.3	2.04
	Baseball y Religion y Guns y Graphics	2.3	1.66
	Baseball y Religion y Guns y Graphics	8.16	4.68
	Baseball y Religion y Graphics y Autos	1.72	1.28
	Baseball y Guns y Graphics y Autos	2.64	1.87
	Religion y Guns	7.7	6.08
	Religion y Graphics	1.84	1.62
	Religion y Autos	1.38	1.23
	Religion y Guns y Graphics	5.4	2.25
	Religion y Guns y Autos	6.67	3.62
	Religion y Graphics y Autos	1.38	1.02
	Religion y Guns y Graphics y Autos	4.83	2.68
	Guns y Graphics	1.03	1.53
	Guns y Autos	3.33	3.83
	Guns y Graphics y Autos	3.45	2.00
	Graphics y Autos	2.64	1.91

El análisis de las palabras consta de dos etapas, evaluación de las palabras antes del proceso de selección (datos completos incluyendo stop words y steming) y después del proceso (reducción de la dimensión). En la tercera columna de datos se consideran aquellas palabras que aparecen en cierta cantidad de documentos. En éste trabajo consideramos que

una palabra que aparece solo en 5 documentos no es significativa para identificar clases (poca información para un total de 300 o 10000 documentos). Las características principales del análisis de selección de palabras son:

- Los datos de la tercera columna se obtuvieron con las palabras que aparecen en más de 4 documentos.
- La cuarta columna corresponde al análisis después de la selección de palabras, donde la dimensión se reduce después de cambiar las palabras con la misma raíz y eliminar las *stop words* y aquellas que son de uso común o que aparecen en muy pocos documentos.
- Una palabra se considera exclusiva de una clase si aparece en más del $50\,\%$ del total de documentos de una clase. Ésta misma palabra puede aparecer en otras clases con un máximo de ocurrencia del $14\,\%$, es decir, que casi no es mencionada en otras clases.
- Se considera palabra compartida por todas las clases, a aquellas que aparecen en más del $15\,\%$ de las clases.
- Las palabras compartidas por dos o más clases son aquellas que aparecen en más del $15\,\%$ de los documentos de cada clase; pero en las otras clases a lo más aparecen en el $14\,\%$ de los documentos.

Los resultados mostrados en la Tabla VI son útiles por dos aspectos importantes, el primero se refiere a la evaluación de la facilidad para separar los documentos según su clase y el segundo en la evaluación de la eficiencia del modelo de selección de palabras. En la siguiente sub-sección se define la clasificación de tres diferentes formas en que se presentan los VA. Dicha clasificación fue realizada basándose en el análisis de las palabras y de los documentos ubicados en diferente lugar a su clase en resultados previos a los experimentos de evaluación del modelo.

V.1.1. Clasificación de VA

Después de un arduo análisis en las BD y de revisar el análisis de los resultados de experimentos previos, se identificaron ciertas características en los documentos que fueron ubicados en diferente lugar (fuera de su clase). De acuerdo a la evaluación los VA fueron etiquetados de cuatro formas diferentes que corresponden a tres categorías, la clasificación se realizó de acuerdo a la frecuencia y el criterio RP de la siguiente forma:

- *VA tipo 1*. Son los documentos que contienen pocas palabras, menos de la media de los documentos que forman la BD con el siguiente criterio,

$$\#palabras(\mathbf{t}_n) < \mu - \sigma$$

donde $\#palabras(\mathbf{t}_n)$ es el total de palabras en el documento n, μ es el promedio de palabras por documento en la BD, σ es la desviación estándar. Estos valores se calculan en base a la matriz de frecuencia después de la selección de palabras; para eliminar aquellas palabras de poco uso y que no aportan información sobre la clase a la que pertenecen los documentos. En este tipo de VA se consideran dos clases; el tipo $1a$ es aquel donde el número de palabras en el documento antes de la selección de palabras también es pequeño. El VA $1b$ cuando el total de palabras es de tamaño promedio, pero comienza a ser pequeño después del procedimiento RP. Esto es porque contiene palabras que pocas veces son usadas -a las que denomino *paja*; en la gráfica (2) se localizan en la sección *C*.

- *Tipo 2*. Se considera que un documento es de este tipo cuando la mayoría de las palabras en el documento son utilizadas en la mayoría de las clases. En el histograma de RP se localizan junto a la sección *A* (Figura 2) La característica numérica es que el total de palabras exclusivas de la clase sea menor al $40\,\%$ del total de palabras en el documento, y las otras palabras sean compartidas en todas las clases o en algunas de ellas.

- *Tipo 3*. Este tipo de VA considera a los documentos que contienen suficientes palabras; la mayoría de ellas se encuentra en la región B de la Figura (2), pero la mayoría de ellas corresponden a una clase diferente. Un documento en esta categoría posee más del $40\,\%$ de palabras exclusivas, pero de otra clase; además también posee palabras compartidas entre dos clases. Por ejemplo un documento de la BD *S1* que pertenece a la clase *Acq*, con un total de 20 palabras de las cuales 2 son palabras exclusivas de *Acq*, 5 de *Crude*, 6 compartidas por todas las clases y 7 compartidas por *Acq* y *Crude*; es considerado VA tipo 3 por su similitud con las palabras de *Crude*.

V.1.2. Definición de los escenarios de trabajo

Un elemento importante en el diseño de los experimentos, es corroborar que las BD seleccionadas poseen las características de los escenarios deseados para los experimentos. Es decir que el modelo funcione adecuadamente en las mejores condiciones y en las más

complicadas donde la separación de las clases no siempre es posible.

En esta tesis se consideran cinco características principales para los escenarios: clases separables, difíciles de separar, documentos pequeños (considerados VA), ruido (palabras mal escritas o abreviadas de diferente forma como en 20-Newsgroup), documentos escritos con palabras muy comunes o con palabras pocos descriptivas de la clase (VA) y un conjunto con al menos 4 clases. El primer escenario se refiere al mejor de los casos, donde la separación es posible y no existe complicación para separar las clases. En el segundo escenario se requiere que al menos dos clases tengan gran dificultad para ser separadas, de tal forma que el modelo obtenga la separación de los grupos aún con elementos mezclados. El tercer escenario es complicado por la presencia de documentos pequeños, ya que su clasificación es difícil porque no existe la información suficiente para relacionarlos con un grupo u otro. A éstos documentos se les considera VA, su tamaño disminuye después de la selección de palabras. El cuarto escenario o condición se refiere al ruido y VA; retomando la definición de ruido (sección II.1) consideramos aquellas palabras mal escritas, modismos locales ó abreviaciones de palabras utilizadas de forma indistinta (conjunto *S2* y *S4*). Los VA son los documentos con problemas de uso frecuente o mal uso de las palabras, la mayoría de las BD poseen algunos en menor o mayor cantidad. Esto se debe a que los documentos son escritos por diferentes personas. El quinto escenario se refiere a la separación de diferentes clases, para constatar que el modelo funciona con mayor número de clases y datos.

A continuación describimos las características principales de las BD y en que forma cumplen con los escenarios descritos.

La BD *S1* posee las características del primer escenario; como se observa en la tercer columna de la Tabla VI. Donde la clase *Crude* es claramente identificable de las demás, por otra parte las clases *Acq* y *Earn* no poseen suficiente información como *Crude* pero, no hay dificultad para separarlas. Otro aspecto que dificulta la separación de las clases es la cantidad de palabras compartidas, en ésta BD la cantidad de palabras compartidas entre las tres clases son pocas, sólo el $18{,}46\,\%$. En las clases *Acq* y *Earn* existe poca probabilidad de documentos traslapados; aunque la clase *Crude* posee mayor número de palabras también comparte más palabras con las otras dos clases. Esta última característica es un posible factor de los documentos considerados VA, como se mencionó en el Capítulo II. El ruido está relacionado con aquellas palabras que pertenecen al grupo de las no utilizadas, por lo que no son un factor que interfiera en la identificación de una clase. La relación de pala-

bras entre diferentes clases también es un indicador de posibles VA.

S2 es una BD complicada, cumple las características del segundo caso de estudio y posee características de ruido y VA. Respecto a esto último, dos de las clases (*IBM* y *Mac*) se consideran difíciles de separar porque más del $25\,\%$ de las palabras son compartidas por ellas. Indicando que las clases se traslapan y la separación es difícil. Como se mencionó en la descripción de la BD, este repositorio está formado por la recopilación de los correos electrónicos redactados por diferentes usuarios con un estilo informal. El ruido se va formando por la diversidad de palabras, modismos locales, abreviaciones realizadas de forma indistinta, léxico y expresiones que se alejan de la semántica que en pocas ocasiones coincide con otros. Esto provoca que el diccionario de palabras vaya creciendo, las palabras son de poco uso o de uso general (Tabla VI); el $40\,\%$ de las palabras son compartidas por las tres clases.

El resultado de la Tabla VI indica que en la BD *S3* algunos documentos de diferentes clases comparten demasiadas palabras comunes en todas las clases o en pares de ellas. La BD posiblemente posee clases traslapadas y gran cantidad de VA de los tres tipos. Once documentos poseen pocas palabras (de 9 a 16), por lo que se espera que después de la selección de palabras su tamaño se reduzca más. En la tercer columna de la Tabla VI los parámetros indican que no existe traslape entre clases y que la separación puede ser posible. Por lo que el mayor reto del modelo en este caso de estudio es demostrar su robustez para trabajar con VA.

Finalmente, el quinto caso de estudio está orientado a demostrar la eficiencia del modelo para separar más clases, de tal forma que demuestre ser eficiente en la mayoría de las situaciones. La BD *S4* posee cinco clases, al igual que en la BD *S2* los documentos son la recopilación de correos electrónicos o mensajes de un foro de discusión sobre temas específicos, *S4* posee alta probabilidad de VA. Pocas son las palabras exclusivas de cada clase; las clases *Religion* y *Guns* poseen muchas palabras compartidas, por lo que su separación parece que será complicada o habrá mucha proximidad en la proyección.

V.1.3. Selección de palabras

Después de representar los documentos en un espacio vectorial y formar la matriz de datos, se tiene una matriz con alta dimensión. Aunque en el proceso de recuperación de

información se evitó elevar la dimensión de la matriz de datos[1], ésta continúa siendo alta. Para reducir considerablemente la dimensión y obtener mejores resultados se aplica un modelo de selección de palabras. Seleccionamos la técnica de RP con ordenamiento de las palabras en orden descendente; debido a que no necesita conocimiento a priori. Existen dos razones para preferir esta técnica, la primera es porque al no necesitar análisis previo de los datos hay probabilidad de seleccionar palabras de poco uso, forzando al modelo a explotar su aspecto robusto. Por otra parte si el algoritmo VL-ZIP es parte de un proyecto que trabaje en tiempo real se desconocería el conocimiento a priori; entonces, ésa tesis sirve como precedente para demostrar su eficiencia en condiciones que no son favorables.

La técnica de RP está formada por cinco pasos, como se explicó en la sección II.1.2; RP considera que las palabras útiles son las que se encuentra a la mitad de la curva graficada, esto implica que los dos extremos de las frecuencias no son de importancia. El extremo superior se refiere a aquellas palabras que se repiten en muchos documentos. El límite superior propuesto consiste en identificar el punto donde la diferencia entre frecuencias es mínima (las frecuencias se encuentran ordenadas de mayor a menor). Por ejemplo, en la Tabla VII están las primeras palabras de la BD *S4*; donde el posible límite se encuentra entre las líneas 17 y 20 donde la diferencia es de 1. Por otra parte, la selección del límite inferior es simple: definir un número mínimo de documentos que posee la palabra, y esto depende de la cantidad de documentos por ejemplo en ésta misma BD el límite se selecciona entre 15 y 20 documentos. Como se observa en la Tabla VII el número de palabras que se eliminan en la parte superior es mínima con respecto a las que se eliminan en la parte inferior que en éste ejemplo radica entre 9705 y 9662.

La técnica de selección de palabras parte de que no existe conocimiento a priori de la información, por lo que no es posible detectar la eficiencia de la selección. Cada selección de palabras se evaluará por medio del procedimiento de prueba y error; es decir, ejecutar el modelo ZIP con cada selección de palabras e identificar su eficiencia. Después de obtener los resultados se seleccionarán aquellos cuya visualización sea más clara; en caso de no haber diferencias significativas en los resultados se seleccionará el que tenga menor número de palabras.

[1]Al eliminar las palabras comunes *stop words* y aplicar el algoritmo de *stemming* (que evita las palabras con la raíz)

Tabla VII. Palabras de la BD *S4*, ordenadas de forma descendente, para seleccionar el limite superior de corte con la técnica de RP.

	Palabra	frecuencia x documentos		Palabra	frecuencia x documentos
1	just	395	16	really	178
2	know	356	17	take	175
3	think	346	18	believe	174
4	time	322	19	car	173
5	people	279	20	run	165
6	best	261	21	work	163
7	look	253	22	start	158
8	way	242	23	read	153
9	see	240	24	differ	151
10	want	226	25	mean	149
11	thing	216	26	post	145
12	point	202	27	problem	143
13	right	200	28	call	142
14	need	190	29	try	141
15	come	188	30	long	136

V.2. Calibración del modelo

En esta etapa del proceso definimos los pasos a realizar para cada simulación del modelo a partir del conjunto de datos extraídos por las técnicas de RI. La primer parte de la simulación corresponde al modelo de visualización de datos, modelo de espacios latentes VL-ZIP; y la segunda parte de la simulación es la separación de clases. El modelo de VL-ZIP es el más complejo, pues en él están involucrados diferentes parámetros que afectan la dispersión y separación de los datos.

V.2.1. Calibración del modelo VL-ZIP

La simulación del algoritmo VL-ZIP consiste en ejecutar varias veces con diferentes valores para η_1, η_2 y diferentes inicializaciones para W, γ y β. En Bishop *et al.* (1998) se describe la importancia de inicializar los parámetros W y β de forma apropiada. Existen dos opciones para iniciar el parámetro W, la primera utilizando el mismo principio que Bishop *et al.* (1998)[2]. La segunda opción es utilizar la adaptación de la ecuación (38), donde sólo se

[2]Traslada la estructura latente (mapping) a un hiper plano interceptado por los L componentes principales del conjunto de datos.

considera la parte positiva para obtener (39).

Se decidió trabajar con la segunda opción; la propuesta de Bishop *et al.* (1998) está basada en componentes Gaussianos y no se considera adecuada para la mezcla de funciones ZIP. Además, en (Lambert, 1992) se comenta que el modelo converge rápidamente con ésta opción. Primero se obtienen valores iniciales de θ (seleccionados de forma aleatoria); posteriormente W (aleatorio), Φ (28) y β (aleatorio) se inicializan para calcular $\lambda^{(0)}$ como en (23); finalmente se inicializa W^0 como en (36 y β^0 como en (37).

Como el parámetro G es la matriz de *covarianza* para calcular la probabilidad de que la frecuencia de una palabra en el documento sea 0, se selecciona como el promedio de las frecuencias de la matriz de datos T de la siguiente forma,

$$G_{nj} = \begin{cases} \frac{1}{D - n_0} & \text{Si } t_{nj} = 0 \\ \frac{\mathbf{t}_{nj}}{D - n_+} & \text{de otra forma,} \end{cases} \tag{45}$$

donde D es el total de palabras utilizadas en la matriz de datos. n_0 es el total de palabras en todos los documentos cuya frecuencia es igual a cero; y n_+ es la suma de las palabras en los documentos cuya frecuencia es mayor a cero.

El parámetro $\gamma^{(0)}$ es seleccionado de forma aleatoria con distribución $\mathcal{N}(0,0{,}1)$, y $\alpha^{(0)}$ es inicializado como el promedio de las variables latentes $(1/K)$. Por otra parte, los parámetros que definen la rapidez de convergencia del modelo (η_1 y η_2) tienen valores pequeños para evitar saltos bruscos en las iteraciones del EM.

En la etapa de calibración del modelo LV-ZIP además de encontrar los valores óptimos para algunos parámetros importantes, también podemos definir el comportamiento de los resultados en relación a éstos valores. En la siguiente sección de resultados, anexaremos la relación del comportamiento del modelo y los resultados.

V.3. Evaluación de los resultados

Uno de los objetivos del modelo de visualización es identificar las estructuras de conglomerado, por si solo el modelo no las puede identificar; para lograrlo requiere trabajo a posteriori. En la literatura revisada encontramos diversos métodos para evaluar conglomerados, la mayoría de ellos se basan en la separación y formación de grupos (Halkidi *et al.*, 2001). Sin embargo, en el área de visualización de datos existe poca información

relacionada con la evaluación de los resultados. Esto se puede deber a la variedad de características buscadas en la proyección de datos, además de que las medidas de evaluación no siempre reflejan las características visuales (proceso subjetivo).

