Benemérita Universidad Autónoma de Puebla

Estudio Comparativo entre Algoritmos de Clasificación Automática usando Simulaciones con el paquete WEKA

Trabajo de Tesis
presentado a la
Facultad de Ciencias de la Computación
por

José Santiago Capilla Ávila

Asesor: Hortensia J. Reyes Cervantes
Asesor: Gladys Linares Fleites
Coasesor: Lourdes Sandoval Solís

Para optar al Título de
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

Puebla,Puebla
2010

AGRADECIMIENTOS

A Dios, por darme la oportunidad al darme vida y fuerza.

Deseo agradecer a mis padres por ser la base fundamental en mi vida.

Deseo expresar mi más sincero agradecimiento a mi asesora de tesis Doctora Hortensia J. Reyes Cervantes por su apoyo y orientación constante durante la elaboración de mi tesis.

#65279;A PROMEP/103.5/08/3332 por otorgarme una beca para realizar mi tesis ya que sin ella no hubiera sido posible.

A mis hermanas por el cariño que me han mostrado en todo momento y ayudarme a ser mejor persona.

A todos mis amigos por su amistad invaluable y su apoyo en los momentos difíciles.

Al Director de la Facultad de Ciencias de Computación de la Benemérita Universidad Autónoma de Puebla (BUAP) por su apoyo.

A la Dra. Gladys Linares Fleites por su ayuda, para el mejoramiento de esta tesis.

A la Dra Lourdes Sandoval Solis Coasesora de mi tesis por sus comentarios para mejorar la tesis.

DEDICATORIAS

Dedico todo mi trabajo, a mi esposa Zenaida, por su amor, apoyo y compañía en los momentos difíciles y a mi hijo el cual es la personita que me impulsa.

Índice

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

En el mundo moderno existen muchas bases de datos de diferentes situaciones reales, que en cualquier computadora se pueden acceder a ellas. Esta información está compuesta de grandes volúmenes de registros que hacen imposible manejarlas. Si un investigador de cualquier área desea realizar algún análisis para tomar decisiones sobre agrupaciones de datos, y no se cuenta con alguna metodología, poco se puede realizar. En muy pocas ocasiones puede suceder que se tenga información de las relaciones intrínsecas que guardan los datos, para que el investigador pueda encontrar una buena clasificación. En la mayoría de los casos no hay antecedentes, por lo cual se debe utilizar herramientas computacionales que le ayuden a obtener información de los datos, tratando de identificarlos de alguna manera. Estas técnicas, generalmente, no usan los mismos criterios y terminan dando estructuras diferentes y, por lo tanto, diferentes interpretaciones de los resultados. Luego aquí, se tiene un problema de decisión, ***¿cual criterio escoger para tomar el mejor agrupamiento?***.

En este trabajo se presenta un algoritmo computacional basado en el paquete RWEKA, que ayuda al investigador a decidir cual método de clasificación seleccionar cuando tiene sus corridas. A partir de la metodología propuesta por [Von A. y Mair P. (2008)], se persigue el objetivo de obtener una clasificación de objetos evaluando los algoritmos utilizados, conforme a los resultados obtenidos. Los algoritmos utilizados para este fín son de clasificación no supervisada, también conocidos como "clustering"(agrupamiento).

El problema de evaluar el resultado de un algoritmo de clasificación no supervisada es importante en cualquier disciplina donde se aplique, entendiendo por ello, cualquier tipo de datos. Una de las premisas para usar métodos de clustering es ejecutar los métodos varias veces [Han J. y Kamber M. (2006)], variando la entrada de datos, entonces es de gran interés hacer un análisis de dichas variaciones con datos que poseen cierta distribución, con la finalidad de reducir costo computacional para el investigador, y reducir el tiempo de análisis del método, para concentrarlo en el verdadero

análisis, como por ejemplo análisis ambientales [Reyes H. (2008)].

La tesis se encuentra estructurada de la siguiente forma:

En el capítulo 1, se exponen los objetivos generales y específicos, la metodología a seguir y el estado del arte del tema de la tesis.

En el capítulo 2, se da el marco teórico, abordando brevemente la historia de la minería de datos, su proceso, sus tareas y sus técnicas de clustering, profundizando en los algoritmos: CobWeb, EM (Expectation Maximization) y K-medias. Asimismo se hace un breve desarrollo de los conceptos indispensables que se manejan de las áreas de probabilidad y estadística, abordando el tema de generación de números aleatorios, y análisis de varianza (ANOVA), parte importante en nuestra investigación.

En el capítulo 3, se presenta una breve revisión del sistema WEKA. Se muestra el ambiente en que funciona el sistema, su constitución, sus secciones y métodos, así como, una aproximación al proceso de minería de datos llevado a la práctica con datos simulados, con los diferentes algoritmos de clustering, mencionados anteriormente. Posteriormente, se realiza la fusión de los dos sistemas mediante RWeka. Se presenta la implementación de un programa en R, que toma en cuenta las características que se desean para realizar la comparación entre diferentes números de datos y diferentes números de variables; con las técnicas de agrupamiento antes mencionadas, mediante un ANOVA. Finalmente, se exponen las conclusiones obtenidas en este trabajo.

1.1. Objetivos Generales y Específicos del Proyecto

1.1.1. Objetivo General

Realizar un estudio comparativo, usando simulación, con el fin de analizar el comportamiento de las agrupaciones de datos obtenidos por diferentes algoritmos de clasificación para diferentes distribuciones.

1.1.2. Objetivos Específicos

- Estudiar y manejar el software WEKA, en particular, su módulo de agrupamientos
- Estudiar y manejar el software R para generar los modelos aleatorios (Distribución Normal, Distribución Uniforme y Distribución Exponencial).

- Realizar un estudio comparativo sobre los métodos de agrupamiento CobWeb, EM (Expectation Maximization) y K-medias; variando los tamaños de muestra, el número de variables y utilizando los modelos aleatorios generados.
- Usar el análisis de varianza (ANOVA) para comparar las agrupaciones.
- Dar las aportaciones y conclusiones de la investigación.

1.2. Metodología

- Hacer la revisión bibliográfica de los temas a investigar en este proyecto.
- Comprender y analizar la programación R y WEKA.
- Comprender y analizar temas en el área de la estadística, tales como: análisis de varianza, densidades utilizadas y análisis de cluster, utilizando algoritmo de k-medias, EM y CobWeb.
- Investigar y analizar el paquete estadístico "R".
- Programar un procedimiento aleatorio en R, para realizar las simulaciones de los algoritmos de clasificación usando WEKA, tomando diferentes tamaños de muestras y diferentes números de variables.
- Comparar las clasificaciones obtenidas usando procedimiento estadístico ANOVA.
- Plantear conclusiones acerca de la investigación realizada.

1.3. Estado del Arte

En la actualidad la minería de datos es utilizada como una tecnología emergente para la ayuda de toma de decisiones, mediante la abstracción de patrones importantes que se encuentran en las bases de datos. A manera de ejemplo de este manejo en el 2007, se llevó a cabo un estudio comparativo de diferentes algoritmos clasificadores disponibles en el software WEKA, en donde se seleccionó el proceso que ofrece mejores resultados para la evaluación del desarrollo de software [Dapozo G. *et al.* (2007)].

En diferentes ambientes tecnológicos cada vez hay más trabajos relacionados con estudios de comparación en algoritmos de agrupamiento. Dentro de cada problemática el investigador o interesado tiene que descubrir las distribuciones de prueba que rigen al fenómeno aleatorio inmerso en la investigación que se realiza, lo cual puede ser resuelto con el empleo de algún algoritmo de agrupamiento conveniente, como en el trabajo de [Montes N. (2001)] en el cual utiliza algoritmos de segmentación para

visualizar imágenes de frutos maduros y así lograr dar una propuesta a la mejora de recolección de esos fruto; también en el trabajo de [Maya C. (2001)] donde se desarrolla un algoritmo para caracterizar un grano de café mediante el análisis de su tamaño, calculando la media, desviación y simetría de las componentes en los espacios de color RGB y así se logra una clasificación de acuerdo al grado de madurez ó en el caso del análisis de voz, como en el estudio de [Docío L. y García C. (2005)] donde se realiza un análisis de locutores a dos voces con algoritmos de agrupamiento, para un ámbito forense y logrando realizar un sistema de segmentación de audio que es capaz de detectar y seguir a un conjunto de determinados locutores de los que se dispone información acústica a priori. De aquí el interés de estudiar las aplicaciones y creaciones de nuevos algoritmos, que ayuden a resolver diversas situaciones. Esto constituye un desafío importante en la actualidad como se puede observar en el trabajo de Pascual y otros [Pascual D. *et al.* (2007)]. En el trabajo de [Lorrio A. (2009)] consiste en manipular los métodos de agrupamiento para realizar una comparación de objetos, en este caso de vasijas con respecto a mediciones de variables discretas, utilizando los enlaces de tipo 'medio', 'completo', y el método del centroide con distintos tipos de métricas. Estos procedimientos se comparan con los métodos no jerárquicos como k-medias simple y con sistema de análisis de variables denominado Componentes Principales (C.P.) en donde se recurrió al paquete SPSS/PC (Statistical Package for the Social Sciences), el cual obtuvo una comparación significativa entre los jerárquicos y el kmedias, dando como mejor método el segundo y da una pauta hacia las C.P. para validar conglomerados pues no dependen de los resultados obtenidos ni del método usado.

Para tener mayor claridad del potencial de la tecnología de minería de datos, se describen brevemente dos casos exitosos, llevados a la practica en el ámbito comercial:

El primero es el sistema de Minería de Datos de la NBA "Advanced Scout", este sistema fue desarrollado por la IBM para la Asociación Nacional de Baloncesto Norteamericano (NBA), según Tom Sterner se toma tiempo para hacer un análisis profundo y complejo, pero con el programa mencionado ya que soló tomaba unos cuantos minutos y el tiempo restante se utiliza para evaluar la toma de decisiones [Reese S. (1996)].

El segundo es el sistema CRIS que es una red neuronal, éste aprende a reconocer patrones de gastos de los titulares de las tarjetas, y de las transacciones de las cuentas. En el año 2000 se estimo 5250 tran/seg., estos sistemas se mencionan en [Medina J. (2000)].