En éste trabajo de investigación se decide medir la dispersión de y entre los datos y la preservación de la topología de los datos. Esto es posible por medio de los índices SF (dispersión basada en el criterio de Fisher (Duda *et al.*, 2001)) y ESS (Error Samon's Stress) -que mide la preservación de la topología (Sammon, 1969).

El índice SF evalúa la dispersión de la proyección de datos, es decir, el cociente de la variación en-clases y entre-clases, dado por la formula,

$$S_F = \frac{\|S_B\|}{\|S_W\|}, \tag{46}$$

donde $\|.\|$ es la norma de Frobenius $\|A\| = \sqrt{\sum_i \sum_j a_{ij}^2}$. S_B y S_W son las matrices de dispersión entre-clases y en-clases respectivamente dadas por,

$$\begin{aligned} S_B &= \sum_k N_k(\mathbf{m}_k - \bar{\mathbf{m}})(\mathbf{m}_k - \bar{\mathbf{m}})^T, \\ S_W &= \sum_k \sum_{i \in K} (\mathbf{t}_i - \mathbf{m}_k)(\mathbf{t}_i - \mathbf{m}_k)^T, \end{aligned}$$

con

$$\mathbf{m_k} = \frac{\mathbf{1}}{\mathbf{N_k}} \sum_{\mathbf{i} \in \mathbf{k}} \mathbf{t_i},$$

y

$$\bar{\mathbf{m}} = \frac{1}{N} \sum_i \mathbf{t}_i = \frac{1}{N} \sum_k N_k \mathbf{m}_k.$$

donde N_k es el número de elementos en la clase k, $\mathbf{m}_k$ es la media de la clase k y $\bar{\mathbf{m}}$ es la media total. Éste índice proporciona información cuantitativa de los datos en los conglomerados, pero solo es posible si se tiene conocimiento de la clase a la que pertenece cada dato.

El SF nos indica la dispersión de los datos en cada conglomerado. Como se mencionó anteriormente, la dispersión de la proyección de los datos sólo se puede aplicar en la etapa de evaluación del modelo, pues en la práctica se desconoce la clase a la que los documentos pertenecen. Entre más grande sea el valor SF significa que la dispersión de los datos y las clases es mejor, es decir, que la mayoría de los datos pertenecen a su clase origen, que

éstos se encuentran proyectados de forma cercana y que las clases son fácilmente identificables. Por el contrario, un valor muy pequeño indica que los datos se encuentran muy dispersos y/o que la separación de las clases no es clara.

Por otra parte, el índice ESS mide la preservación de la topología de los datos entre la proyección y los datos originales ($\mathbb{R}^D \rightarrow \mathbb{R}^2$); lo cual nos indica que la función de probabilidad del modelo ajusta los datos. La versión normalizada está dada por,

$$E_{SS} = \frac{1}{\sum_{n=1}^{N-1}\sum_{j=n+1}^{N} d(a_n, a_j)} \sum_{n=1}^{N-1}\sum_{j=n+1}^{N} \frac{[d(a_n, a_j) - d(b_n, b_j)]^2}{d(a_n, a_j)} \tag{47}$$

donde, d es la distancia Euclideana; a y b son los datos de entrada y los proyectados respectivamente. Entre menor sea éste resultado mayor es la preservación de la topología. Si el modelo de proyección de datos está basado en cierta función de distribución, un valor pequeño significa que dicha función ajusta los datos adecuadamente. A diferencia del índice anterior, el ESS no requiere conocimiento a priori para ser evaluado. Por lo que se convierte en un índice útil en tiempo real, eficiente para confiar en la dispersión de los datos proyectados y la separación de las clases.

La prueba SF sólo es usada en la fase de experimentación en donde se conoce la clase a la que cada documento pertenece, mientras que el ESS se usa en la experimentación y en tiempo real, para indicar qué tan parecidos son los resultados obtenidos con respecto a los completos. En términos numéricos para medir cuantitativamente la proyección del los datos con el modelo, un buen resultado es aquel donde el SF es grande y el ESS pequeño. Con lo que se supone que los datos proyectados poseen buena dispersión, las clases son separadas adecuadamente y la representación visual capta la topología de los datos completos.

Capítulo VI

Resultados

El algoritmo de visualización LV-ZIP proporciona una gráfica en $2D$ con los documentos proyectados; uno de los objetivos es extraer las características de dispersión de los datos y su separación en los diferentes grupos temáticos. A partir de estos datos proyectados en un nuevo plano, se aplica el esquema propuesto para separar e identificar los diferentes cúmulos de datos (formar clases y subclases). Los resultados que se presentan a continuación describen la eficiencia de los dos algoritmos propuestos. En la primer parte se describen los resultados de la visualización y la evaluación de los parámetros descritos en la sección anterior. Después se describe el análisis de los resultados obtenidos y la identificación de algunos VA encontrados en las diferentes bases de datos. Finalmente se presentan los resultados de la separación de clases y el análisis de éstos.

VI.1. Resultados de la visualización

Como se mencionó en la descripción de experimentos, cada BD fue seleccionada para encontrar las bondades y debilidades del algoritmo propuesto. Los VA están presentes en la mayoría de los escenarios y son de gran importancia para mostrar la robustez del algoritmo. Antes de presentar los resultados de la visualización se describen los resultados en la selección de palabras (por medio de la técnica Resolving Power").

En la Tabla VIII se presenta la dimensión de los datos completos y después del proceso de RP. La primer columna identifica el conjunto de datos, en la segunda se encuentra el tamaño original de la matriz de frecuencia, sin considerar a las "stop words"; en la tercer columna se presenta el número de palabras que se encuentran en menos de 5 documentos. Y en la cuarta columna el tamaño final del conjunto de datos. Como se aprecia la mayor cantidad de palabras de un conjunto de datos pocas veces se repiten, y no son

relevantes en la identificación de grupos y sub-grupos. Como se menciona en la teoría del RP, su ubicación en la gráfica del histograma se localiza en la parte derecha.

Tabla VIII. Reducción de la dimensión con Resolving Power.

Conjunto de Documentos	**Número inicial de palabras**	**Número de palabras en menos de 5 documentos**	**Dimensión final**
S1	3811	3003	199
S2	4669	3610	201
S3	3415	2641	244
S4	10727	7988	993

La técnica de RP requiere un límite superior e inferior, donde se encuentran las palabras muy o poco comunes que no aportan información para la identificación de clases. En la Tabla IX se especifican los límites superior e inferior para el RP; dados en función del número de documentos (frecuencia por documentos; ver sección II.1.2). En la Figura 12 se presentan las gráficas con el histograma según el número de documentos donde se menciona cada palabra. En el eje de las *x* se encuentran las palabras ordenadas según la frecuencia por documentos (el número de documentos donde se menciona cada palabra) en orden descendente (izquierda mayor uso, derecha menor uso). Y el eje de las *y* representa el producto $frecuencia \times rango$ tal como se especificó en el capítulo anterior.

Tabla IX. Límites superior e inferior para la técnica de Resolving Power; es decir, el mínimo número de documentos donde la palabra es mencionada.

Conjunto de datos	**Límite superior**	**Límite inferior**
S1	84 documentos	18 documentos
S2	80 documentos	20 documentos
S3	62 documentos	15 documentos
S4	122 documentos	17 documentos

En el conjunto de datos *S3* el límite inferior requirió utilizar más palabras, debido a que algunos de los elementos quedaban vacíos si se utilizaban menos palabras, por lo que hubo necesidad de ampliar el límite inferior. Ésta misma condición genera la existencia de VA tipo 1, en la Tabla X se aprecia una descripción detallada de los VA tipo 1 de los datos *S3*. De todas las BD, ésta es la que más VA tipo 1 posee, probablemente por el tipo de redacción

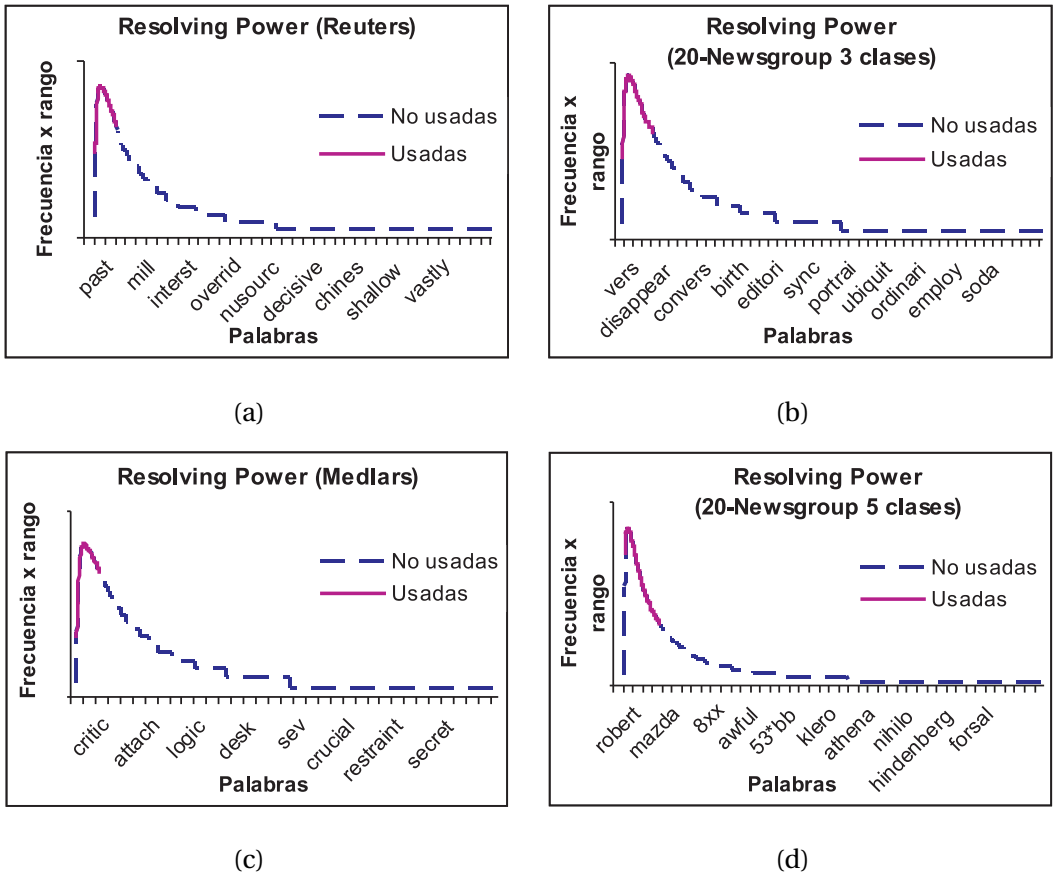


Figura 12. Resolving power de las cuatro BD; a) conjunto *S1*, b) conjunto *S2*, c) conjunto *S3* y d) conjunto *S4*.

de los artículos.

En los experimentos realizados, se observó que el límite superior tiene gran impacto en los resultados de proyección; esto se debe a que existe gran cantidad de palabras que tienen mucho uso común aunque no son consideradas ”stop words”. En el proceso de selección de palabras al aplicar la técnica de RP el límite superior es importante. Esto se hace como se indicó en el capítulo de diseño de experimentos. En todos los conjuntos de datos con los que se experimentó se presentó la misma característica; las palabras que se repiten en mayor cantidad de documentos, poseen una diferencia marcada con respecto a las que se repiten regularmente. Es decir, que cuando se encuentran ordenadas descendentemente por frecuencia en documentos, la diferencia entre frecuencias de las palabras de uso común es mayor que en las de uso regular.

Tabla X. Descripción de VA tipo 1 en el conjunto de datos $S3$.

Identificador del documento	**Clase**	**Tamaño inicial**	**Final**	**Frecuencia $\leq$ 5 docs.**	**Frecuencia $\leq$ 10 docs.**	**Frecuencia $\leq$ 15 docs.**
1	Cranfield	18	2	13	2	1
2	CISI	17	2	9	4	2
3	CISI	7	2	4	1	0
4	Cranfield	14	3	8	0	3
5	Cranfield	16	4	9	1	2
6	Cranfield	11	4	6	1	0
7	Cranfield	9	4	5	0	0
8	Cranfield	13	4	5	4	0
9	Cranfield	26	4	15	5	2
10	CISI	15	4	7	4	0
11	CISI	27	5	18	2	2
12	Cranfield	25	5	14	2	4
13	CISI	27	5	13	5	4
14	Cranfield	27	5	17	3	2
15	CISI	17	6	6	3	2
16	Cranfield	23	6	11	4	2
17	CISI	16	6	6	4	0
18	CISI	10	6	1	1	2
19	Cranfield	23	6	10	4	3

Por otra parte, el límite inferior también tiene un cierto impacto, pues de forma hipotética sería conveniente usar todas las palabras para obtener un mejor resultado. Sin embargo en la práctica se convierte en un proceso poco redituable; debido a que el costo computacional se incrementa, y los resultados no son adecuados en todos los casos. En los experimentos de selección de los límites para la técnica RP se consideraron diferentes valores según el siguiente criterio: que la palabra apareciera en al menos en 30, 25, 20 y 15 documentos. Se encontró que a partir de la frecuencia de 25 documentos el algoritmo comienza a proporcionar buenos resultados; pero de 20 a 15 documentos los resultados casi siempre son mejores. Por lo que el valor del límite inferior es de 20 documentos; con la limitante de que se disminuirá el límite sólo si algún documento posee menos de 3 palabras.

Se realizaron los experimentos de calibración del algoritmo una vez que las palabras fueron seleccionadas. Se decidió trabajar con una malla de centros de tamaño 7×7 ($M = 49$) y 105 variables latentes ($K = 105$), ya que son una cantidad adecuada para no aumentar el tiempo computacional. Los primeros parámetros a calibrar fueron la velocidad de conver-

gencia del algoritmo (η_1 y η_2). Después de varios experimentos se encontró que los mejores resultados se obtuvieron para $0{,}08 < \eta_1 < 0{,}2$ y $0{,}001 < \eta_2 < 0{,}0007$ (a prueba y error).

A partir de éste momento se realizaron los experimentos para obtener la proyección de los datos en un plano de 2D. En esta etapa de experimentación se presenta la evaluación de la proyección del algoritmo LV-ZIP, comparados con cuatro modelos diferentes: el modelo GTM con la función Gaussiana de Bishop *et al.* (1998), Multinomial de Kabán y Girolami (2001), el modelo GTM con Poisson y el modelo Latent Dirichlet Allocation (LDA) de Blei *et al.* (2003). El primer experimento corresponde al conjunto de datos *S1*, en la Tabla XI se presentan los resultados de los parámetros S_F y E_{SS} para los cuatro conjuntos de datos de los resultados con el algoritmo VL-ZIP y los otro cuatro modelos con los que se comparó. Y en la Figura 13 se muestran las gráficas con los datos proyectados en 2D.

Tabla XI. Comparación de la eficiencia de los modelos Gaussiano, Multinomial, Poisson, LDA y VL-ZIP.

	S1		**S2**		**S3**		**S4**	
Modelo	S_F	E_{SS}	S_F	E_{SS}	S_F	E_{SS}	S_F	E_{SS}
Gaussiano	0.042	0.42	0.013	0.69	0.028	0.47	-	-
Multinomial	0.051	0.43	0.030	0.45	0.069	0.48	0.065	0.47
Poisson	0.034	0.52	0.012	0.53	0.038	0.49	0.050	0.70
LDA	0.069	0.62	0.029	0.67	0.091	0.67	0.066	0.68
ZIP	0.096	0.41	0.150	0.470	0.130	0.39	0.098	0.47

Como se aprecia en la Figura 13, los resultados con el modelo Gaussiano son los peores, esto es de esperarse debido a que el modelo Gaussiano no es adecuado en la representación de este tipo de datos, por lo que en las evaluaciones posteriores no se considera este modelo. Como se puede apreciar en la Figura 13 la separación de clases se aprecia mejor con el algoritmo VL-ZIP y el Multinomial. En la Tabla XI se comprueba que existe menor dispersión con el algoritmo propuesto y que la topología de los datos se preserva. De la Figura (13) se puede apreciar que para los modelo Multinomial, LDA y VL-ZIP la mayoría de los documentos están agrupados en sus clases. Pero en los primeros dos modelos las clases no poseen suficiente espacio entre cada una. En las proyecciones con el algoritmo VL-ZIP las clases están separadas (el S_F es mayor que los demás). En la siguiente sub-sección se realiza el análisis de los documentos fuera de clase en los resultados del algoritmo LV-ZIP, que nos indica por qué se encuentran dentro de otra clase o alejados de su clase (se identifican los VA y el tipo al que pertenecen según su características).