Sin embargo, todavía es una tecnología en pañales, y es así que Microsoft se ha interesado en la ciencia computacional para el análisis de datos [Piatestsky y Shapiro G. (2006)]. Actualmente, hay otras empresas que se han interesado en la comercialización de metodologías y programas para sistematizar problemas de minería de datos. En estos ambientes hay algunas metodologías disponibles como: CRISP-DM que es usado principalmente en problemas de negocios y marketing. Esta metodología es de

uso libre aunque actualmente es el soporte para el software Clementine SPSS Data Mining [Jackson J. (2002)]. Otros paquetes usados son: S-plus Insightful Miner y Oracle Data Mining. También WEKA de sus siglas en inglés *Waikato Environment for Knowledge Analysis*, es un software libre muy poderoso, al igual que los antes mencionados y el lenguaje de programación estadístico R que es muy compatible con S-plus.

Además, la minería de datos ha demostrado ser una valiosa metodología para descubrir comportamientos que se generan con el tiempo; como en grupos, la identificación se realiza mediante características del comportamiento [Delgadillo G. *et al.* (2006)]. Existen varios factores para que la minería de datos funcione: se necesita una gran cantidad de datos, un experto en el tema sobre el cual se va a desarrollar la minería de datos, necesita una metodología y varios métodos de validación de resultados. Con esto, se logran tomar decisiones y saber su efecto, no será una condición de incertidumbre pues se sabe de antemano qué efecto tendrá una decisión. Esto ocasiona conocimiento generado al analizar sus distintas situaciones.

Asimismo, es necesario examinar otras metodologías relacionadas con la comparación de estos métodos; uno de ellos muy importante es el estudio de simulación, es el trabajo donde se compara el método Ward's, complete linkage, average linkage, etc., [Von A. y Mair P. (2008)]. Estos algoritmos son jerárquicos [Hair J. *et al.* (1999)] y tratan de encontrar una estructura para los datos generados después de haberlos transformado.

Así, como el estudio fuertemente relacionado es la comparación de los algoritmos de clasificación automática se ha planteado en el "Estudio Comparativo De Métodos De Clasificación Automática En La Zonificación Agro Ecológica Del Sur Del Estado De Puebla". Este estudio busca comparar las agrupaciones mediante datos reales, es un trabajo importante ya que se puede tomar como un manual para WEKA y una guía para los métodos de agrupación [Vásquez B. (2009)].

Para terminar este estado del arte, se habla de los dos primeros ambientes disponibles para el aprendizaje automático y análisis estadístico: WEKA y R con licencia (GNU), estos ambientes además, cuentan con la ventaja de ser lenguajes de programación, asimismo han surgido de sus comunidades estadísticas y de aprendizaje automático, siempre están innovándose continuamente, y de estos surge **RWeka** como una herramienta que realiza la unificación de ambos entornos, que tomando lo mejor de cada uno, unifica sus procesos. Los autores han escrito a RWeka como un package o librería [Schauerhuber M. *et al.* (2007);Hornik Kurt *et al.* (2008)], con la cual se concibe la funcionalidad de la interfaz WEKA en R, por tanto, se reutiliza las clases de aprendizaje como son: Clasificadores, agrupaciones, asociaciones, filtros, cargadores, etc.

Como se puede observar en el documento titulado R Meet Weka, creado por los mismos autores del paquete. RWeka a su vez necesita de otra librería denominada **rjava** [Urbanek S. (2007)] la cual se encarga de gestionar la maquina virtual de java (se utiliza para este estudio), pero hay otras librerías que tienen la misma función como son **SJava** [Lang D. y Chambers J. (2005)] o **arji** [Carey V. (2007)], para los distintos sistemas operativos.

Capítulo 2

MARCO TEÓRICO

En este capítulo se da a conocer la tecnología llamada minería de datos, así como su utilidad para encontrar relaciones y patrones ocultos explorando bases de datos. De modo específico, se da una breve introducción sobre esta tecnología, enseguida se describe a detalle sobre su proceso, sus tareas y sus técnicas, para lo cual se profundiza sobre el tema de algoritmos de agrupamiento. Posteriormente, se continúa con el análisis de la varianza también llamada ANOVA, que nos servirá para hacer las comparaciones de los diferentes resultados de los algoritmos utilizados, después se sigue con una breve descripción de números aleatorios y finalmente, se dan las propiedades matemáticas de las distintas distribuciones de probabilidad que se utilizan en el estudio.

2.1. Minería de Datos

Esta tecnología no es reciente, ya que en los años 70 había términos como: arqueología de los datos, cosecha de la información, descubrimiento de la información, en donde principalmente los investigadores estadísticos utilizaban esos términos. Fue a mediados de los 90's cuando se impartió la primera conferencia que incluyó el término minería de datos que se denominó "Internacional Conferences on Knowledge Discovery in Database and Data Mining"[Christen P. (2005)].

Una definición muy aceptada en los textos de Minería de Datos está dada como un proceso no trivial de identificación válida, novedosa, potencialmente útil y entendible de patrones comprensibles que se encuentran ocultos en los datos [Fayyad U. *et al.* (1996)]. También puede decirse que la minería de datos consiste en el uso de algoritmos que generan patrones y cabe mencionar que es parte de un proceso mayor denominado KDD (Knowledge Discovery from Database). Es importante remarcar que la minería

de datos y el KDD realizan: selección de datos, hacen preprocesos y transforman a los datos para su exploración ó descubrimiento de la información. Además, la minería de datos es una tecnología que está compuesta de análisis estadístico, Inteligencia Artificial y Visualización. Por consiguiente, la Minería de Datos tiene un enfoque exploratorio y no confirmatorio, es decir, que no hay una hipótesis previa, ni ideas predeterminadas sobre un resultado interesante, todo se va a descubrir.

2.1.1. Proceso de Minería de Datos

En los textos revisados se encuentra que no hay un estándar para decir cómo hacer minería de datos [Bramer M. (2007); Giudici P. (2003); Zhengxin C. (2001); Kantardzic M. (2003)].

Nota: Se debe tomar en cuenta que todo el proceso minero es iterativo, esto quiere decir, que en cualquier paso se puede retroceder. Por consiguiente un resultado obtenido en el paso de prueba se puede alimentar de nuevo de cualquiera de los pasos anteriores. Por ejemplo, el resultado puede indicar que es necesario redefinir el problema, para recoger más datos, o quizá realizar de una forma diferente el pre procesamiento de datos, o para la selección de un algoritmo diferente o diferentes parámetros en el mismo algoritmo. El ciclo continuará hasta que sea satisfactorio el resultado que ha sido obtenido. Se muestra a continuación en la Figura 2.1, el ciclo de vida de minería de datos [Zhengxin C. (2001)].

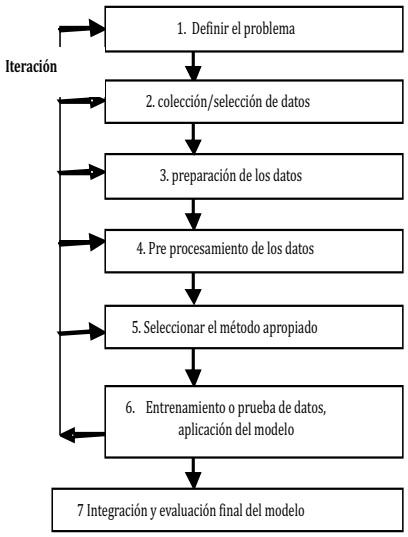

Figura 2.1: El ciclo de vida de minería de datos [Zhengxin C. (2001)]

A continuación se presenta una estrategia de organización para realizar minería de datos [Zhengxin C. (2001); Kantardzic M. (2003)], con respecto a la Figura 2.1.

Definir el Problema

Esta es una de las partes más difíciles de realizar, ya que se necesita de la colaboración del analista y el usuario. Lo cual implica, combinar el dominio de la aplicación con un modelo de minería de datos. Para analizar el problema, se necesita identificar los objetivos y definir las metas, para determinar el apropiado uso de la minería de datos. En términos computacionales es definir la forma de entrada y salida de los datos. Esta cooperación no se detiene en la fase inicial sino que continúa durante todo el proceso.

Colección y Selección de datos

Con base al objetivo del problema, en ocasiones se necesita seleccionar datos de diferentes tipos de información. Tomando fuentes de datos múltiples que se pueden combinar para crear colecciones de información para ser aplicado a un proceso de minería de datos. La colección de datos se procesa y se evalúa con respecto a su homogeneidad y variabilidad, para realizar el descubrimiento.

Preparación de los datos

La preparación de datos juega un papel muy importante en la minería de datos. Ya que en la mayoría de los casos, los datos en bruto no se pueden minar y la tarea es construir representaciones de datos para minar. En la investigación realizada para este trabajo, la mayoría de los profesionales de minería de datos concuerdan que el 50-80 % del total de ciclo de vida de un proyecto de minería de datos, puede ser asumido por la etapa de preparación de datos. El objetivo de esta etapa es para limpiar los datos y transformarlos en un formato adecuado para la aplicación de técnicas de descubrimiento.
Se presentan algunos ejemplos de preguntas, que ayudan en el razonamiento de la preparación de los datos:

¿Qué condición tienen los datos (discretos, continuos o tiene una distribución especifica)?
¿Qué medidas son necesarias para preparar los datos para el análisis?
¿Qué conversiones son necesarias antes del análisis?
¿Son estos procesos aceptables para los usuarios y la presentación del resultado?
¿Existe la necesidad de normalizar los datos?

Hay que observar que generalmente, la preparación de datos implica la extracción de los datos y su transformación al formato requerido para los algoritmos de minería de datos específicos. Esto incluye la limpieza de datos, la agregación de datos, la búsqueda de nuevos atributos y la normalización de los mismos, entre otras medidas

importantes. En el contexto de la preparación de datos, el término normalización se refiere a una estandarización de los datos de entrada, es decir, que los datos estén en la misma escala [Zhengxin C. (2001)].

Pre procesamiento de los datos

En el pre procesamiento de los datos se debe detectar los valores inusuales, es decir, son los datos que no son compatibles con la mayoría de las observaciones; que comúnmente, son de resultados anormales, errores de codificación y a veces, son naturales los valores inusuales. Estos datos pueden afectar seriamente al modelo producido, por lo cual, se deben detectar y eliminar.