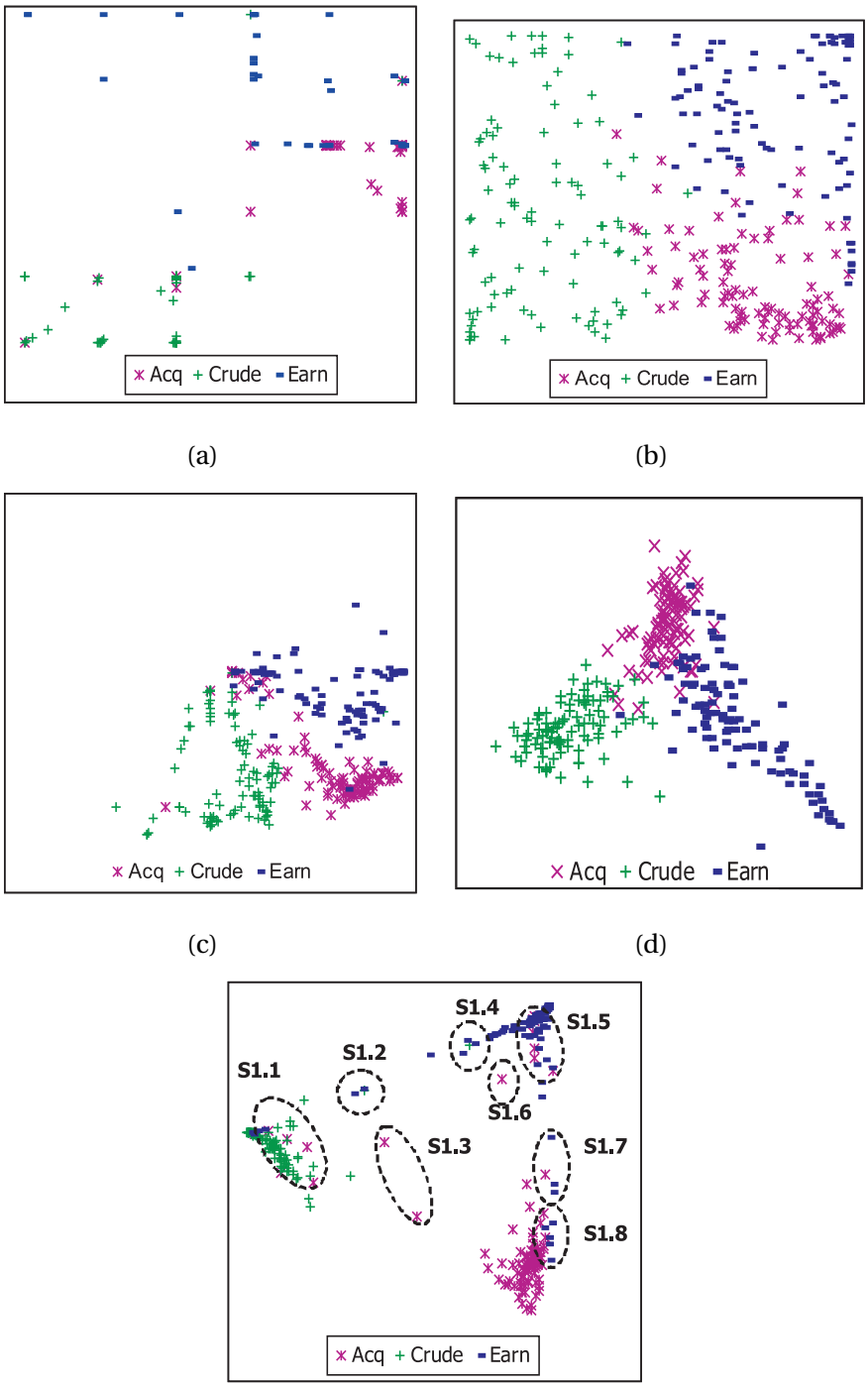


Figura 13. Proyección de la BD *S1*(Reuters) con 5 modelos diferentes: a) Gaussiano, b)Multinomial, c) Poisson, d) LDA y e) VL-ZIP.

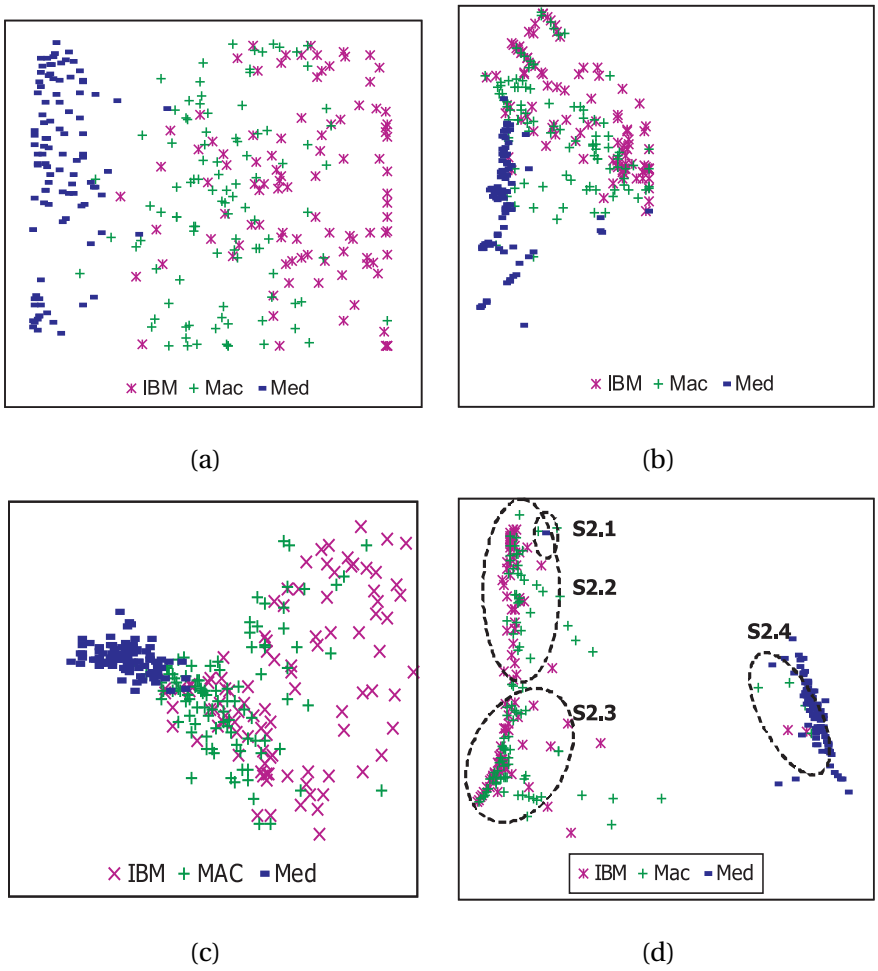


Figura 14. Proyección de la BD *S2* (20-Newsgroup) con 4 modelos diferentes: a) Multinomial, b) Poisson, c) LDA y d) VL-ZIP.

Los resultados de la BD *S2* (20-Newsgroup) se presentan en la Figura 14. Como se mencionó en la descripción de los datos, en este conjunto es difícil separar las clases IBM y Mac, debido a la gran similitud del contenido de los documentos (ambos hablan sobre aspectos técnicos de software y hardware de computadoras, pero de diferente marca). El algoritmo VL-ZIP separa la clase *Med* por completo. Y de las otras dos clases se generaron dos grupos; pero cada grupo poseen cierta cantidad (al menos el $50\,\%$) de datos mezclados de ambas clases. Mientras que con el modelo Multinomial solo se identifican dos clases: *Med* y las otras dos de forma mezcladas.

El siguiente conjunto de datos evaluado es *S3* (Medlars), que se muestra en la Figura 15 el cual posee mayor cantidad de VA tipo 1 como se mostró en la Tabla X. A pesar de poseer

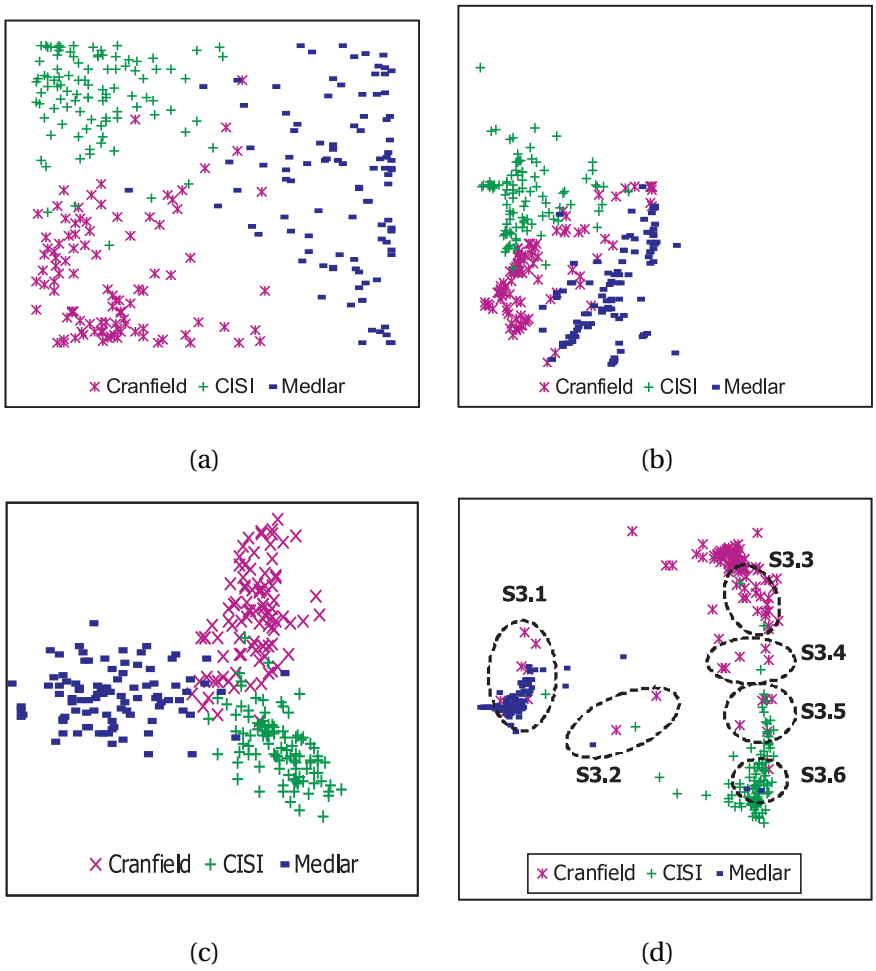


Figura 15. Proyección de la BD *S3*(Medlars) con 4 modelos diferentes: a) Multinomial, b) Poisson, c) LDA y d) VL-ZIP.

mayor cantidad de elementos con poca información, el algoritmo VL-ZIP proporciona los mejores resultados de dispersión y preservación de la topología con este conjunto de datos; como se muestra en la Tabla XI. La separación de las clases también se encuentra muy marcada, igual que en los conjuntos anteriores (*S1* y *S2*).

Finalmente, el conjunto *S4* (20-Newsgroup con 5 clases) marca la eficiencia del algoritmo VL-ZIP para un conjunto de datos grande y con más de tres clases. Como se mencionó en la descripción de los escenarios, con este conjunto de datos se evaluará la robustez del algoritmo y su eficiencia para separar clases; aún con la similitud de palabras entre clases (compartir diversas palabras en más de dos clases). En el análisis de palabras de éste conjunto de datos (Tabla VI) se intuye que la clase *Baseball* es separada de las demás sin pro-

blema alguno; esto es porque posee suficientes palabras exclusivas y en la mayoría de los casos donde se comparte la palabra el porcentaje es pequeño. Por otra parte, se observa que las clases *Religion* y *Guns* comparten gran cantidad de palabras. El otro conjunto de clases que también comparten palabras son las dos clases anteriores y *Baseball*, pero como ésta última posee mayor cantidad de palabras exclusivas existe poca evidencia de que sea un problema. Además, las clases *Religion* y *Autos* poseen pocas palabras exclusivas.

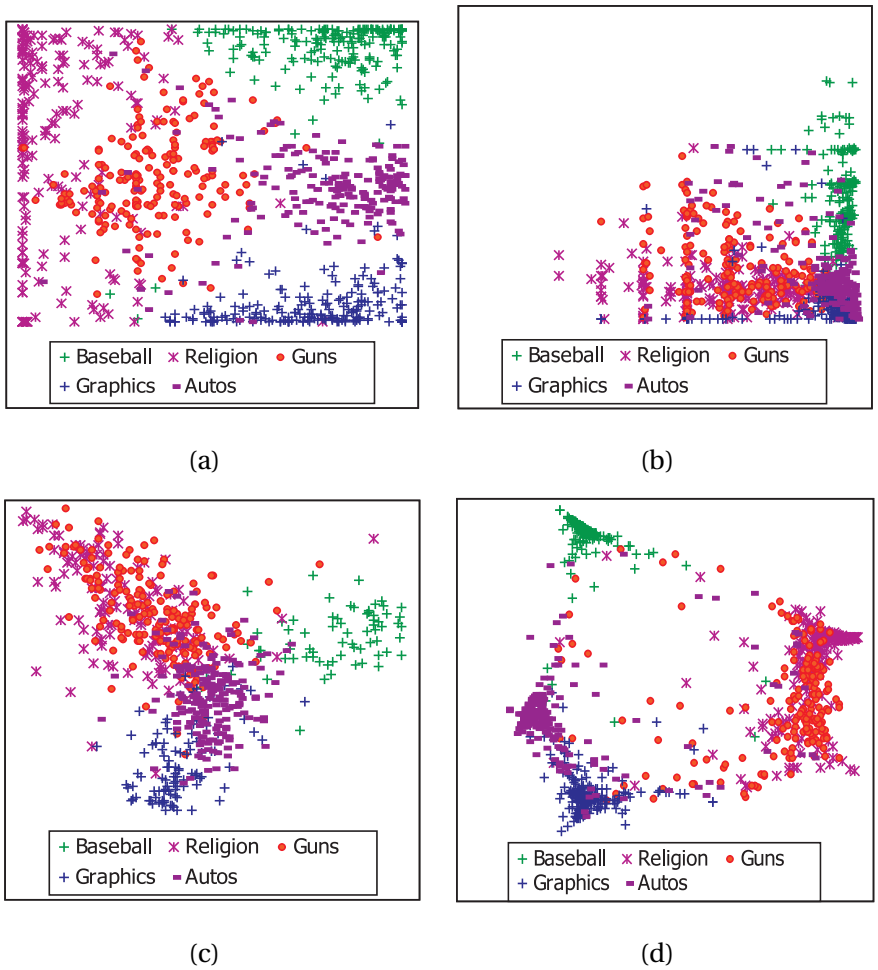


Figura 16. Proyección de la BD *S4*(20-Newsgroup) con 4 modelos diferentes: a) Multinomial, b) Poisson, c) LDA y d) VL-ZIP.

En la Figura 16 se aprecia que, en la mayoría de los casos existen documentos de las clases *Religion* y *Guns* mezclados, tal como se esperaba según el análisis de palabras de este conjunto de datos. La proyección de los modelos Multinomial y el propuesto (VL-ZIP) poseen la misma eficiencia en preservar la topología (Tabla XI). Pero en cuanto a dispersión y se-

paración de clases, el algoritmo VL-ZIP posee mayor valor S_F, que se traduce como mejor separación de los datos y los datos se encuentran menos dispersos. Visualmente con el VL-ZIP la mayoría de las clases se pueden separar, y con el Multinomial algunas también son separables.

De las gráficas anteriores, se puede observar una característica en la dispersión de los datos que persiste en la mayoría de los experimentos. Cada clase representada como un cúmulo, posee una protuberancia que sobresale a partir de la mayor concentración de datos y se extiende formando un pico. Tal como se muestra en la Figura 17, en la que se extrajeron algunas clases donde la protuberancia se presenta de diferentes formas, en algunas más marcadas que en otras. De tal forma que en las gráficas de proyección de los datos con el algoritmo VL-ZIP las clases se identifican cuando se observa alguna imagen similar a las de la Figura 17.