Es importante marcar que se deben hacer modelos robustos insensibles al ruido ó al menos explicar por qué surge el ruido y decidir qué hacer sobre las estrategias para datos faltantes.

También en muchos casos hay que hacer tratamientos para estandarizar y codificar los datos, como en el paso anterior. Por ejemplo, una función con el intervalo [0, 1], y otra sobre [-100, 1000] no tendrán el mismo peso en la técnica aplicada, en consecuencia esto influye en el resultado final de minería de datos. Además, la aplicación específica de los métodos de codificación suelen lograr la reducción de dimensión, proporcionando un número menor de elementos de información.

Seleccionar un método apropiado de minería

Se debe seleccionar la tarea a realizar, que lleva a la apropiada selección de un algoritmo, para decidir en qué modelo y que parámetros son adecuados a los objetivos. Además, cabe mencionar que cada tarea lleva a un pre proceso diferente en los datos. Y que al aplicar los métodos de minería se encuentran patrones interesantes, que hacen una representación específica; estas representaciones son conocidas, como: Técnicas de agrupamiento, reglas de clasificación, árboles de decisión, etc.

Entrenamiento o prueba de datos, aplicación del modelo

En esta etapa de entrenamiento o prueba de datos, primeramente se práctica con varios modelos, como: validación cruzada, segmentación de datos, simulación, etcétera. Si durante el proceso no se cumplen los resultados esperados, se debe alterar alguno de los pasos anteriores para generar un nuevo modelo. Es importante mencionar que en

cualquier etapa del proceso, se puede retroceder para poder tener un modelo exitoso como se muestra en la figura 2.1.

Integración y evaluación final del modelo generado

Los resultados de los modelos deben ayudar a tomar decisiones o deben permitir adquirir nuevo conocimiento de los datos. Estos modelos deben ser interpretados para ser útiles, ya que los seres humanos no pueden basar sus decisiones en modelos complejos. Normalmente, los modelos simples son más interpretables, pero también son menos exactos. Un usuario no quiere solamente páginas llenas de números, él necesita comprender, sintetizar, interpretar y utilizar conocimientos abstractos para generar un modelo.
Así que, la visualización y las técnicas de representación son importantes para la presentación del conocimiento extraído. A continuación, se debe proceder a realizar la validación, para comprobar que el conocimiento arrojado es válido y suficientemente satisfactorio.

Una buena comprensión de todo el proceso es importante para el éxito de cualquier aplicación. No importa qué tan poderoso sea el método utilizado. El modelo resultante no será válido si los datos no son recogidos y pre procesados correctamente, o si la definición del problema no es relevante.

Dependiendo de la técnica de aplicación, se pueden reducir estos pasos. Por ejemplo, en el caso de los cluster, generalmente se reducen a cuatro pasos, esto se verá en el capítulo 3 [Von A. y Mair P. (2008)].

2.1.2. Tareas de la Minería de Datos

Dependiendo principalmente de la aplicación especifica y en el interés del Investigador, se pueden identificar ciertos tipos de tareas de minería de datos. Las clases o categorías de minería de datos utilizados para la descripción y/o predicción son las siguientes:

Clasificación Frecuentemente queremos clasificar datos de acuerdo a un valor de la función objetivo. Por lo cual, se dividen los datos en tres conjuntos ajenos: uno de ellos es entrenamiento, otro de validación y un último es de la prueba [Carmona L. (2006)]. El objetivo de la clasificación es analizar los datos en el entrenamiento y desarrollar una descripción precisa o bien identificar un modelo probabilístico para cada clase usando las características disponibles de los datos. La función objetivo puede ser categórica o nominal.

Regresión Esta tarea es conceptualmente similar a la clasificación. Su función ob-

jetivo es de tipo continua o binaria y responde a datos que generalmente están en el tiempo.

Agrupamiento El objetivo es clasificar una muestra de entidades (personas u objetos) en un número pequeño de grupos mutuamente excluyentes basados en similitudes entre las entidades. Por tanto, se usa esta técnica para identificar alguna taxonomía o estructuras con sentido.

Sumarización o Resumen Proporcionan una descripción compacta de un conjunto ó subconjunto de datos, mediante algunas estadísticas.

Dependencia de Modelos La búsqueda de un modelo que describe las variables importantes y sus relaciones significativas entre los grados de dependencias o entre los valores de una función en un conjunto o sub conjunto de datos.

Cambio y detección de desviación Descubrir los cambios más significativos en el conjunto de datos.

Estas son las tareas principales que describe [Kantardzic M. (2003)]. Pero hay otros autores que tienen como tareas principales de minería la asociación, la generación de reglas y árboles de inducción, otros también incluyen análisis de secuencias [Hand D. *et al.* (2001); Zhengxin C. (2001); versión 3.5.8 (2008)]. Todas son tareas importantes, pero no relevantes para el trabajo.
Aquí nos centraremos en técnicas de agrupamiento, las cuales se dividen en dos grandes categorías:

* Predictiva - como su nombre lo dice trata de predecir valores desconocidos en otros campos o variables que son de interés.
* Descriptiva - está enfocada a encontrar relaciones, patrones basados en un conjunto de datos validos.

En algunas ocasiones debido a la flexibilidad del problema se toman estas categorías de manera contraria, por ejemplo: Redes Neuronales, árboles de decisión, K-NN y k-Medias, estas tareas a su vez están orientadas al clustering [Hernández J. (2003)].En el apartado siguiente se toca el tema de aprendizaje supervisado y no supervisado.

2.1.3. Técnicas de Minería de Datos

Es importante saber que hay dos tipos de aprendizaje: uno supervisado y no supervisado; el primer término tiene una función de bondad que clasifica los datos según sea su criterio en el cual se logra predecir la respuesta con nuevos datos, por ejemplo: el método de árbol J48 equivalente a C4.5, el Naive Bayes con métodos bayesianos, o

con lógica difusa como KStar. El no supervisado solo observa los datos, no hay una función de bondad para guiar el aprendizaje y su único conocimiento será inducido a partir de un conjunto de muestras, como a continuación se presentan las técnicas relevantes para este estudio de (clustering) K-medias, EM y CobWeb.

Clustering

En este procedimiento, la tarea es dividir los datos en grupos de objetos parecidos, es decir, simplemente se descubren las estructuras de datos sin explicar porqué existen. Para hacer la discriminación se utilizan diferentes métricas de distancia, y se utilizan medidas de similitud entre datos u objetos, como son: la distancia Euclidiana, la Manhatan, la Mahalanobis, entre otras. El clustering es principalmente una técnica de aprendizaje automático y juega un papel importante en minería de datos, como es la exploración de datos científicos, recuperación de información, aplicaciones web, marketing, análisis de ADN en biología computacional, etc.
Estas técnicas a su vez se dividen en métodos jerárquicos y no jerárquicos. También estos procedimientos se dividen en paramétricos y no paramétricos [Bishop M. (2006)].

Jerárquicos Este procedimiento puede ser Aglomerativo o Individual, el primero se empieza suponiendo que toda la información se encuentra en grupos diferentes (tantos grupos como objetos o variables se tengan) y después se van uniendo con base a un criterio de clasificación dado. El segundo procedimiento, parte de que todos los objetos o variables que pertenecen a un mismo grupo y después se van dividiendo e igual que el anterior, se toma un criterio de clasificación [Marín J. (2008)]. A continuación se muestra una gráfica en la figura 2.2, un ejemplo de un agrupamiento jerárquico, con datos tomados de un repositorio de R, sobre las distancias de ciudades en USA.

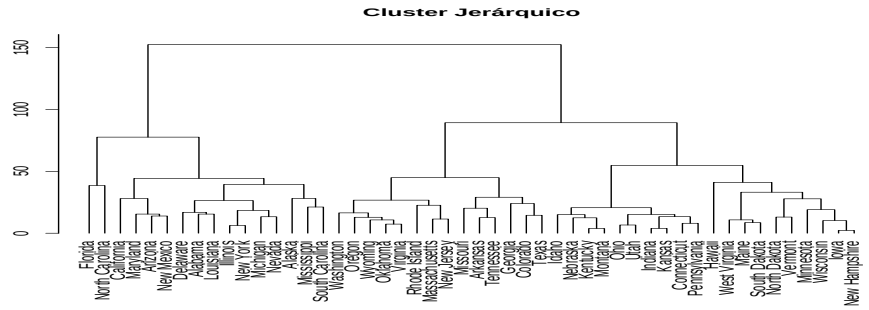

Figura 2.2: Dendograma de ejemplo con datos de distancia sobre los estados de estados unidos

CobWeb Este procedimiento es un método jerárquico, ya que va formando un árbol de clasificación. Las conexiones constan de nodos, que representan un concepto de descripción probabilístico, pues cada nodo está formada con probabilidades condicionales del atributo - valor ($P(A_i = V_i|C_k)$). Se utiliza una medida a la que nombran "utilidad de categoría", que se define a continuación

$$CU = \frac{\sum_{k=1}^{n} P(c_k)[\sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2]}{n}$$

Este método hace una búsqueda en forma descendente dirigiéndose hacia encontrar el mejor lugar (nodo) para cada objeto, es decir, el objeto se va recorriendo en cada nodo y se compara con el nodo en turno, para ver que objeto da mayor ganancia de utilidad de categoría. Este proceso en cada iteración pone a consideración el unir los dos mejores nodos conforme a tener una mayor utilidad y viceversa. Si no resulta beneficiosa la unión de nodos, se considera dividir el nodo de tal manera que resulte mejor para el objeto. Algo importante de este método, es que el orden de los objetos si importa, así que es bueno probar con diferentes posiciones y además, de asumir que la probabilidad de los atributos es independiente de las demás. El algoritmo COBWEB no necesita de un número exacto de cluster, ya que automáticamente encuentra el óptimo [Vásquez B. (2009)].
A continuación se presenta el algoritmo:

Algoritmo CobWeb

1. Actualizar cuenta de Raíz
2. Si Root es una hoja

   Entonces retorna la hoja expandida para dar origen al nuevo objeto, Si no encuentra los hijos de raíz, se coloca al objeto
   Según sea el caso:

   a) Verificar la necesidad de crear una clase nueva,
   b) Verificar la necesidad de fusionar el nodo,
   c) Verificar la necesidad de dividir el nodo,
   d) Si ninguna de las anteriores se realiza (a, b, o c)

   THEN llamar a COBWEB (Objeto, Mejor hijo de raíz)
3. Fin

Además COBWEB pertenece a los métodos de aprendizaje conceptual ó basado en modelos. Este procedimiento tiene como interés el reconocer y asociar características comunes a un grupo de objetos [Martínez J. (2001)]. Esto significa que cada cluster se considera como un modelo que puede describirse intrínsecamente, más que un grupo formado por una colección de puntos [Morales E. (2005)]. Se muestra a continuación

un gráfico del algoritmo CobWeb en la figura 2.3, con datos simulados de una Normal, con diferentes medias.

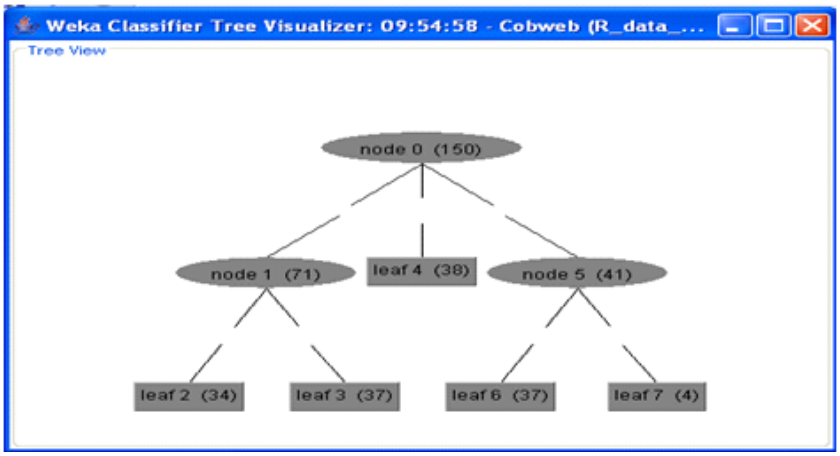

Figura 2.3: Salida del programa WEKA, del método cobweb

Aglomerativo Este procedimiento depende de la distancia qué se tome, algunas métricas que se usan frecuentemente son: La distancia mínima (single linkage), la distancia máxima (complete linkage), la distancia entre centros (centroid), la distancia mediana (median) y la distancia promedio [Marín J. (2008)].
A continuación se presenta el algoritmo de agrupamiento aglomerativo, que puede tomar algunas de las métricas antes mencionadas.

Algoritmo Aglomerativo

1. Empezar con N cluster (N = al número inicial de elementos) y una matriz $N \times N$ simétrica de distancias o similitudes. $D = [d_{ij}]_{ij}$, donde $i = \overrightarrow{(1, N)}$; corresponde a las filas de la matriz y $j = \overrightarrow{(1, N)}$; son las columnas.
2. Dentro de la matriz de distancias, se buscan los cluster U y V, tales que cumplan con la métrica especificada, para d_{uv}.
3. Juntar los cluster U y V en uno solo. Actualizar la matriz de distancias:
   (i) Borrar las filas y columnas de los cluster U y V.
   (ii) Formar la fila y columna de las distancias con respecto al nuevo cluster (UV).
4. Repetir los pasos (2) y (3) un total de $(N - 1)$ veces.

Se presenta un ejemplo en el Anexo A.

No jerárquicos Los siguientes algoritmos son importantes para el estudio debido a la forma en que se trabaja con los grupos. En este estudio comparativo la diferencia entre cada uno de los métodos es importante, porque si se toma el algoritmo EM, este tiene sus bases en un método estadístico denominado estimación de máxima verosimilitud. En cambio el algoritmo K-medias trabaja con centroides y se debe tener en cuenta cuantos grupos hay que generar. Otra diferencia consiste en que el método EM estima por sí mismo los K grupos (un número óptimo de grupos) y por último, el método COBWEB que se identifica por tratar de ajustarse al mejor modelo.

Algoritmo EM (Paramétrico) Lo que se busca en este método es agrupar los cluster más probables dados los datos, entonces los objetos (datos) tienen una probabilidad de pertenecer a un grupo. El principio de este método está basada en el modelo estadístico denominado *"Finite Mixture Models"* (Modelo de mezclas finitas), como se muestra en la figura 2.4. Una mezcla es un conjunto de "K"distribuciones que representan "K"grupos, cada distribución nos da la probabilidad de que un objeto tenga en particular una asociación de "Atributo-Valor"[Rui X. y Donald W. II (2005)].

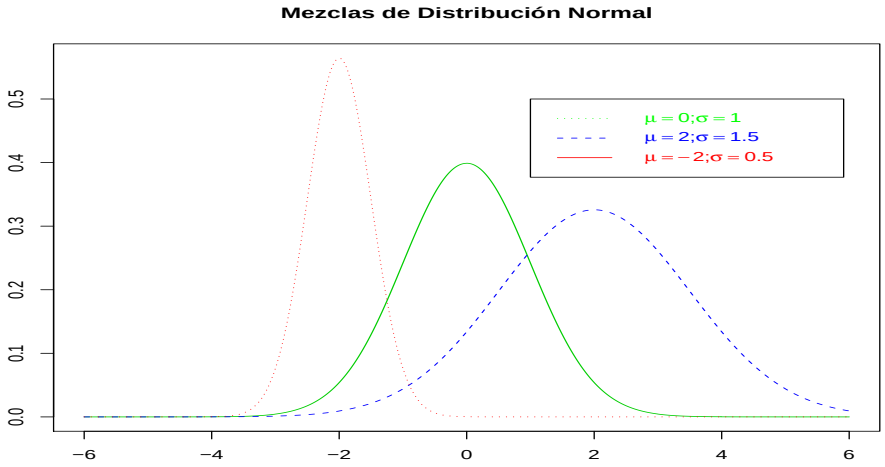

Figura 2.4: Se muestra una gráfica con 3 distribuciones Normales, ejemplificando las mezclas, salida con el paquete R.

El problema radica, que no se sabe de qué distribución tienen los datos y tampoco se conocen los parámetros de dichas distribuciones. Entonces el algoritmo EM (Expectation Maximization) aborda el problema estimando los parámetros de forma aleatoria y estos a su vez los usa para calcular las probabilidades para volver a estimar los parámetros de las probabilidades, estos dos pasos se repiten hasta converger, (se puede hacer al revés, en donde primero se adivinan las probabilidades y después se calculan los parámetros).

Entonces el algoritmo EM, procede en dos pasos y estos a su vez se repiten iterativamente, a continuación se describe de la siguiente manera.

Algoritmo EM

1. **Paso E (Expectation)** Utiliza los valores de los parámetros, en primera instancia se utilizan los parámetros iniciales y posteriormente los proporcionados por el paso "Maximization"de la iteración anterior y así se van obteniendo las probabilidades de los grupos, que son los valores esperados de los grupos.

2. **Paso M (Maximization)** Se obtienen los valores de los parámetros de las distribuciones, este paso se maximiza la verosimilitud de las distribuciones dados los datos.

Pero, para estimar los parámetros, se considera que únicamente contamos con las probabilidades de pertenecer a cada cluster y así las probabilidades actúan como pesos, como se describe a continuación:

$$\mu_A = \frac{w_1x_1 + w_2x_2 + \ldots + w_nx_n}{w_1 + w_2 + \ldots + w_n}$$

$$\sigma_A^2 = \frac{w_1(x_1 - \mu)^2 + w_2(x_2 - \mu)^2 + \ldots + w_n(x_n - \mu)^2}{w_1 + w_2 + \ldots + w_n}$$

donde w_i es la probabilidad de que el objeto i pertenezca, por ejemplo, al cluster A y se suma sobre todos los objetos.
Podemos destacar que este método converge pero casi nunca llega a un punto fijo, se puede saber que tanto se acerca, calculando la verosimilitud de los objetos individuales (i = datos):

$$\prod_i^N (P_A P(x_i|A) + P_B P(x_i|B) + \ldots)$$

donde N es el número de instancias, que suponemos independientes entre sí. Esta medida crece en cada iteración, por tanto se itera hasta que el crecimiento sea despreciable.

k-means clustering (no paramétrico) Este es un algoritmo para encontrar los K grupos en términos de una distancia (media) usando un conjunto arbitrario de datos [Marín J. (2008); Morales E. (2005); John W. y John W. Jr. (2007); Hartigan J. y Wong M. (1979)]. Este método encuentra agrupamientos circulares, a continuación se presenta el algoritmo.

Algoritmo K-means o "K-medias"

1. Dividir aleatoriamente los datos en K conjuntos y calcular la media de cada conjunto.
2. Reasignar cada dato al conjunto con el punto medio más cercano.
3. Calcular los puntos medios de los k conjuntos de datos.
4. Repetir los pasos 2 y 3 hasta que los conjuntos no varíen.

Es importante remarcar, que regularmente se utiliza la medida de similaridad basada en el error cuadrático $E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$, donde p representa al objeto y m_i a la media del cluster C_i. Este método es susceptible a tomar valores extremos ya que se distorsiona la distribución de los datos. Se pueden utilizar otras medidas aparte del error cuadrático como son, las modas, las medias, y mediana, etc. Se agrupa con respecto al objeto más representativo del cluster, se busca encontrar un objeto representativo, cambiando la mediana en forma aleatoria y se mide si se mejora la calidad de los cluster [Morales E. (2005)]. A continuación en la figura2.5, se muestra el resultado de una corrida de "K-medias" de WEKA con datos simulados, pero en R.

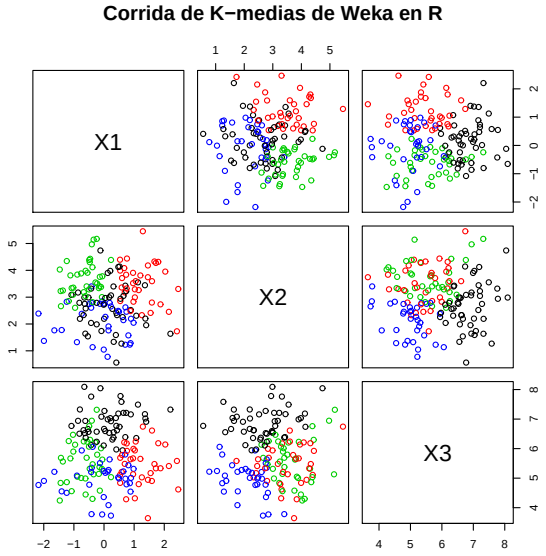

Figura 2.5: Gráficos de Kmedias de WEKA con el paquete R.