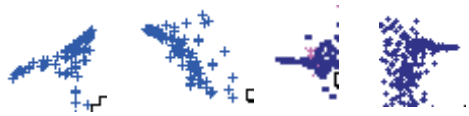
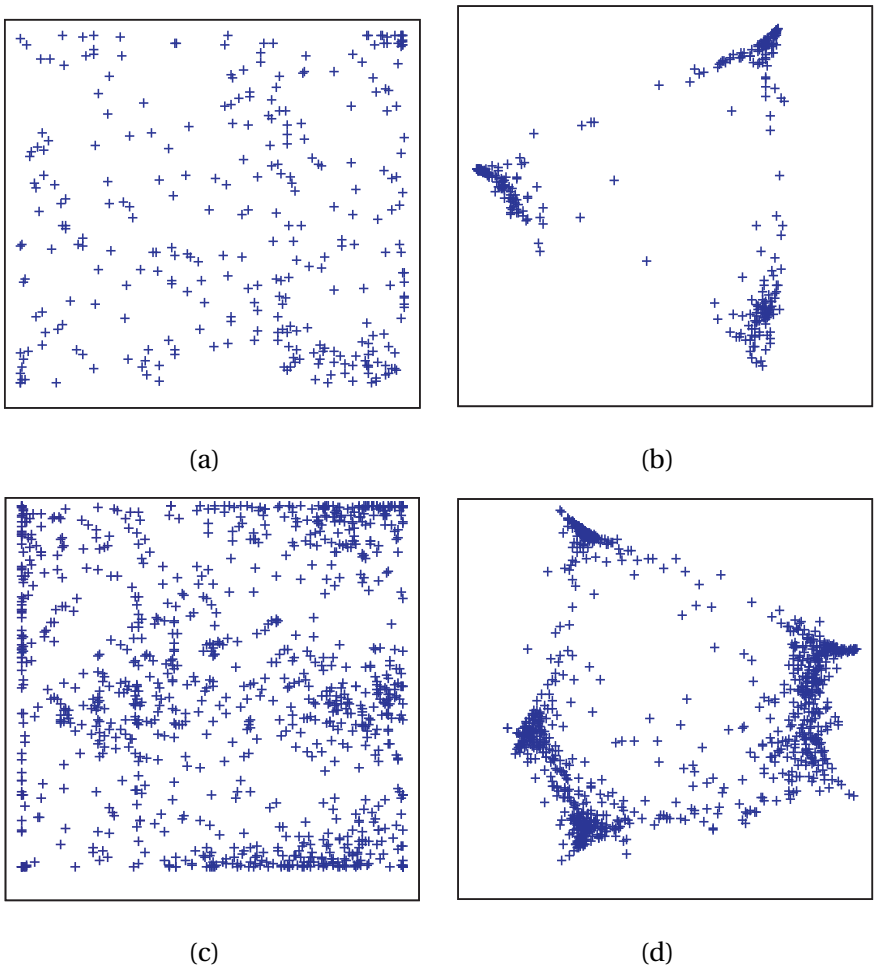


Figura 17. Característica de dispersión que identifica a las clases con el algoritmo VL-ZIP.

La identificación visual de las clases depende de la percepción de cada persona que las identifica. Cuando éstas son marcadas con diferente etiqueta y/o color, el ojo humano percibe la cercanía de las diferentes formas y colores para hacer la separación visual de las clases. Pero en escenarios reales, donde se desconoce la clase a la que pertenece cada documento, la visualización presenta la proyección con el mismo color y etiqueta. Por ejemplo, en la Figura 18 se presentan los resultados con los modelos Multinomial y VL-ZIP (con los datos de *S1* y *S4*) que proporcionaron mayor eficiencia en la proyección. Con la información de la característica mostrada en la Figura 17, es posible identificar las clases cuando el espacio que separa a dos de ellas es pequeño (Figura 18.d). Cuando las clases no poseen gran cantidad de palabras compartidas, el algoritmo VL-ZIP proyecta las clases con suficiente separación, de tal forma que no existe duda donde comienza una clase y termina la otra (Figura 18.b).

A continuación se presenta el análisis de documentos de las proyecciones obtenidas con nuestro algoritmo propuesto. El análisis se enfoca en los documentos que se proyectaron



(a) (b)

(c) (d)

Figura 18. Proyección de las BD *S1* y *S4* con etiquetas del mismo color y estilo: a) y c) con el modelo Multinomial, b) y d) con el algoritmo VL-ZIP.

en un lugar diferente a su clase, o están mezcladas en una clase a la que no pertenecen. Como se mencionó en el capítulo de "Descripción de experimentos"(Capítulo V), existen tres tipos diferentes de VA: los que poseen pocas palabras (VA tipo 1.a y 1.b), los que poseen demasiadas palabras compartidas y muy pocas palabras exclusivas de su clase (VA tipo 2) y los que pertenecen a una clase pero la mayoría de las palabras son exclusivas a otra clase.

VI.1.1. Análisis de VA de los resultados

En las Figuras 13.e, 14.d, 15.d y 16.d se muestran las proyecciones del algoritmo VL-ZIP. Algunos documentos que se encuentran ubicados en una clase a la que no pertenecen o

en medio de dos clases, son encerrados en una elipse y marcados para identificarlos [1].

En el conjunto de datos *S1* se encuentran ocho elementos, solo para esta BD se presentan las características numéricas de los documentos VA (Tabla XII). La primer columna especifica el conjunto de datos con posibles VA, la segunda columna indica la clase a la que pertenece el documento analizado. Las columnas tercera, cuarta y quinta son el número de palabras que pertenecen exclusivamente a una clase; mientras que la sexta columna pertenece a las palabras compartidas por las tres clases. Por otra parte, de la séptima a la novena columna pertenecen a las palabras compartidas por dos clases. Y la décima columna corresponde al total de palabras del documento después y de la selección de palabras, finalmente la décimo primera indica el total de palabras que tiene el documento considerando las de uso muy común y de poco uso. Con las dos últimas columnas se identifica si el documento es un VA tipo 1.a o 1.b.

Tabla XII: Características de los documentos proyectados con el algoritmo VL-ZIP, etiquetados como posibles VA del conjunto de datos *S1*.

Id.		**Exclusivas**			**Compartidas**				**Total-palabras**	
	Clase	Acq	Crude	Earn	Todas	Acq y Crude	Acq y Earn	Crude y Earn	Después de selec.	Antes de selec.
	Acq	3	7	1	13	4	5	7	40	67
	Acq	3	1	1	5	4	1	1	16	56
	Acq	9	4	2	19	5	7	5	51	31
	Acq	1	6	3	9	3	1	5	28	120
S1.1	Acq	1	3	0	7	1	1	1	14	187
	Acq	3	5	1	6	2	2	1	20	84
	Earn	2	6	2	17	2	3	10	42	67
	Earn	9	3	6	19	7	6	13	63	56
	Earn	1	10	6	16	3	1	15	52	31
	Earn	2	6	4	7	1	4	11	35	120
	Crude	0	4	1	3	1	0	3	12	67
S1.2	Earn	4	0	6	14	1	1	7	33	67

[1]La nomenclatura para nombrarlos consiste en dos partes: la primera identifica el conjunto de datos y la segunda un número consecutivo que identifica uno de otro.

Tabla XII: Continuación.

Id.		**Exclusivas**			**Compartidas**				**Total-palabras**	
	Clase	Acq	Crude	Earn	Todas	Acq y Crude	Acq y Earn	Crude y Earn	Después de selec.	Antes de selec.
	Earn	3	3	3	15	2	1	8	35	56
S1.3	Acq	2	4	0	8	5	0	1	20	67
	Acq	1	5	0	5	3	1	4	19	56
	Crude	0	2	1	4	2	0	4	13	67
S1.4	Earn	0	7	10	4	1	2	10	34	67
	Earn	2	2	4	7	3	0	9	27	56
	Acq	3	4	3	11	4	4	5	34	67
	Acq	3	3	6	10	4	2	4	32	56
	Acq	4	4	2	8	6	4	4	32	31
S1.5	Acq	2	4	1	9	2	1	4	23	120
	Acq	0	2	1	2	4	0	2	11	187
	Acq	1	1	1	6	4	2	2	17	84
	Acq	5	2	1	8	3	3	2	24	116
S1.6	Acq	1	4	2	3	2	2	2	16	67
	Earn	1	1	5	1	0	2	0	10	67
S1.7	Earn	1	1	4	3	0	4	1	14	56
	Earn	2	1	4	3	0	3	2	15	31
	Earn	4	1	4	2	0	6	3	20	67
	Earn	3	0	1	4	0	0	3	11	56
S1.8	Earn	3	0	3	3	0	4	2	15	31
	Earn	4	0	2	6	0	5	3	20	120
	Earn	3	1	4	7	1	6	5	27	187

Cuando se observa detalladamente los documentos localizados fuera de su clase, se puede entender por qué no están dentro de la clase a la que pertenecen. Del conjunto *S1* el grupo marcado como *S1.1* contiene 6 documentos *Acq* y 4 *Earn* dentro de la clase *Crude*. Dos de los documentos de la clase *Acq* poseen muchas palabras compartidas y pocas exclusivas por lo que se identifican como VA tipo 2, mientras que los otros tres poseen más palabras de la clase *Crude* que de su propia clase por lo que se consideran VA tipo 3. Tres de los documentos *Earn* por las palabras compartidas de las tres clases y en menor proporción

entre *Crude* y *Earn*, considerados VA tipo 2. Mientras que el otro documento *Earn* posee menor cantidad de palabras compartidas entre todos y prevalecen las palabras de *Crude* y las compartidas entre *Crude* y *Earn*, por lo que se considera VA tipo 3. El grupo etiquetado como *S1.2* se localiza entre la clase *Crude* y *Earn*; y contiene un documento de *Crude* y dos *Earn*. El documento de la primer clase mencionada, posee una palabra más del límite estadístico establecido para considerarse VA tipo 1[2] no existe evidencia suficiente de ser algún tipo de VA, solo se puede decir que es un documento con poca información para ubicarse cerca de su clase. Por otra parte en los otros dos documentos la mayoría de las palabras pertenecen a todas las clases, y se les identificó como VA tipo 2.

El grupo *S1.3*, se encuentra entre las clases *Crude* y *Acq* que contiene dos elementos de ésta última clase. De acuerdo al análisis, los dos documentos poseen mayor número de palabras de la clase *Crude*. El primero también posee gran cantidad de palabras compartidas por lo que se clasifica como VA tipo 2. Mientras que el segundo en menor cantidad también posee palabras compartidas, pero prevalecen las de la clase *Crude*; por lo que se considera que es un VA tipo 3. En el grupo *S1.4* cerca de la clase *Earn*, se encuentran dos documentos de esa misma clase y otro de *Crude*. El mayor número de palabras del documento *Crude* son compartidas por todas las clases y entre pares de clases, y no posee suficiente información (sin ser VA tipo 1) por lo que solo puede ser VA tipo 2. Por otra parte, un documento *Earn* contiene la misma cantidad de palabras exclusivas de su propia clase y de *Crude*; junto con las compartidas entre *Crude* y *Earn* por lo que se considera VA tipo 3. Mientras que el otro documento *Earn* posee información de todas las clases, por lo que se considera VA tipo 2. El grupo *S1.5* dentro de la clase *Earn* contiene 7 documentos *Acq*; de los cuales uno contiene pocas palabras por lo que se considera VA tipo 1.b. Otro de ellos contiene mayor información de la clase *Earn* y se considera VA tipo 3. Los cinco restantes comparten demasiada información de todas las clases y entre pares de clases, por lo que son considerados VA tipo 2.

El grupo *S1.6* contiene solo un documento de la clase *Acq* localizado cerca de la clase *Earn*; que posee demasiada información compartida y es considerado VA tipo 2. En el grupo *S1.7* localizado cerca de la clase *Acq* se identifican tres documentos *Earn* que poseen muy poca información, pero solo uno de ellos es considerado VA tipo 1.b. Mientras que los otros dos a pesar de poseer palabras exclusivas de su clase se encuentran lejos de ella. Finalmente el grupo *S1.8* localizado dentro de la clase *Acq* posee cinco documentos *Earn*, donde uno es

[2] 1 desviación estándar menos del promedio de palabras por documento, siendo que $\mu = 22{,}4$ y $\sigma^2 = 11{,}2$; por lo tanto *total de palabras* $\leq 11{,}2$ es VA tipo 1.

VA tipo 1.b. Dos de ellos poseen la misma cantidad de palabras exclusivas de su clase y de *Acq*; de tal forma que no son completamente VA tipo 3 pero poseen mucha similitud con esa clase. Los dos documentos restantes poseen más palabras exclusivas de *Acq* que de su misma clase (*Earn*) y también compartidas entre las dos, por lo que si se pueden considerar VA tipo3.

Como el conjunto de datos *S2* posee dos clases difíciles de separar y en la proyección se forman dos grupos con documentos entremezclados, dentro de éste no se identificó separación entre documentos *Mac* e *IBM*. En la Tabla VI se describe la relación de palabras y los documentos donde son mencionados. Las clases *IBM* y *Mac* poseen demasiadas palabras compartidas y su separación resulta complicada. Pero la clase *Med* es completamente separable y en el análisis de documentos se identificó solo un documento fuera de su grupo (el *S2.1*). *S2.1* posee solo 7 palabras, que son compartidas entre todas las clases y entre *IBM* y *Mac* por lo que se considera VA tipo 1.b. Por otra parte los grupos *S2.2* y *S2.3* poseen documentos de las dos clases difíciles de separar, el primer grupo contiene 46 documentos *IBM* y 37 *Mac*. Y el grupo *S2.3* contiene 49 documentos *IBM* y 51 *Mac*. Finalmente en el grupo *S2.4* se encuentran dos documentos *IBM* y cuatro *Mac* localizados dentro de la clase *Med*. Un documento *IBM* es VA tipo 1.b y el otro VA tipo 3 que además de poseer pocas palabras cercano al límite para ser considerado VA tipo 1, posee más palabras exclusivas de *Med*. Y de los cuatro documentos *Mac* dos son VA tipo 1.b y los otros dos son VA tipo 2.

El último conjunto de datos analizado es *S3* en el que se marcaron 6 grupos. El primero ubicado dentro de la clase *Medlars* se encuentran 7 documentos *Cranfield* cinco de ello identificados como VA tipo 2 y dos como VA tipo 3. Mientras que solo un documento pertenece a la clase *CISI* considerado VA tipo 1.a. El grupo *S3.2* localizado entre las clases *CISI* y *Medlars* contiene dos elementos *Cranfield* con muy poca información (4 y 2 palabras) uno de ellos VA tipo 1.a y el otro 1.b. Otro documento pertenece a la clase *CISI* también con 2 documentos del tipo 1.a. Y el documento *Medlar* corresponde a VA tipo 2. Dentro de la clase *Cranfield* se localiza el grupo *S3.3* que contiene dos documentos *CISI* identificados como VA tipo 2. Y cerca de la clase *Cranfield* se identifico el grupo *S3.4* que contiene 5 documentos *Cranfield* todos ellos VA tipo 1, uno de ellos es tipo a y los otros tipo b. El documento *CISI* también es identificado como VA tipo 1.a. Al final de la clase *CISI* se identificó el grupo *S3.5* que contiene tres documentos *Cranfield* uno es VA tipo 1.b, otro es tipo2 y otro tipo 3. Mientras que de la clase *CISI* se identificaron ocho elementos, uno de ellos VA tipo 1.a, cuatro tipo 1.b y tres tipo 2. Finalmente el grupo *S3.6* dentro de la clase

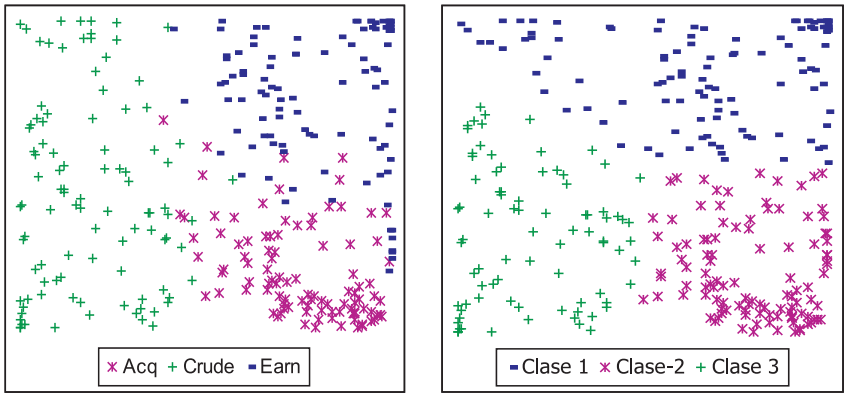
CISI contiene un documento *Cranfield* VA tipo 2. Y dos documentos *Medlar* VA tipo 2.