En el siguiente apartado se introducen los conceptos probabilísticos y estadísticos que ayudan a manejar el ANOVA, que es una herramienta de decisión de gran importancia para el presente trabajo.

2.2. Análisis de la Varianza (ANOVA)

El análisis de la varianza es un método estadístico, introducido por Fisher para poder evaluar los efectos de los distintos niveles de un factor sobre una variable continua [Arriaza A. *et al.* (2008)]. El ANOVA es usado para realizar el contraste de igualdad de medias en dos o hasta "K"poblaciones normales; donde se plantea la hipótesis $H_0 : \mu_1 = \mu_2 = \ldots = \mu_k$. vs H_a :*no todas las medias son iguales*, fijando un nivel de significancia α. Si existe una población con medias diferentes se rechazara H_0 a un nivel de significancia establecido.

2.2.1. Premisas para el Análisis de la Varianza

* Las poblaciones tienen distribuciones que son aproximadamente normales. Este requisito no es demasiado estricto, ya que este método funciona bien, a menos que la población tenga una distribución muy diferente a la normal [Mendenhall D. y Scheaffer W. (1986)].

* Las poblaciones tienen la misma varianza σ^2(o desviación estándar σ). En la literatura se menciona que este criterio no es tan estricto pues funciona si los tamaños de las muestras son casi iguales, de tal forma que la más grande sea nueve veces el tamaño de la más pequeña, por lo cual, los resultados del ANOVA siguen siendo confiables [Triola M. (1999)].

* Las muestras deben ser aleatorias e independientes e idénticamente distribuidas, es decir,

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i).$$

2.2.2. Análisis de Varianza Simple

En este apartado, se explica de forma explícita el razonamiento de este proceso. Se debe suponer que tenemos una variable aleatoria (formada de mediciones), a la que se le denomina Y_i, se le asocia un error ϵ_i; entonces cada variable aleatoria Y_i puede representarse como la suma de dos cantidades: la cantidad que se desea medir, a la cual se le identifica con la letra μ y el error de medición correspondiente. Se admite la siguiente representación formal:

$$Y_i = \mu + \epsilon_i \quad ; i = 1, 2, \ldots.$$

En la ecuación anterior,ϵ_i puede tomar tanto valores positivos como negativos, de lo contrario se presupone que se toman sólo valores negativos (o positivos),además, dado que los ϵ_i son variables aleatorias, ya que las Y_i lo son, mientras que μ es una contante.
Entonces suponiendo que $E(\epsilon_i) = 0 \quad y \quad Var(\epsilon_i) = \sigma^2$ se tiene:

$$E(Y_i) = E(\mu + \epsilon_i) = \mu + E(\epsilon_i) = \mu \quad ; \quad Var(Y_i) = Var(\mu + \epsilon_i) = Var(\epsilon_i) = \sigma^2.$$

Entonces se llega a que $Y_i \sim N(\mu, \sigma^2) \quad ; \quad i = 1, 2, \ldots, n$; con n el número de variables.

2.2.3. Análisis de Varianza Diseño Completamente Aleatorizado

En un análisis de la varianza en un diseño completamente aleatorizado, primero se debe ver la forma que tienen los datos, como se muestra en la tabla de muestras, en donde Y_{ij} es observación i-ésima de la población j-ésima, K es el número de poblaciones, $Y_{(i.)}$ es el total de observaciones de la población i.

Tabla de muestras

	1	2	3	...	K	
	Y_{11}	Y_{12}	Y_{13}	...	Y_{1k}	
	Y_{21}	Y_{22}	Y_{23}	...	Y_{2k}	$1 \leq j \leq k; 1 \leq i \leq n_i.$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
	$Y_{n_{1k}}$	$Y_{n_{2k}}$	$Y_{n_{3k}}$	...	$Y_{n_{kk}}$	
Totales de las muestras	$Y_{1.}$	$Y_{2.}$	$Y_{3.}$	...	$Y_{k.}$	$Y_{..}$
Medias de las muestra	$\overline{Y}_{1.}$	$\overline{Y}_{2.}$	$\overline{Y}_{3.}$	...	$\overline{Y}_{k.}$	

El total de todas las muestras del tipo i, se representa:

$$Y_{i.} = \sum_{j=1}^{n_i} Y_{ij} \quad ; \quad i = 1, 2, \ldots, k;$$

Por $Y_{..}$ se representa la suma total de todas las muestras y su representación formal es:

$$Y_{..} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} Y_{ij},$$

las medias se representan de la forma:

$$\overline{Y}_{i.} = \frac{Y_{i.}}{n_i} = \sum_{j=1}^{n_i} \frac{Y_{ij}}{n_i} \quad ; \quad i = 1, 2, \ldots, k$$

y finalmente la media general es:

$$\overline{Y}_{\cdot\cdot} = \frac{Y_{\cdot\cdot}}{\sum_{i=1}^{k} n_i}.$$

Es fácil ver que la notación adoptada consiste en sustituir un índice por un punto una vez que sea sumado sobre ese índice. Ahora se formula la siguiente expresión:

$$Y_{ij} = \mu_i + \epsilon_{ij} \quad ; \quad j = 1, 2, \ldots, n_i; i = 1, 2, \ldots, k.$$

En este caso el ϵ_{ij} se integra por las diferencias obtenidas en las muestras. Se sigue que al aplicar la muestra i-ésima a un grupo (de tamaño n_i) se introduce un efecto (τ_i) de esa muestra en la variable de observación.
Entonces puede escribirse:

$$Y_{ij} = \mu + \tau_i + \epsilon_{ij} \quad ; \quad j = 1, 2, \ldots, n_i; i = 1, 2, \ldots, k.$$

Una suposición clave es la homogeneidad de las muestras, debido a un diseño completamente aleatorizado, donde:
μ : es la media general,común a todas las unidades antes de aplicar los tratamientos.
τ_i : es el efecto del i-ésima muestra.
ϵ_{ij} : es el error experimental en la j-ésima repetición de la i-ésima muestra.
Dada la identidad y su equivalente expresión, se tiene que:

$$Y_{ij} - \overline{Y}_{\cdot\cdot} = (Y_{ij} - \overline{Y}_{i\cdot}) + (\overline{Y}_{i\cdot} - \overline{Y}_{\cdot\cdot}). \qquad (*)$$

donde la desviación total de una observación con respecto a la media general se descompone en dos fuentes de variación. La primera parte $Y_{ij} - \overline{Y}_{i\cdot}$ se explica de la siguiente manera, para una muestra dada (la i-ésima), la variabilidad se debe exclusivamente de diferentes repeticiones de la misma muestra, por tanto es una desviación debida al error experimental.

En la segunda parte de la variación $\overline{Y}_{i\cdot} - \overline{Y}_{\cdot\cdot}$, es una desviación entre la media de la i-ésima muestra y la media general del experimento, si todas las muestras tienen el mismo efecto, las medias $\overline{Y}_{i\cdot}$ son iguales. Debido a esta razón $\overline{Y}_{i\cdot} - \overline{Y}_{\cdot\cdot}$se le llama una desviación al tratamiento.

Dado que la igualdad $(*)$ se cumple para todas las respuestas Y_{ij}, se puede escribir la siguiente relación:

$$\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij} - \overline{Y}_{\cdot\cdot})^2 = \sum_{i=1}^{k}\sum_{j=1}^{n_i}[(Y_{ij} - \overline{Y}_{i\cdot}) + (\overline{Y}_{i\cdot} - \overline{Y}_{\cdot\cdot})]^2$$

Distribuyendo las sumatorias y desarrollando el binomio, se obtiene:

$$\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{..})^2=\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{i.})^2+\sum_{i=1}^{k}\sum_{j=1}^{n_i}(\overline{Y}_{i.}-\overline{Y}_{..})^2 + 2\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{i.})(\overline{Y}_{i.}-\overline{Y}_{..}).$$

El doble producto de la ecuación anterior es cero. Por lo que se obtiene la partición de la suma de cuadrados totales:

$$\begin{array}{rcll}\displaystyle\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{..})^2 & = & \displaystyle\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{i.})^2 & + \quad \displaystyle\sum_{i=1}^{k}\sum_{j=1}^{n_i}(\overline{Y}_{i.}-\overline{Y}_{..})^2. \\ S.C.Total & = & S.C.Error & + \quad S.C.Tratamientos\end{array}$$

Como se observa, el segundo término no depende del índice "j" por lo que se reduce a:

$$\begin{array}{rcll}\displaystyle\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{..})^2 & = & \displaystyle\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{i.})^2 & + \quad \displaystyle\sum_{i=1}^{k} n_i(\overline{Y}_{i.}-\overline{Y}_{..})^2. \\ S.C.Total & = & S.C.Error & + \quad S.C.Tratamientos\end{array}$$

Usando las propiedades ya conocidas de la Teoría Distribucional [Mendenhall D. y Scheaffer W. (1986)], se asientan los siguientes resultados:

$$\frac{S.C.Error}{\sigma^2}=\frac{\displaystyle\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{i.})^2}{\sigma^2}\sim\chi^2\left(\sum_{i=1}^{k}n_i-k\right);$$

$$\frac{S.C.Tratamientos}{\sigma^2}=\frac{\displaystyle\sum_{i=1}^{k}n_i(\overline{Y}_{i.}-\overline{Y}_{..})^2}{\sigma^2}\sim\chi^2(k-1),$$

donde, por la independencia entre la suma de cuadrados consideradas previamente, se logra:

$$\frac{S.C.Total}{\sigma^2}=\frac{\displaystyle\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{..})^2}{\sigma^2}\sim\chi^2\left(\sum_{i=1}^{k}n_i-1\right).$$

La estadística para la prueba es F_0 que, de acuerdo con resultados anteriores, tiene una distribución de $F^{k-1}_{\sum n_i - k}$ cuando la hipótesis nula es cierta, así que la regla de asociación es: "Rechazar H_0 si $F_0 \geq F^{k-1}_{\sum n_i - k, \alpha}$ ". Consecuentemente, la región de rechazo para la prueba se localiza en la cola derecha de la distribución F.