VI.2. Resultados del algoritmo de separación de clases

Con respecto a la separación de clases, en éste trabajo se construye un algoritmo de separación de clases para identificar los conglomerados y estructuras (grupos y VA) proyectados con el algoritmo VL-ZIP. En la mayoría de los casos también funciona para el modelo Multinomial, ya que ambos poseen la misma característica: las proyecciones de los datos se colocan alrededor del origen del plano cartesiano. Los resultados de separación se presentan de dos formas:

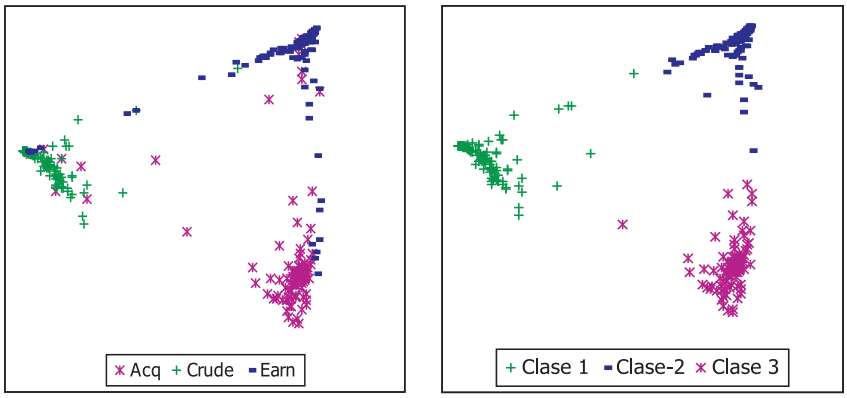
- Separación de los datos de tres clases. Las proyecciones de los conjuntos *S1*, *S2* y *S3* son separadas por clases con el algoritmo de separación; pero no se aplica la segunda parte del algoritmo consistente en re-entrenar los datos en pares de clases ó volver a entrenar solo una clase. El motivo es la gran cantidad de subclases que se obtienen cuando el conjunto de datos es pequeño.
- Separación de los datos del conjunto *S4*. Aquí se presenta la separación de las clases del conjunto de datos y se aplica la segunda parte del algoritmo propuesto; en dos formas diferentes: a) re-entrenamiento cada dos clases consecutivas; b) re-entrenamiento de los documentos de cada clase obtenida con el algoritmo de separación. En ambos casos se vuelven a separar los datos proyectados para identificar los datos que se pudieron haber mezclado debido a la selección de palabras de uso común.

El algoritmo de separación de clases utilizará la proyección obtenida con el algoritmo VL-ZIP y el Multinomial. A los resultados del algoritmo de separación en la primera etapa se les denomina clase y a los obtenidos después de la etapa de re-entrenamiento se les denominará sub-clase. En la Figura 19 se presenta la separación de clases del conjunto *S1* con el modelo Multinomial. La Figura *a* representa los datos originales, la *b* son las clases obtenidas con el algoritmo. Como se observa en la figura, la mayoría de los documentos de las tres clases fueron separados adecuadamente. En la *Clase 1* la mayoría de los documentos son *Earn*, se añadieron algunos elementos de *Crude*. Esto es porque los datos se encuentran proyectados más cercas de *Earn* que de su clase. Existen otros documentos que el algoritmo de visualización proyecta cercanos a una clase diferente a la que pertenecen; y el algoritmo de separación los clasifica dentro de otra clase. No se pueden considerar errores de separación, pues el resultado de la proyección los ubicó de esa forma, el algoritmo de visualización no proporciona suficiente espacio de separación entre clases. Por otra



(a) (b)

Figura 19. Separación de clases con el modelo Multinomial de la BD *S1*: a) clases originales, b) separación de clases con el algoritmo de separación de clases.

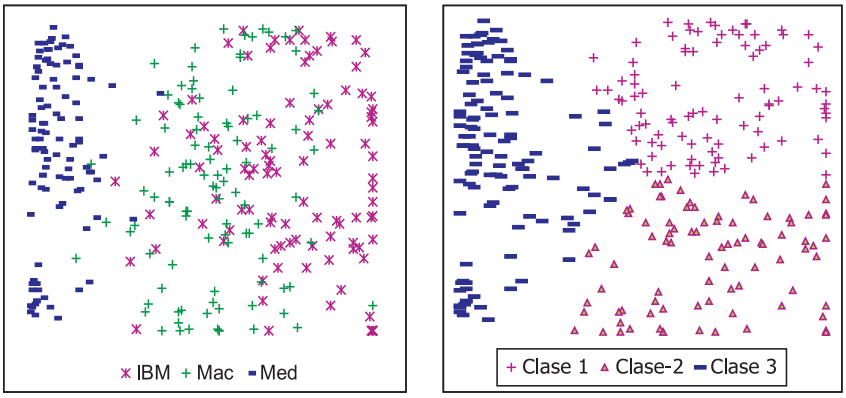


(a) (b)

Figura 20. Separación de clases con el algoritmo VL-ZIP de la BD *S1*: a) clases originales, b) separación de clases con el algoritmo de separación de clases.

parte, la separación de los datos proyectados con el algoritmo VL-ZIP no posee ese tipo de confusión en la separación de clases (Figura 20). Pues como se vio en los resultados de proyección de datos el algoritmo de visualización trata de separar las clases con el mayor espacio posible.

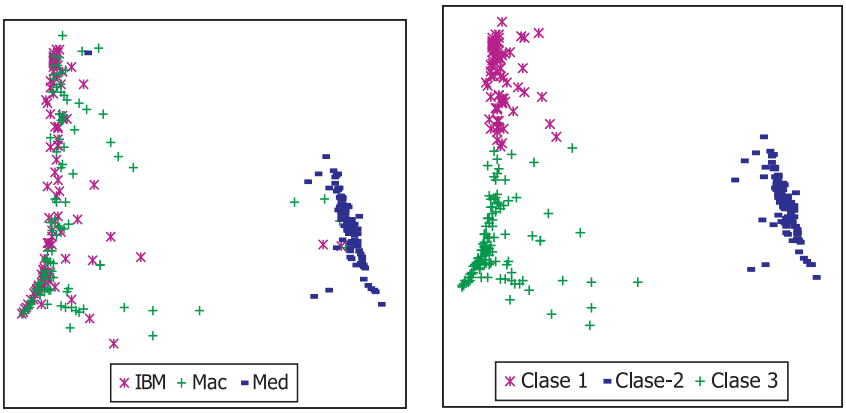
Al conjunto de datos *S2* también se le aplico el algoritmo de separación de clases. Los resultados con el modelo Multinomial se presentan en la Figura 21 y con el algoritmo VL-ZIP



(a) (b)

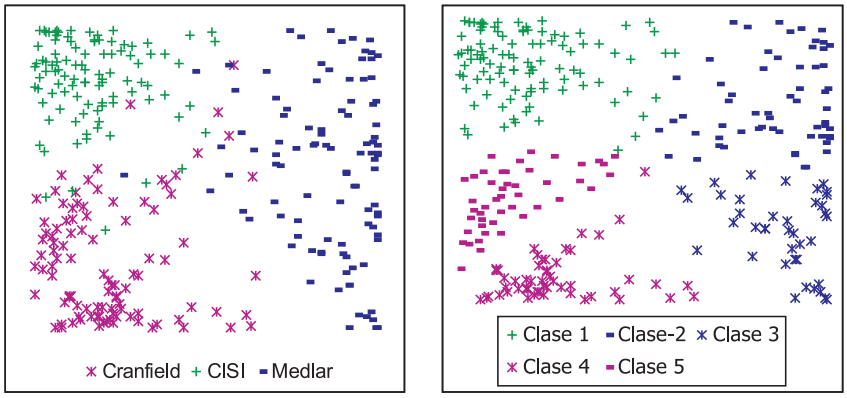
Figura 21. Separación de clases con el modelo Multinomial de la BD *S2*: a) clases originales, b) separación de clases con el algoritmo de separación de clases.

en la Figura 22. Con el conocimiento de que dos de las clases son difíciles de separar; con el algoritmo Multinomial (Figura 22) algunos datos *IBM* y *Mac* son proyectados cerca de *Med* por lo que se clasificaron en esa clase. Mientras que con el algoritmo VL-ZIP prevalece la tendencia a proyectar las clases de forma separada y el algoritmo de separación no tiene mayor problema.



(a) (b)

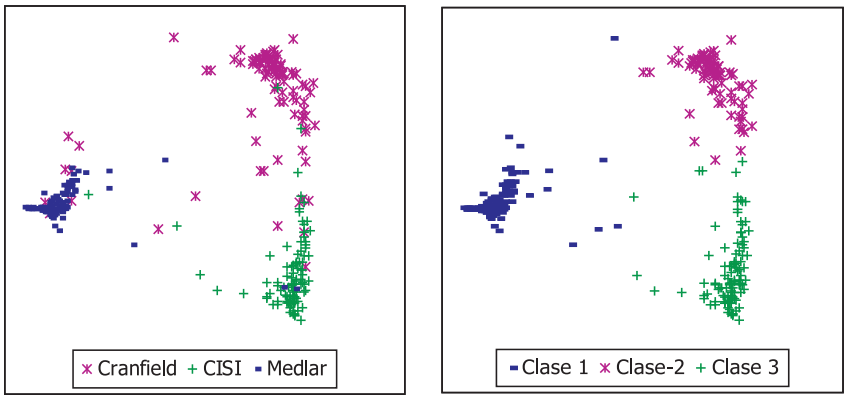
Figura 22. Separación de clases con el algoritmo VL-ZIP de la BD *S2*: a) clases originales, b) separación de clases con el algoritmo de separación de clases.



(a) (b)

Figura 23. Separación de clases con el modelo Multinomial de la BD *S3*: a) clases originales, b) separación de clases con el algoritmo de separación de clases.

La separación de clases del conjunto *S3* para los datos del modelo Multinomial se presenta en la Figura 23 y para los datos proyectados con el algoritmo VL-ZIP en la Figura 24.

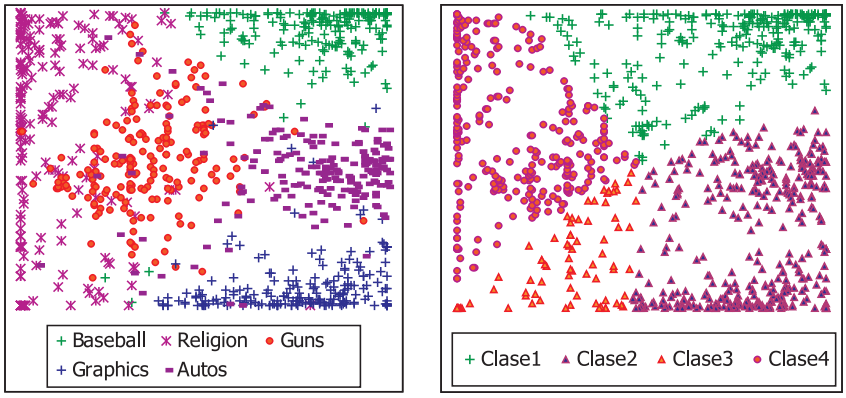


(a) (b)

Figura 24. Separación de clases con el algoritmo VL-ZIP de la BD *S3*: a) clases originales, b) separación de clases con el algoritmo de separación de clases.

La separación de clases se realiza según el número de documentos que existe en el espacio denominado $Rango\_s$. Si los datos de una clase se encuentran separados por un espacio sin documentos el algoritmo de separación los identifica como dos clases diferentes. Este es el caso de la proyección con el modelo Multinomial, las clases *Cranfield* y *Medlar* se

dividen en dos sub-grupos. Y el algoritmo de separación los identifica como cuatro clases diferentes, como se aprecia en la Figura 25. Por otra parte, con los datos del algoritmo VL-ZIP las tres clases son separadas sin problema alguno.



(a) (b)

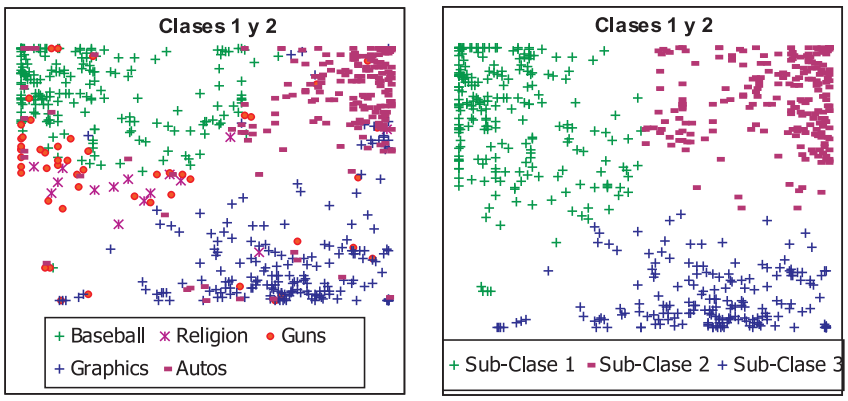
Figura 25. Separación de clases con el algoritmo Multinomial de la BD *S4* con cinco clases: a) clases originales, b) separación de clases con el algoritmo de separación de clases.

Como se mostró en los resultados anteriores, el algoritmo de separación de clases alcanzó su objetivo con la proyección de tres clases; por ser un conjunto pequeño de datos presentó las clases con suficiente espacio de separación. Para comprobar la eficiencia del algoritmo en situaciones de mayor complejidad, se aplica a la BD *S4* (que contiene 5 clases y doscientos documentos de cada una). Para cada conjunto de datos se aplica la fase denominada re-entrenamiento[3] con una y dos clases. Introducir una o dos clases al algoritmo de separación tiene un objetivo particular:

- Dos clases consecutivas en el plano cartesiano: Cuando dos clases no poseen suficiente espacio de separación entre ellas, o algunos elementos se encuentran muy mezclados puede ser que aislando los datos de las demás clases y re-entrenando se obtienen proyecciones que logre separar los datos. También existe la posibilidad de tener elementos entre dos clases y al volverlos a introducir al mismo proceso con mejores condiciones (eliminar las palabras de otras clases) se obtenga una mejor separación de clases.

[3]Volver a introducir los datos al algoritmo de visualización; en lugar de que el conjunto de datos sea toda la BD solo se introducen una o dos clases obtenida. Posteriormente se vuelve a aplicar el algoritmo de separación de clases.

- Una clase: Con ésta selección de datos, se pretende identificar sub-clases dentro de una misma clase y separar aquellos que pertenecen a otra. Al igual que en el caso anterior, se esperan mejores resultados al eliminar las palabras de otras clases.

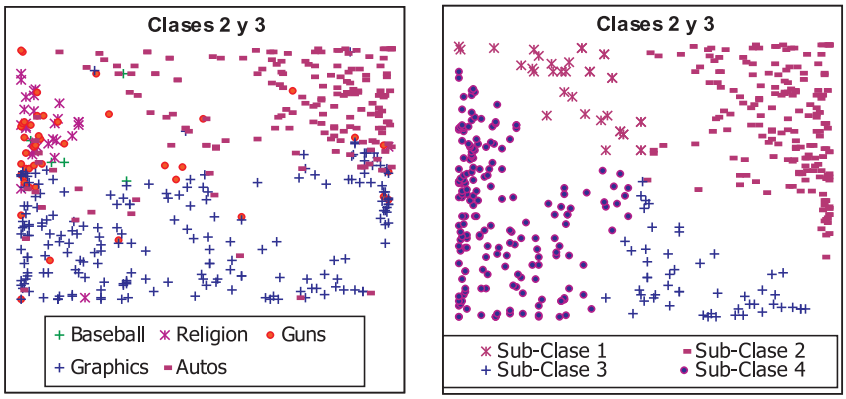


(a) (b)

Figura 26. Re-entrenamiento de la *Clases 1 y 2* con la proyección del modelo Mutinomial: a) clases reales, b) clases obtenidas.

En la Figura 25 se presentan los resultados de separación de las proyecciones obtenidas con el modelo Multinomial. En la Figura a se presentan las clases originales, y en la b la separación de las *Clases* obtenidas por el algoritmo de separación de clases. De las cinco clases existentes en el conjunto de datos, el algoritmo identifica cuatro. Esto es porque las clases *Graphics* y *Autos* se proyectan de forma paralela en el plano cartesiano y el algoritmo las considera una sola clase. La separación de clases se basa en la separaciones de líneas proyectadas alrededor del origen; considerando este tipo de situación se propone re-entrenar cada dos clases. Para reubicar las clases localizadas de forma paralela; es decir, separarlas. Las clases *Religion* y *Guns* también poseen algunas características de dispersión que las unen. Una parte de *Religion* se encuentra separada de su clase y pareciera que pertenece a *Guns*; además, la separación entre ellas es de forma diagonal, por lo que el algoritmo no identifica esto y las junta.