Para finalizar esta pequeña reseña del método de ANOVA se da un resumen en la siguiente tabla.

Fuente de validación	Suma de Cuadrados	Grados de libertad	Estimadores	F_0
$S.C.Trata.$	$\sum_{i=1}^{k} n_i(\overline{Y}_i - \overline{Y}_{..})^2$	$k-1$	$C.M.T = \frac{SCT}{k-1}$	$F = \frac{C.M.T}{C.M.Error}$
$S.C.Error$	$\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij} - \overline{Y}_{i.})^2$	$k(n-1)$	$C.M.E = \frac{SCE}{k(n-1)}$	
$S.C.Total$	$\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij} - \overline{Y}_{..})^2$	$n-1$		

Tabla 2.1: Resumen de los cálculos para la ANOVA.

Para lograr la Tabla 2.1, se revisaron varias bibliografías como: [Mendenhall D. y Scheaffer W. (1986);Infante S. y Zárate G. (1990)].

Es claro que la Suma de Cuadrados ($S.C.$) de tratamiento depende la variabilidad de las medias de tratamiento. En cambio S.C. Error, se tiene que depende de la variabilidad de las repeticiones de una misma muestra. Por esta razón, en algunos textos [Triola M. (1999); Weinner R. (2001)], a la fuente de variación a la que se le denomino tratamiento se nombra **entre poblaciones**, mientras que al error se le llama **dentro de poblaciones**. También se presenta a continuación la distribución F, que es necesaria para el funcionamiento del anova.

2.2.4. Distribución F

Esta distribución se utiliza para comparar las varianzas de dos poblaciones normales, cuando se tienen muestras grandes independientes comparar efectos de dos o más muestras con respecto a unos factores, [Infante S. y Zárate G. (1990)]. Como se puede observar la distribución F (la función de densidad de probabilidad es conocida como distribución F) es importante para este estudio. Algunas propiedades interesantes [Triola M. (1999)], como son:

1. La distribución F es no simétrica; se sesga hacia la derecha.
2. Los valores de F pueden ser 0 o positivos, pero no negativos.
3. La forma de la distribución F cambia para cada grado de libertad.

Como se muestra a continuación en la figura 2.6, las tres propiedades mencionadas.

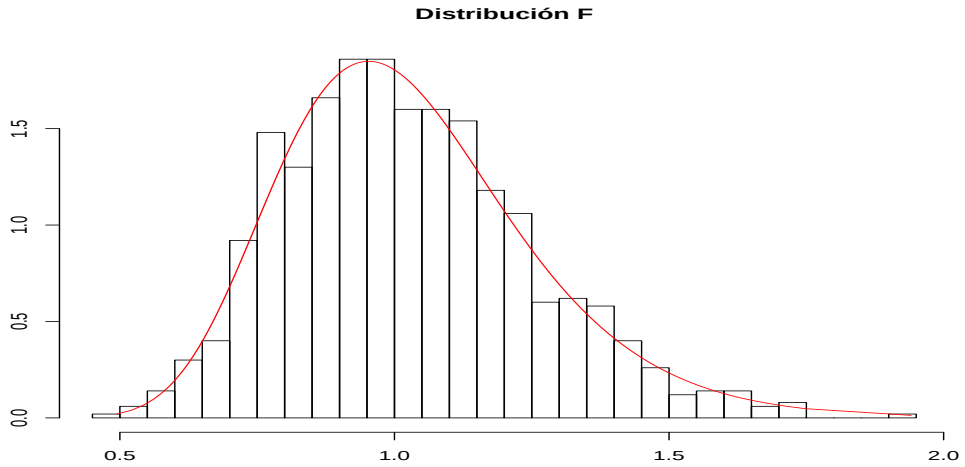

Figura 2.6: Gráfica de la Fisher-Snedecor, con datos simulados con (df = grados de libertad) $df = 100$ y $df = 100$.

Diseño para la Comparación

Este análisis permite saber si hay alguna disimilitud entre la hipótesis nula denominada H_0 contra la hipótesis alternativa llamada H_a, escribiendo esta proposición en un lenguaje estadístico, se tiene:

$$\begin{aligned} H_0: &\quad \mu_{nor} = \mu_{unif} = \mu_{exp} \\ &\quad V_s \\ H_a: &\quad Exista\ alguna \neq . \end{aligned}$$

Con un nivel de significancia, por ejemplo un $\sigma = 0.05$. El Anova se basa en una prueba F explicada en el capitulo2. En este análisis se tiene el valor observado y el valor teórico, como se puede ver en la gráfica de la figura 2.7:

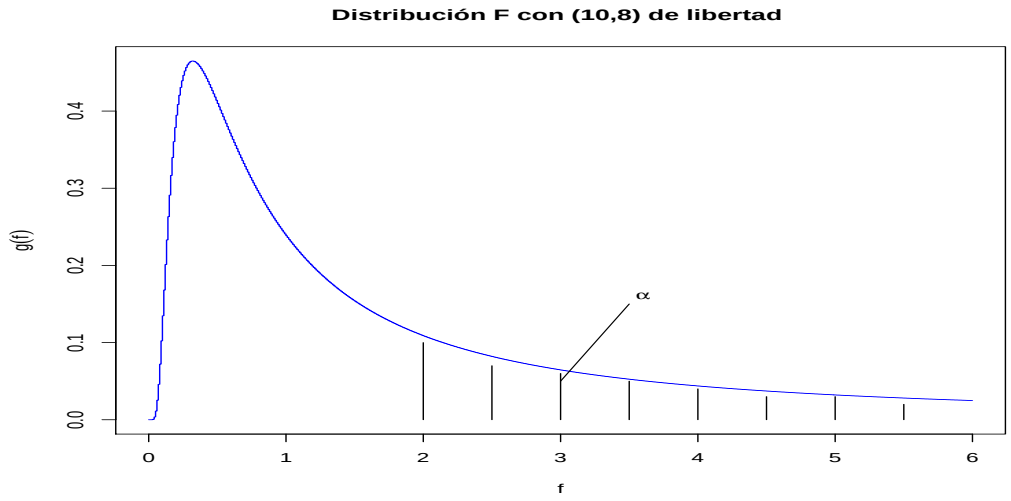

Figura 2.7: Ejemplo de región de rechazo para H_0.

Fijando un nivel de significancia propuesto por el investigador, se rechaza H_0 cuando el estadístico F cae en la región rayada del gráfico (Región de rechazo). Usualmente, se toma como nivel de significancia = 0.05. El área de la cola es 0.05 y buscamos en la tabla F de Fisher el valor de $F^0_{k-1,k(n-1)}.05$, donde $k-1$ es el grado de libertad del numerador y $k(n-1)$es el al grado de libertad del denominador.

En la sección de Resultados, se analizan los resultados obtenidos al ejecutar la base de datos obtenida mediante los supuestos de construcción propuestos en el capítulo 3 para los métodos EM y Kmedias sobre el sistema R, obtenidos mediante RWeka. Para este fin se utilizan dos funciones importantes en el análisis de datos, que son ***lm***() y ***anova***(), estas dos funciones ya están implementadas en el entorno.

En un análisis automatizado, se introduce el modelo ANOVA y se obtiene una tabla, con los atributos **Sum Sq** ≡ suma de cuadrados, **Mean Sq** ≡ media de cuadrados, **F value** ≡ valor F, **P(X > F)** ≡ P valor. A continuación se muestra el código para la obtención del anova.

Modelo $<-$ lm(Y∼Distribucion*Metodo* Ndtos*Nvar*Ncluster*Tamaño,data=datos)
Result $<-$ anova(modelo)
Result

Tabla 2.2: Código para realizar el anova en "R"

A continuación se presenta una breve descripción sobre el tema de generación de números aleatorios.

2.3. Generación de números aleatorios

La generación de números aleatorios forma parte importante en las metodologías de los algoritmos de simulación de sistemas. A demás tiene muchas más aplicaciones como en criptografía o encriptación de datos (ayuda a tener comunicaciones seguras en datos), en la aplicación de generación de claves secretas, como también se usan en máquinas de azar o en video juegos y los algoritmos genéticos [Sevillano G. (2005)].

Los números aleatorios son sucesiones de números seleccionados al azar que provienen de una función de densidad uniforme, es decir, es una función de densidad de probabilidad en la que todo número tiene la misma probabilidad de ser escogido en un intervalo definido [Sáez G. (2000)].

2.3.1. Números Aleatorios

Una secuencia de números aleatorios $R1, R2, \cdots, R_n$ debe tener dos importantes propiedades estadísticas: uniformidad e independencia. Cada número aleatorio R_i es una muestra independiente tomada de una distribución continua uniforme entre cero y uno.

- Si el intervalo $(0, 1)$ es dividido en n clases, o sub intervalos de longitudes iguales, el número esperado de observaciones en cada intervalo es N/n , donde N es el número total de observaciones.
- La probabilidad de observar un valor en un intervalo en particular es independiente de los valores previamente observados.

2.3.2. Números pseudoaleatorios

Los números pseudoaleatorios, son generados a partir de una función determinista pero aparentan ser aleatorios. Estos números se generan a partir de un valor inicial aplicando iterativamente la función, hay que cuidar las semillas de los números aleatorios, porque después de un tiempo ya no se cumple la aleatoriedad. Una sucesión de números pseudoaleatorios se somete a diversos test para medir el grado de su aleatoriedad, es decir, para ver hasta qué punto es similar a una sucesión aleatoria [Sáez G. (2000)].

Se mencionan algunos algoritmos de generación de números pseudoaleatorios.

Generador de Congruencia Lineal, este método fue introducido por [Lehmer D. (1949)]; Serrano J. (2009)]. Produce una secuencia de números enteros, $X_1, X_2, ...$, entre cero y $m-1$ de acuerdo a la siguiente relación recursiva:

$$X_{n+1} = (aX_n + c)\ mod\ m,\ con 0 \leq X_n \leq m$$

donde a es el multiplicador, c es el incremento y m es el módulo.