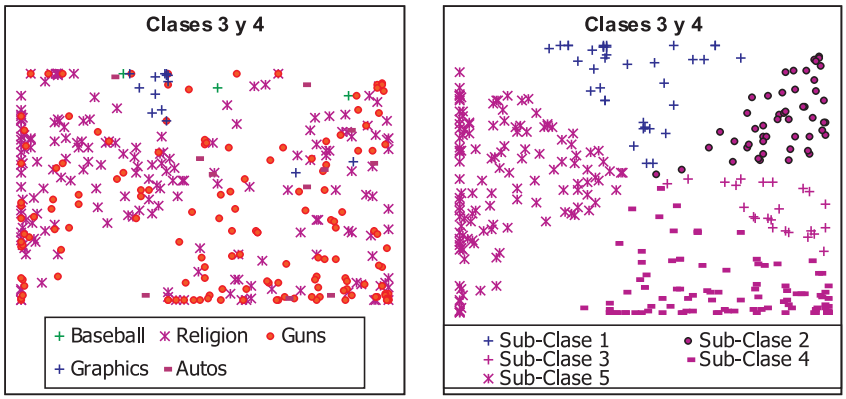
Para comprobar que las clases encontradas pertenecen solo a un tópico, se vuelve a entrenar con los datos de las *Clases* (*1 y* 2), (*2 y* 3), (*3 y* 4) y (*1 y* 4). Otro motivo para volver a entrenar con menos clases es que, si la proyección de una clase se encuentra separada el algoritmo no distingue que pertenecen a la misma clase. De la Figura 26 a la 29 se presentan los resultados del re-entrenamiento de dos en dos clases.



(a) (b)

Figura 27. Re-entrenamiento de la *Clases 2 y* 3 con la proyección del modelo Mutinomial: a) clases reales, b) clases obtenidas.

Los resultados del re-entrenamiento separan aquellas clases que se distinguen fácilmente

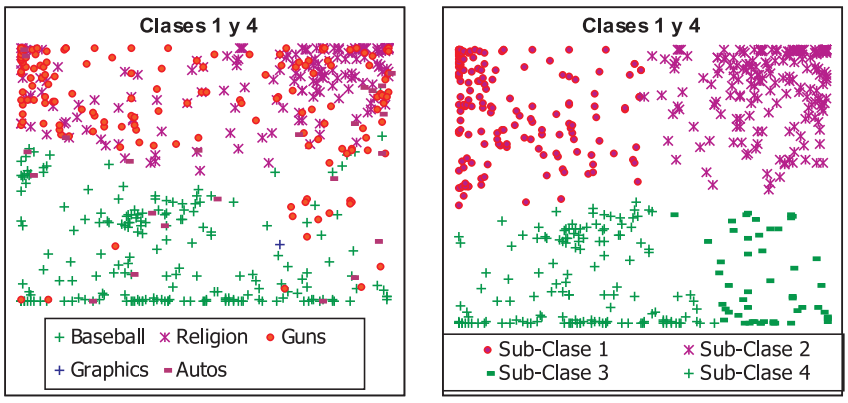


(a) (b)

Figura 28. Re-entrenamiento de la *Clases 3* y *4* con la proyección del modelo Mutinomial: a) clases reales, b) clases obtenidas.

de las otras; ya que, al haber menos documentos el espacio de separación entre las clases es mayor. Un aspecto importante a recalcar es que, si en algunos casos algunos documentos son ubicados en una clase diferente se debe a la similitud de palabras y al algoritmo de proyección. Como sucede en las Figuras 27, 28 y 29, donde se encuentran los documentos de las clases *Religion* y *Guns*. Estos poseen demasiadas palabras compartidas, y muy pocas palabras exclusivas de su clase. La mezcla de estos documentos solo sucede en ciertas re-

giones, por lo que se puede concluir que todos ellos se relacionan ampliamente por un conjunto de palabras.



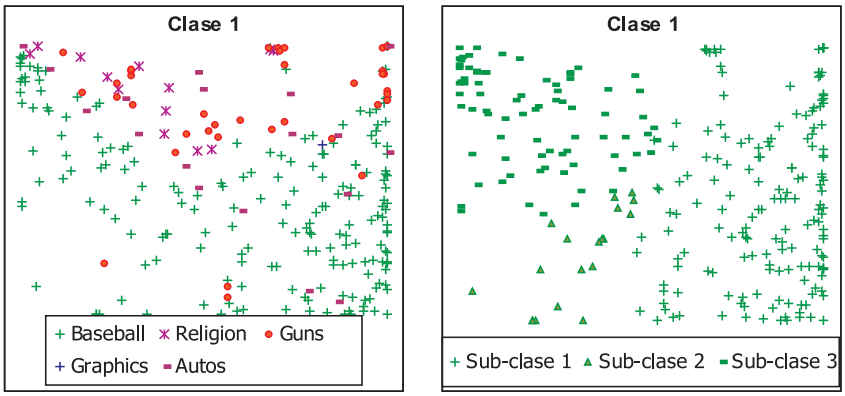
(a) (b)

Figura 29. Re-entrenamiento de la *Clases 1 y* 4 con la proyección del modelo Mutinomial: a) clases reales, b) clases obtenidas.

Otro conjunto formado por documentos de diferentes tópicos (*Baseball*, *Religion* y *Guns*) son los resultados del re-entrenamiento de las *Clases 2* y *3* (ver Figura 27). La cercanía de éstos documentos puede ser explicada por la gran cantidad de palabras compartidas por las clases (Tabla VI). Como se mencionó anteriormente, del análisis de las palabras se espera cierto tipo de resultados. De forma contraria a lo antes mencionado, las clases *Baseball*, *Graphics* y *Autos* son claramente separables. Solo algunos elementos se mezclan con otra clase, que no representan un problema de separación.

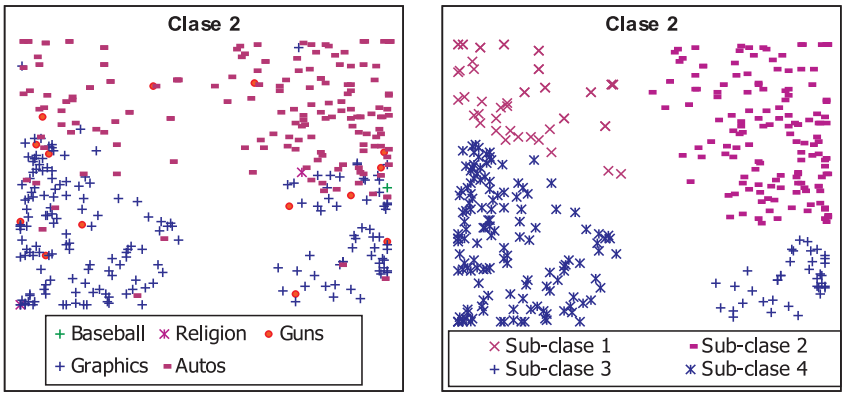
Continuando con el mismo conjunto de datos proyectados (resultados del modelo Multinomial), se aplica la segunda opción de re-entrenamiento de clases: una por una. En donde solo los datos que le corresponden a cada *Clase* identificada por el algoritmo de separación se entrenan nuevamente, y en a selección de palabras se elimina gran cantidad que no corresponden a la clase por lo que se esperan mejores resultados. Cuando el algoritmo de separación junta dos clases en una sola, separa los datos solo si los documentos poseen suficiente información y características para separarlos; si ambas clases comparten demasiadas palabras la separación será difícil.

De la Figura 30 a la 33 se presentan los resultados del re-entrenamiento de las *Clases* obtenidas. La *Clase 1* le corresponde a *Baseball* y algunos elementos de *Religion* y *Guns* como se



(a) (b)

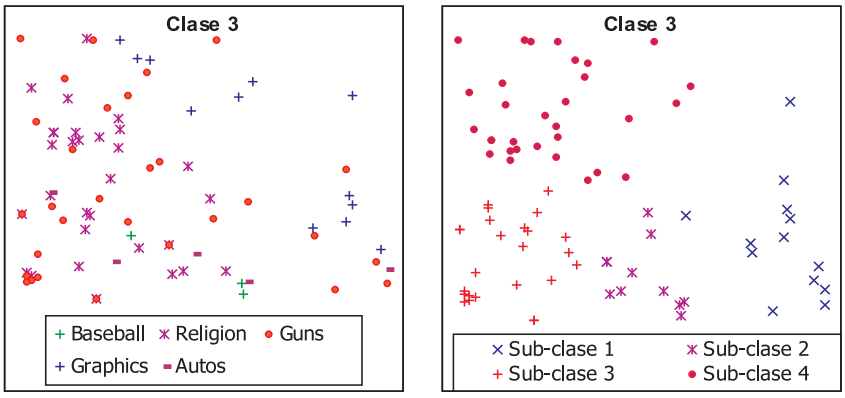
Figura 30. Re-entrenamiento de la *Clase 1* con la proyección del modelo Mutinomial: a) clases reales, b) clases obtenidas.



(a) (b)

Figura 31. Re-entrenamiento de la *Clase 2* con la proyección del modelo Mutinomial: a) clases reales, b) clases obtenidas.

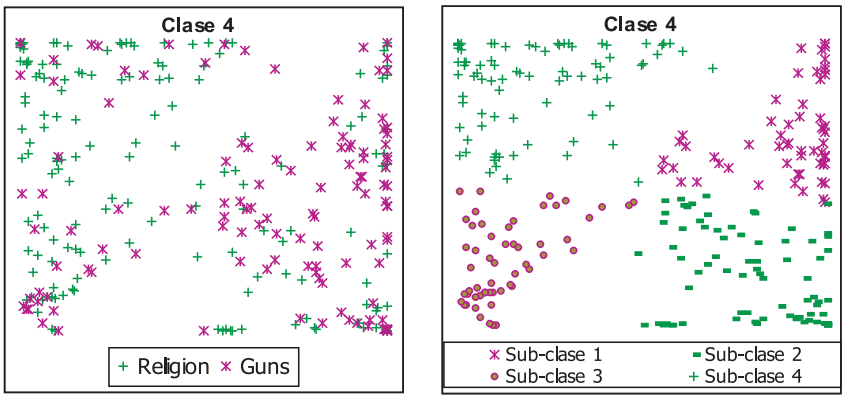
muestra en la Figura 19. El re-entrenamiento la *Clase 1* se divide en tres *Sub-clases* (Figura 30), en la marcada como *Sub-clase 3* se encuentra la mayor parte de documentos *Religion* y *Guns*, como se mencionó en el análisis de palabras (Tabla VI) las tres clases comparten información. La proyección del modelo Multinomial no produjo espacio entre los documentos. Por otra parte, los elementos que se encuentran en la orilla de las *Sub-clases 1* y *3*, son considerados VA por el algoritmo de separación propuesto y los agrega a la concentración de cúmulos más cercana.



(a) (b)

Figura 32. Re-entrenamiento de la *Clase 3* con la proyección del modelo Mutinomial: a) clases reales, b) clases obtenidas.

Por otra parte, la *Clase 2* separa adecuadamente los datos según la dispersión obtenida por el algoritmo. Como se mencionó anteriormente aquí se juntaron las clases *Graphics* y *Autos* debido a la ubicación de la proyección en el plano cartesiano. Sólo algunos de los elementos se encuentran mezclados fuera de su clase (*Sub-clases 2* y *4* de la Figura 31).

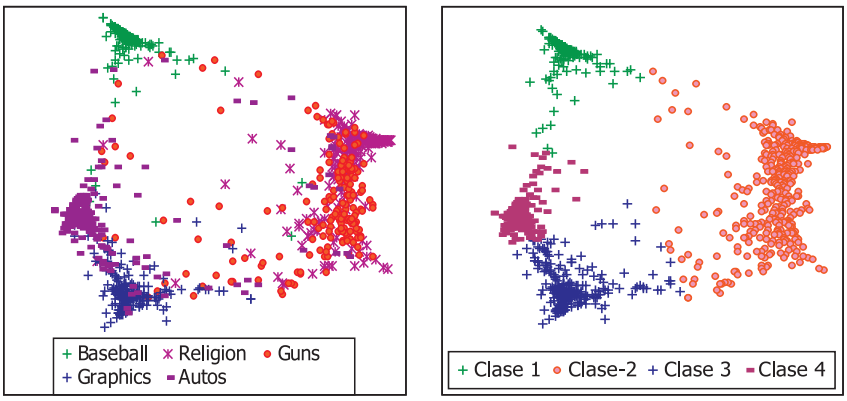


(a) (b)

Figura 33. Re-entrenamiento de la *Clase 4* con la proyección del modelo Mutinomial: a) clases reales, b) clases obtenidas.

La *Clase 3* mezcla algunos documentos *Religion, Graphics* y *Guns*. Pero al re-entrenar los datos no se logró obtener ningún grupo o sub-grupo con menos dispersión, por lo que se

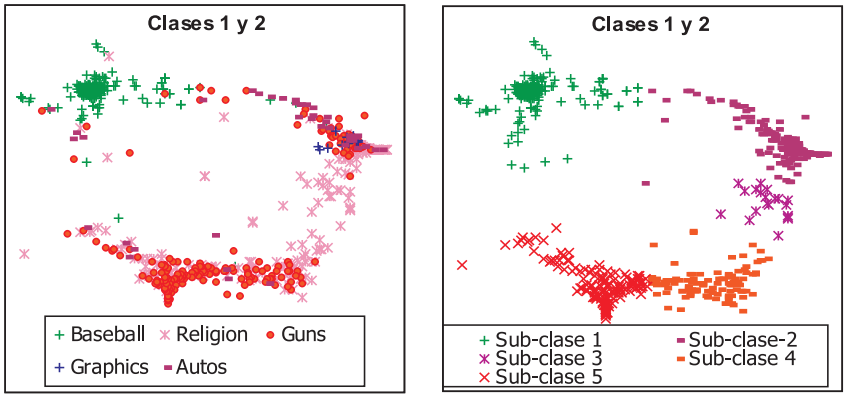
considera que solo se pueden obtener mejores resultados re-entrenando cada dos *Clases*. Finalmente la *Clase 4* que contiene las clases *Religion* y *Guns* al volverse a entrenar proyec-



(a) (b)

Figura 34. Resultados de la separación de clases con el algoritmo propuesto con los datos proyectados con el algoritmo VL-ZIP: a) clases reales, b) clases obtenidas.

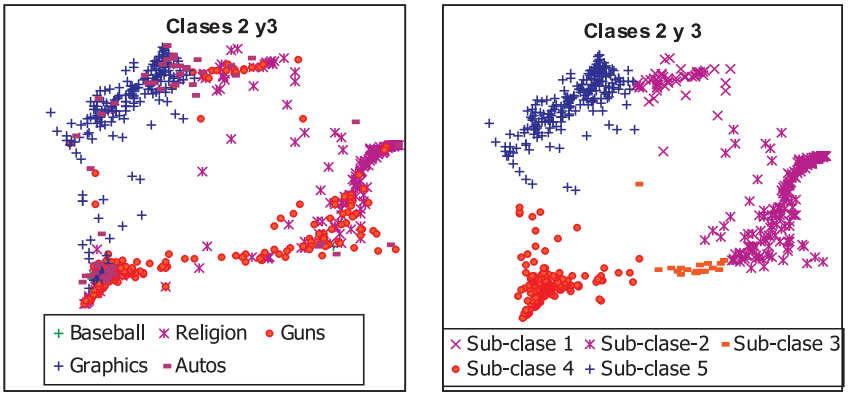
ta algunos datos mezclados. Las *Sub-clases 1* y *4* poseen menor cantidad de elementos mezclados, mientras que en la *2* y *3* los datos de las dos clases origen están muy mezclados. Según los resultados del análisis de palabras es lo que se esperaba; ya que, solo cierto sub-grupo de documentos aparecen traslapados (*Sub-clase 2* y *3*).



(a) (b)

Figura 35. Re-entrenamiento de las *Clases 1* y *2* con la proyección del algoritmo ZIP: a) clases reales, b) clases obtenidas.

Con las gráficas anteriores se terminó de presentar el algoritmo de separación de clases con las dos modalidades de re-entrenamiento (dos clases y una clase) para el conjunto de datos proyectados con el modelo Multinomial. A continuación se describen los resultados para los datos proyectados con el algoritmo de visualización VL-ZIP (fig 34). Como se vio en los resultados, proyecta los documentos que considera de una clase con poca dispersión (es decir, muy juntos) y con suficiente espacio de separación entre clases.

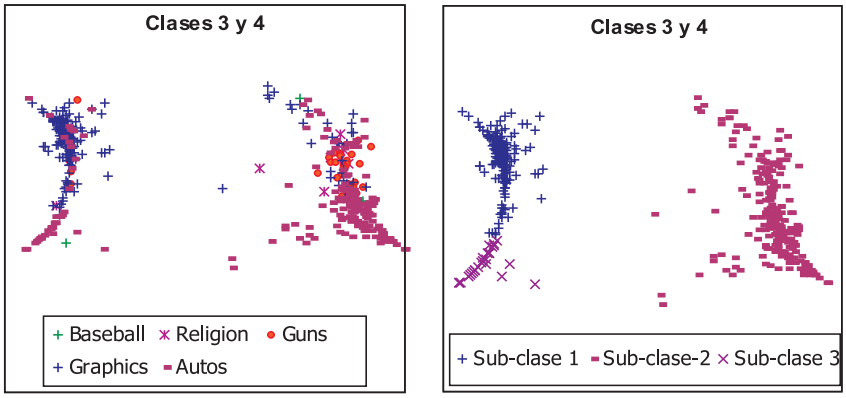


(a) (b)

Figura 36. Re-entrenamiento de las *Clases2* y *3* con la proyección del algoritmo ZIP: a) clases reales, b) clases obtenidas.

Retomando el análisis de palabras de la BD *S4* (Tabla VI), las clases *Religion* y *Guns* poseen demasiadas palabras en común por lo que su separación será difícil. En los resultados de proyección las dos clases aparecen muy juntas (Figura 34, *Clase 2*), y las otras tres clases son separadas adecuadamente.

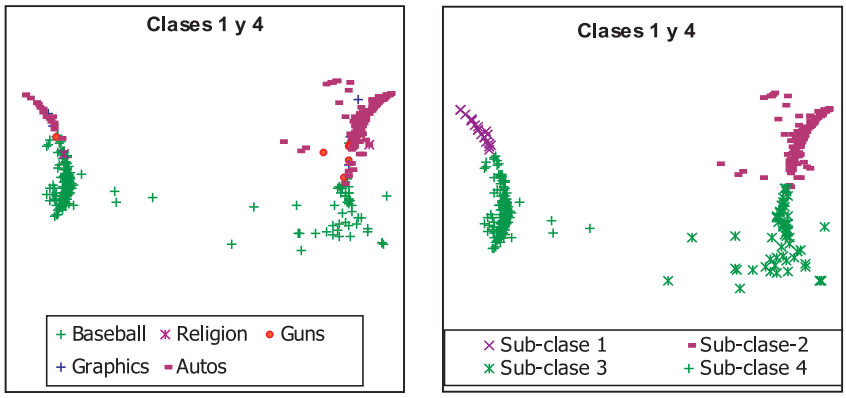
De la gráfica 35 a la 38 se presentan los resultado de proyección y separación de clases después de re-entrenar los datos de cada dos *Clases* obtenidas de la primer separación de clases (*Clases* (*1* y *2*), (*2* y *3*), (*3* y *4*) y (*1* y *4*)). El resultado del re-entrenamiento de las *Clases 1* y *2* se presenta en la gráfica 35, como se aprecia la clase *Baseball* es completamente separable sin problema alguno. Pero, con las clases *Religion* y *Guns* se forman diferentes grupos (*Sub-clases 3* y *4*); la *Sub-clase 3* corresponde completamente a documentos *Religion*, mientras que en *Sub-clase 4* existen algunos documentos *Guns* mezclados. En la *Sub-clase 5* la mayoría de los documentos son *Guns*, pero en la parte izquierda la mayoría de los documentos son de *Religion*. Todas las estructuras son formadas según la similitud



(a) (b)

Figura 37. Re-entrenamiento de las *Clases 3* y *4* con la proyección del algoritmo ZIP: a) clases reales, b) clases obtenidas.

de palabras.

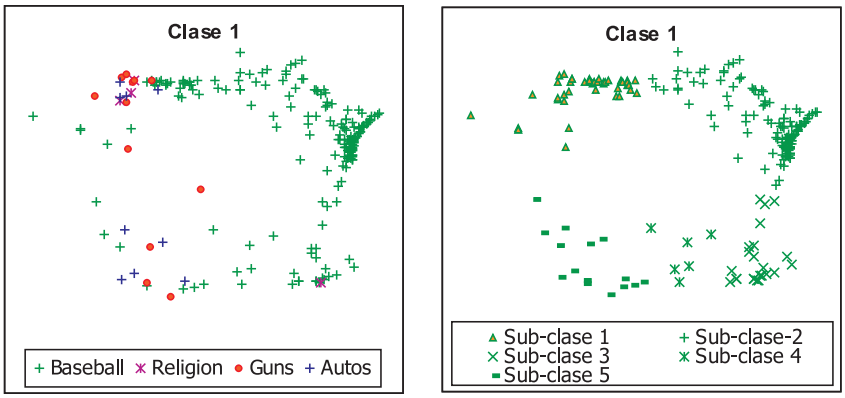


(a) (b)

Figura 38. Re-entrenamiento de las *Clases 1* y *4* con la proyección del algoritmo ZIP: a) clases reales, b) clases obtenidas.

En el siguiente conjunto de datos re-entrenados también se añade la *Clase 2* (dos clases juntas); y en la Figura 36 se muestra la proyección del re-entrenamiento junto con *Clase 3*. Donde se observa que la clase *Guns* posee mejor proyección que el resultado anterior (Figura 35); la mayoría de los documentos están en las *Sub-clases 3* y *4*, y pocos elementos de otras clases se mezclan. Las *Sub-clases 1* y *2* pertenecen a documentos *Religion*; en la

parte izquierda del segundo grupo se proyectan algunos documentos *Guns*. La mayoría de los documentos *Graphics* son proyectados dentro de la *Sub-clase 5*; solo algunos de ellos se ubican en *Sub-clase 4*.



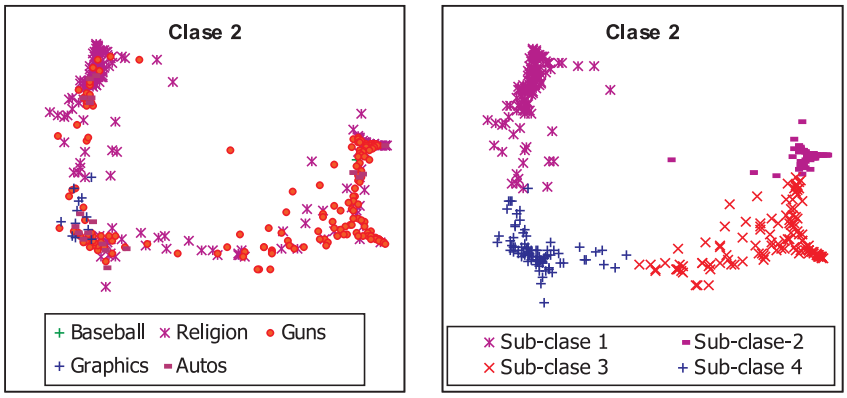
(a) (b)

Figura 39. Re-entrenamiento de la *Clase 1* con la proyección del algoritmo ZIP: a) clases reales, b) clases obtenidas.

En la Figura 37 se muestra la proyección obtenida después de re-entrenar las *Clases 3* y *4* (*Graphics* y *Autos*). Y como se aprecia la separación entre las clases es clara debido a que la proyección en la mayoría de los casos presenta las clases agrupadas adecuadamente. En la parte superior de la *Sub-clase 2* se encuentran algunos elementos de *Graphics* de forma mezclada. Mientras que cerca de la *Sub-clase 1* se forma un subgrupo de la clase *Autos*; que puede ser explicado por la cantidad de palabras compartidas entre ambas clases.

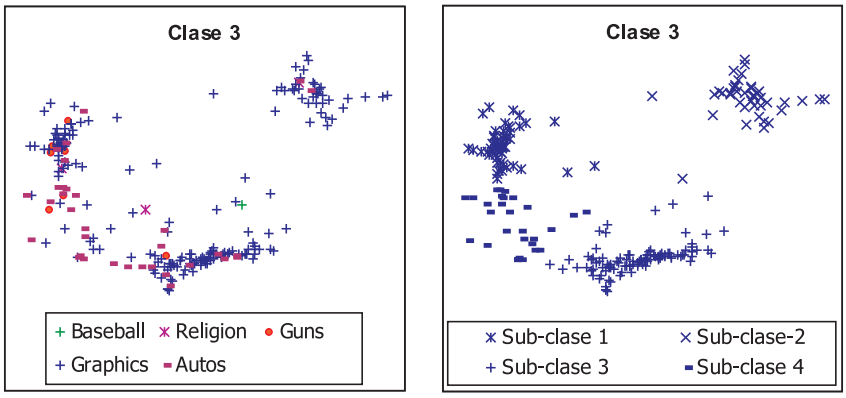
Finalmente en la Figura 38 se presenta los datos de las *Clases 1* y *4* (*Baseball* y *Autos*). La proyección presenta ambas clases divididas en dos *Sub-clases*; ambas son vecinas en las dos partes horizontales del plano cartesiano (x(-) y x(+)). La mayoría de los datos se agrupan con los de su misma clase, y pocos elementos se ubican dentro de la otra clase.
Por otra parte el re-entrenamiento con una sola clase se presentan de la Figura 39 a la 42. La primer Figura (Figura 34) corresponde a la *Clase 1*, que al re-entrenarse proyecta los datos en diferentes *Sub-clases*. La mayoría de los documentos corresponden a esa clase, y en la *Sub-clases 1* y *5* aparecen documentos de otras clases.

De la Figura 34 el grupo marcado como *Clase 2* se encuentran dos clases juntas. Que al re-entrenarse proyecta los datos de forma diferente a la obtenida en las Figura 35 y Figura



(a) (b)

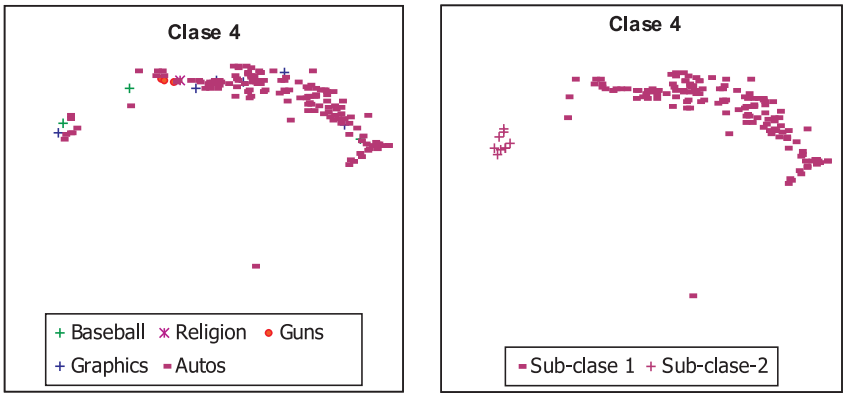
Figura 40. Re-entrenamiento de la *Clase 2* con la proyección del algoritmo ZIP: a) clases reales, b) clases obtenidas.



(a) (b)

Figura 41. Re-entrenamiento de la *Clase 3* con la proyección del algoritmo ZIP: a) clases reales, b) clases obtenidas.

36. De tal forma que cuando se eliminan las clases *Baseball* y *Graphics* los resultados de la proyección separa mayor cantidad de documentos en su propia clase y mezcla pocos documentos en una clase diferente. La *Sub-clase 1* posee la mayor cantidad de documentos *Religion*, mientras que la *Sub-clase 3* posee la mayor cantidad de documentos *Guns*. Pero la *Sub-clase 2* mezcla elementos de las dos clases mencionadas; y la *Sub-clase 4* esta formada por elementos de todas las clases.



(a) (b)

Figura 42. Re-entrenamiento de la *Clase 4* con la proyección del algoritmo ZIP: a) clases reales, b) clases obtenidas.

La Figura 41 presenta los resultados de re-entrenar la *Clase 3*, formada por la clase *Graphics* y pocos elementos de las otras clases. Como se observa en la *Sub-clase 4* se encuentran la mayor parte de los documentos que pertenecen a una clase diferente.

Y por último, los resultados de la proyección y separación de la *Clase 4* se presentan en la Figura 42, que presenta los resultados de clase sin problema alguno; porque existen pocos documentos de otras clases.

Estos fueron los últimos experimentos presentados de ambos algoritmos propuestos: VL-ZIP y el de separación de clases. Los resultados del algoritmo de separación de clases capturó el aspecto visual del VL-ZIP. Se considera que la separación de clases complementa el proceso de obtención de conglomerados e identificación de valores atípicos; presente de forma subjetiva (visualización que depende de la percepción humana). Existe gran acoplamiento entre los dos algoritmos, porque el de separación de datos trabaja bajo las características de proyección del de visualización. Cuando dos clases se proyectan de forma traslapada, la segunda parte del algoritmo de separación de clases separa todos aquellos elementos difíciles que no son considerados VA tipo3.

Por otra parte, el algoritmo de visualización de datos (proyección) ha mostrado ser eficiente en la visualización de las estructuras presentes en el conjunto de datos. Captura los elementos con gran similitud como un conglomerado; donde los componentes se muestran muy densos. Es decir, indica qué tan fuerte es la relación con cada dato vecino; y

cuando algunos de ellos son proyectados distantes se puede entender que la relación es débil. Además, cuando un documento o un conjunto pequeño posee características muy diferentes a las de su grupo los ubica lejos y dispersos (son identificados como VA). En los experimentos de los conjuntos *S1*, *S2* y *S3* con pocas clases; la separación de éstas fue muy marcada. No existió duda de que las estructuras señaladas pertenecen a documentos de una misma clase (documentos con fuerte relación entre las palabras utilizadas). Por otra parte en conjuntos de datos con más clases (*S4*) la separación fue adecuada, a excepción de dos clases que compartían demasiadas palabras como para ser separadas exitosamente (*Religion* y *Guns*). En los experimentos también se observó que, aun con la presencia de más clases cuando los documentos poseen suficientes palabras exclusivas no existe problema para identificar las clases.

También se observó un patrón de comportamiento en la dispersión de los datos que nos proporciona una idea visual para identificar las clases, el cual se presentó en la mayoría de los resultados.

Capítulo VII

Conclusiones

El objetivo principal de ésta tesis fue proponer un algoritmo de visualización para identificar las clases y los VA. El trabajo se extendió de tal modo que, además de visualizar las clases también se pudieran separar de forma automática y acceder a ellas por medio de una lista. Como conclusión general se puede decir que se alcanzaron los objetivos establecidos, y que, la medición de los resultados fue favorable en comparación con los modelos Gaussiano de Bishop *et al.* (1998), LDA de Blei *et al.* (2003) y Multinomial y Poisson de Kabán y Girolami (2001).

El producto final de ésta tesis de investigación fueron dos algoritmos: uno de visualización de datos en 2D y el otro para la separación de las clases de los datos obtenidos. De los resultados del algoritmo de separación de clases, se encontró que éste es un complemento para unir con el algoritmo de visualización y construir un sistema de análisis de conglomerado. Las características principales del algoritmo de visualización consisten en proyectar los datos en una elipse al rededor del origen del plano cartesiano. En las proyecciones se observa poca dispersión y mantiene espacio para identificar aglomerados. De forma adicional, se realizaron experimentos con el segundo algoritmo para los datos proyectados con el modelo Multinomial de Kabán y Girolami (2001). Para estos algoritmos, la característica de dispersión de clases alrededor del origen prevaleció en los resultados obtenidos; de tal forma que la mayoría de los documentos fueron separados de acuerdo a la clase a la que pertenecen, excepto en clases traslapadas o los VA. Con estos experimentos se comprobó que el algoritmo de visualización proporciona resultados con características de fácil separación.

Otro punto importante en la propuesta de esta tesis, fue el análisis de las palabras que proporcionó información importante para evaluar los resultados obtenidos y entender

cuándo un documento se proyectó fuera de su clase. Se constató que la proyección de los documentos está ampliamente relacionada con las palabras utilizadas, y que, los documentos atípicos no son errores de clasificación o ruido. Por lo que se obtuvo una visión diferente del ruido en documentos, en comparación con otro tipo de datos donde el ruido es un proceso aleatorio que se agrega a la variable de interés. El modelo de ruido aditivo no es adecuado en este trabajo, porque ya se realizó un proceso de filtrado de las palabras por medio del steming y el *resolving power*. En el segundo capítulo se habla extensamente del ruido y los VA; en el Capítulo V se definen algunas características que se encontraron en documentos diferentes (VA) a los de su categoría y se clasificaron en tres formas diferentes. Hay una diferencia marcada entre ruido y VA: ruido puede ser error de la medición (palabras mal escritas, modismos locales, abreviaciones equivocadas, entre otras) mientras que los VA son documentos diferentes a los de su clase según el uso de las palabras. Con respecto al ruido se concluye que es difícil su presencia en la matriz de frecuencia después de la fase de pre-procesamiento de datos; debido a que las palabras mal escritas en una colección de documentos son minoritarias y se encuentran en la sección C de la Figura 2. Por otra parte los VA son frecuentes en las BD y su presencia no afecta los resultados de dispersión o en la separación de clases.