A este método se le asocia un concepto de periodo, que es una sub cadena de una serie en donde no hay repetición, y longitud de periodo, es el número de elementos de dicha sub cadena.

En este tipo de métodos, cada elemento depende del anterior. Si $c \neq 0$, se tiene el **método de congruencia mixta** [Thomson W. (1958)]. Cuando $c = 0$, se tiene el **método de congruencia multiplicativa** (1949). La selección de los valores a, c, m, y X_0 afecta fuertemente a las propiedades estadísticas y la longitud del ciclo generado. También, existe el **Generador de Congruencia Cuadráticos**, que nos permite obtener la máxima longitud de periodo m, pero se deben realizar más operaciones, su fórmula es:

$$X_{n+1} = (d\ {X_n}^2 + a\ X_n + c)\ mod\ m,$$

además hay, **Generadores de Métodos Aditivos** a diferencia de los anteriores, estos dependen de dos elementos anteriores de la lista, permiten tener longitudes de periodo de hasta m^2, y como su nombre lo indica son puramente aditivos y por lo tanto más rápidos ya que no usan multiplicadores. Algunos de ellos son:

$$\begin{aligned} Fibonacci\ en\ 1950, &\qquad X_{n+1} = (X_{n-1} + X_n)\ mod\ m, \\ Green, &\qquad X_{n+1} = (X_n + X_{n-K})\ mod\ m, con\ K \geq 16. \end{aligned}$$

Mitchell Moore (1958), en este método se verifica que m sea *par* y se deben proporcionar 55 semillas aleatorias no todas pares, y su fórmula es:

$$X_{n+1} = (X_{n-24} + X_{n-55})\ mod\ m, con\ n \geq 55.$$

Existen más métodos en la literatura, pero los ya descritos cumplen con la finalidad de ejemplificar algunos métodos sobre la teoría de números aleatorios. Se revisaron otros textos aunque no hay afirmaciones, ni frases completas, tales para dar crédito, pero, es necesario poner las bibliografías ya que son textos muy buenos y forman parte de un punto de vista particular, [Tarifa E. (2003); Ross S. (2000)].

2.4. Densidades

En esta sección se describen los rasgos más importantes de las distribuciones Normal, Uniforme, Exponencial usadas para el estudio de comparación.

2.4.1. Distribución Uniforme

En el área de estadística la densidad uniforme es una función de probabilidad cuyos valores tienen la misma probabilidad. El dominio está definido por dos parámetros, α y β, que son sus valores mínimo y máximo, respectivamente.

Su función de densidad de probabilidad es:

$$f(x) = \begin{cases} \dfrac{1}{(\alpha - \beta)}, & \alpha < x < \beta \\ 0, & \text{en cualquier otro caso} \end{cases},$$

La función de distribución es:

$$F(x) = \begin{cases} 0, & x < \alpha \\ \dfrac{x - \alpha}{(\beta - \alpha)}, & \alpha \leq x < \beta \\ 1, & x \geq \beta \end{cases},$$

su media es :

$$E(X) = \frac{\alpha + \beta}{2},$$

y la varianza es :

$$V(X) = \frac{(\alpha - \beta)^2}{12},$$

Se muestra a continuación en la figura 2.8 un gráfico de barras con datos, que tiene la forma de una distribución uniforme continua en un intervalo de 0 a 1.

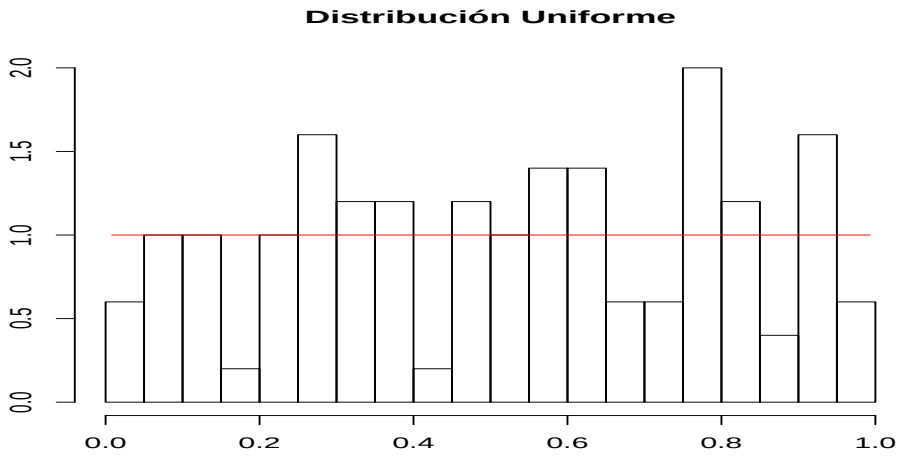

Figura 2.8: Gráfica de datos con densidad Uniforme.

2.4.2. Distribución Exponencial

En estadística la distribución exponencial es una distribución de probabilidad continua con un parámetro $\lambda > 0$, cuya función de densidad es:

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & 0 < x < \infty, \qquad \lambda > 0 \\ 0 \quad , & \textit{en cualquier otro caso} \end{cases}$$

La función de distribución es:

$$F(x) = \begin{cases} 1 - e^{-\lambda x}, & \textit{para} \quad x \geq 0 \\ 0 \quad , & \textit{para} \quad x < 0, \end{cases}$$

su media es :

$$E(X) = \frac{1}{\lambda},$$

y su varianza es :

$$V(X) = \frac{1}{\lambda^2}.$$

Se muestra a continuación un gráfico de barras con datos en la figura 2.9, que tienden a una distribución exponencial.

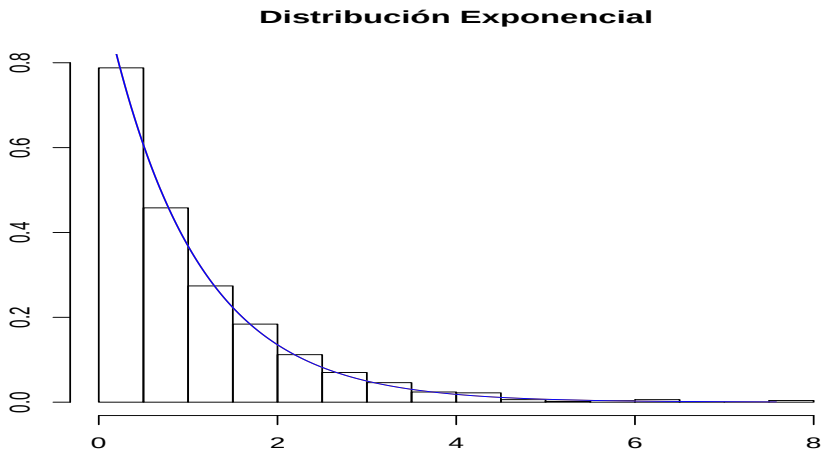

Figura 2.9: Gráfica de datos con densidad Exponencial.

2.4.3. Distribución Normal

La distribución de probabilidad de Gauss o Normal, es una distribución con un uso frecuente en la estadística y teoría de probabilidad debido a sus propiedades, pues esta distribución se tiene cuando hay mucha información de cualquier tipo de distribución, ya sea continua, discreta o mixta. Es una función de densidad simétrica, que tiene interesantes propiedades, que la hacen utilizarla muy frecuentemente, lo que favorece su aplicación como modelo en fenómenos naturales, sociales, psicológicos, etc. [Gamboa A. (2009); Meneses A. *et al.* (2004); Ross S. (2000)].
Su función de densidad de probabilidad es:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \qquad -\infty < x < \infty,$$

la función de distribución es:

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt, \qquad -\infty < x < \infty,$$

la media es:

$$E(X) = \mu,$$

la varianza es:

$$V(X) = \sigma^2,$$

y su función estándar en el caso especial cuando $\mu = 0$, $\sigma = 1$, se muestra a continuación:

$$f_Z(Z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(Z)^2}{2}};$$

A continuación se muestra la figura 2.10, una gráfica con datos que tienen una distribución Normal $(0, 1)$.

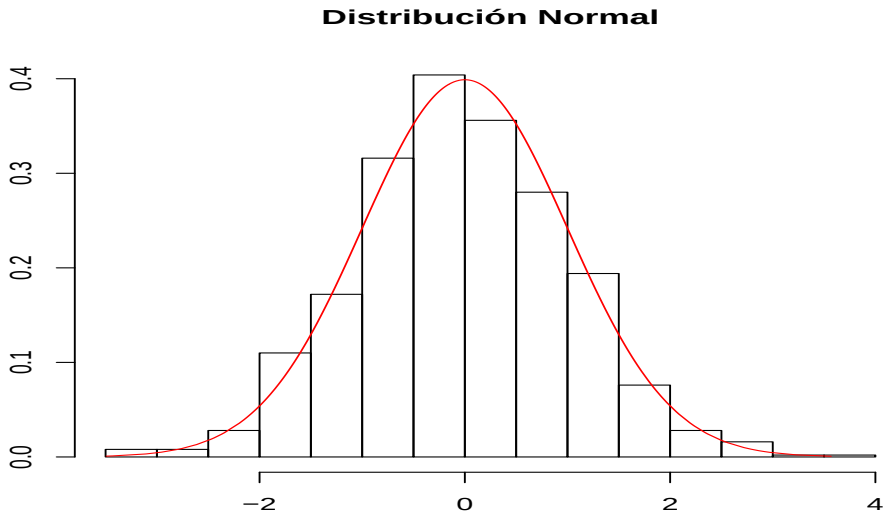

Figura 2.10: Gráfica de datos con densidad Normal.

Se puede observar que las distribuciones continuas pueden tomar cualquier valor y no únicamente un número determinado como ocurre en las distribuciones discretas. Igualmente decimos que una variable es continua cuando uno de los infinitos valores posibles tendrá probabilidad cero y solo se podrá hablar de probabilidad dentro de intervalos, se habla de datos que tienden a una distribución continua, pues, en realidad una computadora no entiende datos continuos.