En cuestión de documentos, cualquier cambio en la frecuencia tiene alto impacto; como se vio en el análisis de palabras de los conjuntos de datos utilizados. Existen diferentes categorías de palabras: las exclusivas de una clase y las compartidas. Las palabras exclusivas son las ideales para trabajar, pero en ocasiones el porcentaje de éstas suele ser muy pequeño, por lo que se requiere considerar otras palabras para evitar tener documentos vacíos. Las palabras compartidas se presentan en las diferentes combinaciones de clases (todas, la primera y la segunda, la primera y la tercera, etc.). El análisis de palabras nos indica que el ruido en los documentos es inherente, es decir que no representa una expresión estadística que pueda ser comprobada. El punto de confusión donde un documento está más relacionado con una clase que con otra, radica en el desequilibrio de palabras exclusivas, las compartidas y la cantidad de estas. Mientras que en el ruido estadístico es una porción numérica que también altera la percepción de los datos.

Con respecto a las palabras a utilizar en el algoritmo de visualización, se puede concluir que el resultado final cambia si se utilizan las palabras de uso muy común (que no son ”stop-words”) o palabras de poco uso (los dos extremos del histograma formado por: frecuencia de palabras $\times$ el rango). Y además es importante encontrar un punto donde el tamaño sea menor para evitar consumir demasiado tiempo computacional.

Otra aportación en ésta tesis, es la definición de tres condiciones que presentan algunos documentos atípicos: tipo 1 (pocas palabras), tipo 2 (la mayoría de las palabras son compartidas y casi nulas las exclusivas) y el tipo 3 (las palabras exclusivas de la clase con escasas y predominan las exclusivas de otra clase). De tal forma que, junto con el análisis de palabras es más fácil comprender por qué algunos documentos aparecen fuera de su clase y/o dentro de otra. La importancia de ésta definición se presenta en la interpretación de los datos. Como se vio en los resultados con diferentes modelos, la mayoría de ellos presentan agrupación de documentos de una misma clase en mayor o menor cantidad. La principal diferencia entre ellos radica en la forma de dispersión de los datos y la separación entre las clases. Pero todos en menor o mayor grado presentan los datos en conglomerados con elementos de la misma clase. Pero, también poseen documentos fuera de su clase y dispersos en cualquier parte del plano cartesiano, los cuales generalmente se pueden entender según las definición de las tres condiciones de VA.

Con respecto al algoritmo de visualización basado en los experimentos, se concluye que:

- En conjuntos con pocas clases, el algoritmo proporcionó datos completamente separables ya que el espacio entre clases era muy grande y no había duda de donde comienza y termina una.
- En el conjunto de datos con mayor número de clases la visualización identificó fácilmente a los documentos de las clases que poseían suficiente información para ser separados.
- En dos conjuntos de datos (*S2* y *S4*) el algoritmo de visualización proyectó juntas a dos clases, debido a la gran cantidad de palabras compartidas entre ambos grupos. Además, los modelos con los que se compararon resultados también obtuvieron clases mezcladas.
- En la mayoría de los resultados el índice de preservación de la topología (E_{SS}) resultó ser menor con el algoritmo VL-ZIP, y el que más se acercó a la eficiencia mostrada fue el modelo Multinomial.
- De todos los resultados de visualización presentados, nuestro algoritmo proyectó los datos con mejor dispersión (mayor S_F).
- Se identificó una característica de dispersión de los datos que ayuda a identificar visualmente un conglomerado de otro (Fig. 17).

Por otra parte, del algoritmo de separación de datos se puede decir que las clases encontradas en los resultados de proyección fueron separadas adecuadamente. Los elementos mal clasificados no son errores de separación de clases, solo son documentos con características diferentes a los de su clase o que poseen mayor similitud con el grupo en el que fueron ubicados. Los datos se proyectan según la similitud de las palabras que los unen, entonces se dice que existen documentos atípicos diferentes a los de su clase.

El proceso de re-entrenamiento en la separación de clases fue de gran utilidad para confirmar la separación de clases en BD de tamaño mayor a 500 documentos. Las dos propuestas de re-entrenamiento son adecuadas en diferentes condiciones. Cuando existen diversos documentos en medio de dos clases el re-entrenamiento con dos clases es el adecuado; ya que al juntar las dos clases existe la posibilidad de que los elementos entre ellas se ubiquen a una clase. Cuando el algoritmo de separación de clases identificó a una con mayor número de documentos puede ser el indicativo de que hay dos clases mezcladas, en donde el re-entrenamiento de una sola clase es el adecuado. Por otra parte, cuando una subclase es pequeña (menor de 150 documentos) no es conveniente utilizar el proceso de re-entrenamiento.

En los experimentos realizados con la BD *S4* se pudo ver que el proceso de re-entrenamiento obtuvo mejores estructuras, tal es el caso de de las clases *Religion* y *Guns*. Por lo que se concluye que el algoritmo de separación de clases con re-entrenamiento proporcionó buenos resultados no solo con los datos proyectados con nuestro algoritmo de separación, también lo hizo con la proyección de otro modelo.

De forma general, se concluye que los dos algoritmos propuestos permiten separar documentos e identificar las estructuras en una BD (conglomerados y VA). Además, el algoritmo de visualización de datos proporciona datos menos dispersos y con mejor separación entre clases en la mayoría de los experimentos.

VII.1. Trabajo a futuro

Como se mencionó en los párrafos anteriores, se considera trabajar con un modelo de evaluación de palabras sin conocimiento a priori. Por medio de una matriz de similitud que relacione las palabras y documentos, y al encontrar coincidencias se identifiquen las palabras como exclusivas de esos documentos. De la misma forma obtener las compartidas

tomando como base los documentos identificados anteriormente.

Con respecto al algoritmo de visualización de datos, hace falta realizar experimentos para encontrar adecuadamente los valores de los parámetros η_1 y η_2 y trabajar con la visualización en un plano 3D para ver si existe mejor separación de clases traslapadas. Es posible que dos clases que aparentan estar proyectadas en el mismo espacio, se encuentren ubicadas en la misma altura pero que posean diferente profundidad.

Respecto al procedimiento de separación de clases, se requiere encontrar otra función de densidad y un método para unir los rangos más densos. De tal forma que el algoritmo pueda obtener los elementos de un conglomerado percibiendo los puntos de mayor concentración de elementos y expandirlo como una mancha según la cercanía de los datos. Con respecto a la definición de VA propuesta, se requiere de un método para identificar los elementos fuera de un conglomerado y etiquetarlos según el tipo.

Bibliografía

Barnett, V. y L. Toby, 1991. *Outliers in statistical data*. John Wiley. 584 pp.

Barquin, R. y H. Edelstein, 1997. *Building, using and managing the data warehouse*. Prentice Hall PTR. 317 pp.

Bishop, C., M. Svensén, y C. Williams, 1998. GTM: the generative topographic mapping. *Neural computation*, 10:215–234 p.

Blei, D., A. Ng, y M. Jordan, 2003. Latent dirichlet allocation. *Journal of machine learning research*, 3:993–1022 p.

Calafiore, G., 2002. Approximation of n-dimensional data using spherical and ellipsoidal primitives. *EEE transactions on Systems, Man and Cybernetics, Part A: systems and humans*, 32(2):269–278 p.

Castellanos, M., 2004. Hotminer: Discovering hot topics from dirty text. *Survey of Text Mining: Clusering, classification, and retrieval*, 1(1):123–158 p.

Chakrabarti, S., B. Dom, R. Agrawal, y P. Raghavan, 1999. Using taxonomy, discriminants, and signatures for navigating in text databases. *I*, 12(1):1229–1252 p.

Cover, T. M. y J. A. Thomas, 1991. *Elements of information theory*. John Wiley and Sons. 542 pp.

Darnton, R., 2008. The library in the new age. *The New York Review of Books*, 55(10).

Dempster, A., N. Larid, y D. Rubin, 1976. Maximun likelihood from incomplete data via the em algorithm (with discusion). *Journal of the Royal Statistical Society: Series B (Statistical metodology)*, 39(1):1–38 p.

Devlin, B., 1997. *Data warehouse from architectured to implementation*. Addison-Wesley. 432 pp.

do Prado, H. A. y E. Ferneda, 2008. *Emerging technologies of text mining: Techniques and applicaions*. Information Science reference. 358 pp.

Dobson, A., 2002. *An introduction to generalized linear models*. Chapman and Hall/CRC. Segunda edición, 226 pp.

Duda, R. O., P. E. Hart, y D. Stork, 2001. *Pattern classification*. John Wiley and Sons. Segunda edición, 654 pp.

Fayyad, U., G. Piatetsky-Shapiro, P. Smyth, y R. Uthurusamy, 1996. *Advances in knowledge discovery and data mining*. AAAI Press. 611 pp.

Feldman, R. y J. Sanger, 2006. *The text mining handbook: Advanced approach in analyzing unstructured data*. Cambridge University Press. 410 pp.

Frigui, H. y O. Nasraoui, 2004. Simultaneous clustering and dynamic keyword weighting for text document. *Survey of Text Mining: Clusering, classification, and retrieval*, 1(1):1–38 p.

Girolami, M., 2001. The topographic organization and visualization of binary data using multivariate-bernoulli latent variable models. *IEEE Transactions on Neural Networks*, 12(6):1367-1374 p.

Guha, S., R. Rastogi, y K. Shim, 1998. Cure: an efficient clustering algorithm for large databases. *Haas, L. M. y A. Tiwary, editores, ACM SIGMOD/PODS 1998: Proceedings of International Conference on Management of Data*, Seatltle, Washington. ACM Press. 73–84 p.

Halkidi, M., Y. Batistakis, y M. Vazirgiannis, 2001. On clustering validation techniques. *J. Intell. Inf. Syst.*, 17(2-3):107–145 p.

Hampel, F. R., E. Ronchetti, P. Rousseeuw, y W. Stahel, 1986. *Robust statistics : the approach based on influence functions*. John Wiley and Sons. 470 pp.

Han, J. y M. Kamber, 2000. *Data mining: Concepts and techniques*. Morgan Kaufmann publishers. 550 pp.

Hawkins, D. M., 1980. *Identification of outliers*. Chapman and Hall. 188 pp.

Honkela, T., S. Kaski, K. Lagus, y T. Kohonen, 1996. Exploration of full-text databases with self-organizing maps. *ICNN96: Proceedings of the IEEE International Conference on Neural Networks*, Washington D.C., USA. Association for Computational Linguistics, IEEE Press. 56–61 p.

Jain, A. y R. C. Dubes, 1988. *Algorithms for clustering data*. Pretince Hall. 304 pp.

Kabán, A. y M. Girolami, 2001. A combined latent class and trait model for the analysis and visualization of discrete data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):859–872 p.

Kaski, S., T. Honkela, K. Lagus, y T. Kohonen, 1996. Creating an order in digital libraries with self-organizing maps. *WCNN'96: Proceedings of World Congress on Neural Networks*, Washington D.C., USA. Lawrence Erlbaum and INNS Press. 814–817 p.

Kaufman, L. y P. J. Rousseeuw, 1990. *Finding groups in data: An introduction in cluster analysis*. Wiley-Interscience. 368 pp.

Kennedy, R. L., Y. Lee, B. V. Roy, C. D. Reed, y R. P. Lippmann, 1997. *Solving data mining problems through pattern recognition*. Prentice Hall. 400 pp.

Knorr, E. M., R. T. Ng, y V. Tucakov, 2000. Distance-based outliers: algorithms and applications. *The international Journal of Very Large Data Bases*, 8(3-4):237–253 p.

Kohonen, T., 1988. *Self-organization and associative memory*. Springer Verlag. Segunda edición, 312 pp.

Kohonen, T., S. Kaski, K. Lagus, y T. Honkela, 1996. Very large two-level som for the browsing of newsgroups. *ICANN96: Proceedings of International Conference on Artificial Neural Networks*, Berlin. Springer. 269–274 p.

Kruskal, J. B., 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27 p.

Kumar, V. y M. Joshi, 1998. Tutorial on high performance data mining. *HiPC'98: Proceedings of the Fifth International Conference on High Performance Computing*, Chennai, India.

Lagus, K., T. Honkela, S. Kaski, y T. Kohonen, 1996. Self-organizing maps of document collections: A new approach to interactive exploration. *KDD-96: Proceedings ofthe Second International Conference on Knowledge Discovery and Data Mining*, Menlo Park, California. AAAI Press. 238–243 p.

Lambert, D., 1992. Zero-inflated poisson regresion, with an application to defects in manufacturing. *Technometrics*, 34(1):1–14 p.

Li, J. y H. Zha, 2004. Two-way Poisson mixture models for simultaneous document classification and word clustering. *J. Kittler, M. Petrou, M. S.Ñ., editor, ICPR2004: Proceedings of the 17th International conference on pattern recognition*, Cambridge, U.K. IEE CS Press.

Liu, X., G. Cheng, y J. X. Wu, 2002. Analyzing outliers cautiously. *IEEE Transaction Knowledge Data Engineering*, 14(2):432–437 p.

Mao, J. y A. K. Jain, 1993. Discriminant analysis neural networks. *IJCNN-93: Proceedings of the IEEE International Joint Conference on Neural Networks*, San Francisco, USA. IEEE. 300–305 p.

McCullagh, P. y J. Nelder, 1989. *Generalized linear models*. Champman and Hall/CRC. Segunda edición, 511 pp.

Miikkulainen, R., 1993. *Subsymbolic natural language processing: an integrated model of scripts, lexicon and memory*. MIT Press. 403 pp.

Norton, M. J., 1996. Short takes in the digital revolution. *Bulletin of the American Society for Information Science*, 22(6):19–21 p.

Peters, C. y C. H. Koster, 2002. Uncertainty-based noise reduction and term selection in text categorization. *ECIR-02: Proceedings of 24th European Colloquium on Information Retrieval Research*, London. Springer Verlag. 248-267 p.

Porter, M. F., 1980. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137 p.

Qian, Y., Q. Shi, y Q. Wang, 2002. Cure-ns: a hierarchical clustering algorithm with new shrinking scheme. *ICMLC 2002: Proceedings IEEE of the First International Conference on Machine Learning and Cybernetics*, Beijing, China, volume 12. 895–899 p.

Ramírez, E. B., 2002. *Comunicaci/ón y cultura en la era digital*. Gedisa. 384 pp.

Ritter, H. y T. Kohonen, 1989. Self organizing semantic maps. *Biological Cybernetics*, 61:241-254 p.

Rockwood, A. P., 1997. Geometric primitives. *The Computer Science and Engineering Handbook*. CRC-Press, 1212-1225 p.

Salton, G. y M. J. McGill, 1983. *Introduction to modern information retrieval*. McGraww Hill. 448 pp.

Sammon, J. W., 1969. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(1):404-409 p.

Svensen, M., 1998. *Gtm:the generative topographic mapping*. Tesis de Doctorado, Aston University, Birmingham, U.K. 111 pp.

Tinŏ, P. y I. T. Nabney, 2002. Hierarchical gtm: constructing localized nonlinear projection manifolds in a principled way. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):639–656 p.

Turaisingham, B., 1999. *Mining: Technologies, techniques, tools, and trends*. CRC Press. 270 pp.

Vellido, A. y P. Lisboa, 2006. Handling outliers in brain tumor mrs data analysis through robust topographic mapping. *Computers in Biology and Medicine*, 10(36):1049–1063 p.

Wang, H., W. Wang, J. Yang, y P. S. Yu, 2002. Clustering by pattern similarity in large data sets. *ACM SIGMOD/PODS 2002: Proceedings of International Conference on Management of Data*, Wisconsin, USA. Madison. 394–405 p.

Wedel, M., W. S. Desarbo, y V. Ramaswamy, 1993. A latent class poisson regression model for heterogeneous count data. *Journal of applied enonometrics*, 8(1):397–411 p.

Yanchang, Z. y S. Junde, 2001. GDILC: a grid-based density-isoline clustering algorithm. *IEE ICII 2001: Proceedings of International Conferences on Info-tech and Info-net*, Beijing, China, volume 1.3. 140–145 p.

Yang, J. y B. T. Zhang, 1995. Noise reduction in a statistical approach to text categorization. *Fox, E. A., P. Ingwersen, y R. Fidel, editores, SIGIR-95: Proceedings of 18th ACM International Conference on Research and Development in Information Retrieval*, New York. ACM Press. 256-263 p.

Yang, J. y B. T. Zhang, 2001. Customer data mining and visualization by generative topographic mapping methods. *VDM@PKDD 2001: Proceedings of the International Workshop on Visual Data Mining*, Freiburg. Springer. 55–66 p.