Capítulo 3

MINERÍA DE DATOS CON WEKA y RWEKA

El propósito de esta sección es realizar una comparación entre los métodos de agrupamiento para encontrar relaciones o patrones entre diferentes características: tomando en cuenta el tipo de densidades, el número de grupos obtenidos, el número de instancias, la frecuencia del número de instancias y el número de variables.

Particularmente en este trabajo se muestra el proceso de minería de datos con el sistema WEKA, de tal manera que se detalla paso a paso, cada una de las interfaces del sistema en correspondencia con cada una de las etapas de minería de datos propuestas en el capítulo 2.

Los datos que se utilizan para la minería se generan en el sistema R, los cuales tienen características específicas y apropiadas. En este caso se tiene el control de manipular los datos por medio de la simulación.

Se ejecutan los procedimientos de agrupamiento EM, SimpleKMeans y CobWeb implementados en WEKA. Posteriormente se comparan los dos primeros con RWeka.

3.1. Proceso de Minería de Datos con WEKA

Siguiendo con la metodología de [Zhengxin C. (2001)] en minería de datos, se necesitan los siguientes pasos, para minar los datos:

* Tener un objetivo,
* Tener una colección de datos,
* Preparar los datos a nuestras necesidades,
* Encontrar el algoritmo (o algoritmos) necesarios e indicados,
* Aplicar los métodos a nuestros datos (entrenarlos) y
* Evaluar el conocimiento adquirido.

3.1.1. Definir el Problema

Encontrar agrupamientos de instancias para diferentes algoritmos, con distintas distribuciones de probabilidad, siguiendo el proceso de minería de datos.

3.1.2. Colección de Datos

La colección consta de datos generados por el sistema R, los cuales tienen características específicas y apropiadas. En este caso se tiene el control de manipular los datos por medio de la simulación.

3.1.3. Preparación de los Datos

Como se menciona con anterioridad, la preparación de datos juega un papel muy importante en minería de datos, así como en el documento de Corporation Two Crows Corporation (1999)

Los datos se construyen usando las funciones que hay en el paquete R, para generación de números aleatorios bajo el modelo considerado, es decir, se crean muestras aleatorias independientes idénticamente distribuidas, se utiliza R ya que nos asegura la independencia. Por ejemplo: los datos de un archivo se generan con la distribución

Normal con igual varianza y medias diferentes, de tal manera que exista un orden creciente entre las medias; el siguiente archivo contiene la generación de cuatro variables con las mismas especificaciones, de las cuales el total de ellas son independientes a las tres variables del archivo anterior con esa misma distribución, en otras palabras. "Por construcción los datos no se repiten y son independientes"no se utilizan en el nuevo archivo, entonces el archivo con cuatro variables no dependen del archivo anterior con tres variables. Obsérvese que se seguirá estrictamente la metodología, propuesta en el estudio de Von A. y Mair P. (2008)

El formato requerido para la realización de las corridas no es necesariamente el ARFF, pues WEKA también acepta el formato CSV y ese formato ya lo proporciona el sistema R mediante la instrucción *write.csv (datosGuardar, file = dirección_ nombre").* En este caso de simulación, los datos no se necesitan limpiar de datos outliers o erróneos, ya que cumplen con las especificaciones que se precisan y los métodos que se utilizan para la comparación no requieren un formato estricto ó necesario. Pero, el formato es la especificación en que los datos deben estar estructurados para que el sistema los pueda leer, se puede ver en el trabajo de Carlos J. (2003) en donde se ejemplifica el ARFF en extenso. El formato de WEKA consta de una estructura bien definida de tres partes, como a continuación se muestra.

@relation <nombre_de_la_relación>; si el nombre de la relación contiene algún espacio será necesario expresarlo entre comillas, cabe mencionar que es de tipo cadena (String de JAVA).

En este apartado se declaran los atributos junto a su tipo.
@attribute <nombre_de_las_variables> <tipo>

* NUMÉRIC, números reales.
* INTEGER, números enteros.
* DATE, indica fechas con su respectivo formato
* STRING, expresa un conjunto de caracteres 'texto' y
* ENUMERADO contiene entre llaves y separados por comas los posibles valores (caracteres o cadenas de caracteres) que puede tomar el atributo.

Por ejemplo, si tenemos un atributo que indica la forma de un objeto podría definirse como sigue:

@attribute forma {Redondo,Cuadrado,Esferico}

Los datos que componen la relación separando entre comas, los atributos y con saltos de línea las relaciones.

@data <valores_de_los_atributos>

En el caso de que algún dato sea desconocido se expresará con un símbolo de cerrar interrogación ("?"). Es posible añadir comentarios con el símbolo "%", que indicará que desde ese símbolo hasta el final de la línea es todo un comentario. Los comentarios pueden situarse en cualquier lugar del fichero. Un ejemplo del formato se da a continuación:

_____Prueba.arff_____

1. Archivo de prueba para Weka.
2. @relation prueba
3. @attribute nombre STRING
4. @attribute ojo_ izquierdo {Bien,Mal}
5. @attribute dimension NUMERIC
6. @attribute fecha_analisis DATE "dd-MM-yyyy HH:mm"
7. @data
8. Antonio,Bien,38.43,"12-04-2003 12:23"
9. 'Maria Jose',?,34.53,"14-05-2003 13:45"
10. Juan,Bien,43,"01-01-2004 08:04"
11. Maria,?,?,"03-04-2003 11:03"

Enseguida se muestra la figura 3.1, extendiendo el menú de opciones para los diferentes tipos de archivos que acepta WEKA.

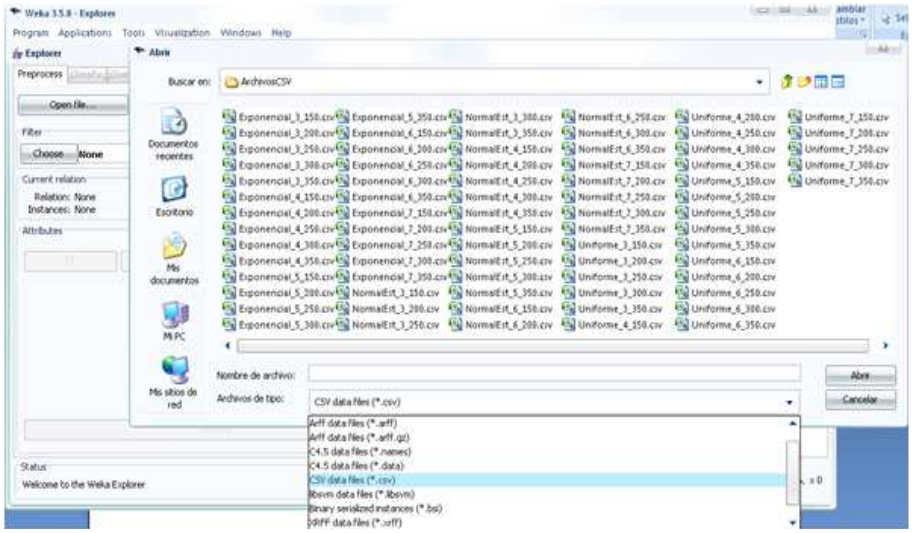

Figura 3.1: Algunos formatos aceptados por WEKA para la minería de datos.

3.1.4. Preprocesamiento de los datos

En esta parte se eliminan o se detectan valores inusuales, ya sean por error de captura o simplemente naturales. En la simulación se tiene el control sobre los datos y no se necesita hacer ningún tipo de filtrado, pero, a continuación se muestra la interfaz del sistema WEKA, situándose en la sección de filtrado. WEKA tienen una gran variedad de métodos de filtrado. Para acceder a esta propiedad sólo hay que pulsar el botón ***Choose*** en la parte que se denomina ***Filter***, se despliega un directorio en forma de árbol en el cual se selecciona la opción que se necesite. Como se muestra en la figura 3.2.

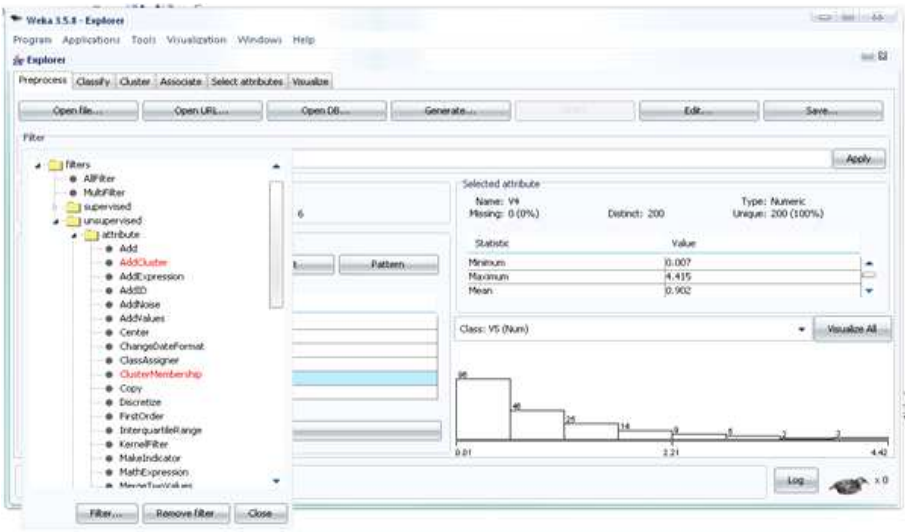

Figura 3.2: Vista de WEKA después de pulsar **Choose**.

3.1.5. Seleccionar un Método Apropiado de Minería

En este apartado se lleva la selección del algoritmo o algoritmos para la exploración, en este caso, se utilizan técnicas de agrupamiento, pues el objetivo es el de comparar métodos que llevan a un mismo fin, pero con metodologías diferentes, por ejemplo: El K-medias utiliza distintas métricas (distancia euclidiana, de Manhatan, de Mahalanobis, etc.) fijando un centroide, el método EM utiliza un método denominado *finite mixture* por eso se dice que es un algoritmo probabilístico y por último CobWeb que es un método de agrupamiento jerárquico, se caracteriza porque utiliza aprendizaje incremental, esto es, realiza las agrupaciones instancia a instancia, que va formando un árbol de clasificación, donde las hojas representan los segmentos y el nodo raíz engloba por completo el conjunto de datos de entrada. Este último método maneja un concepto probabilístico denominado utilidad de categoría, la cual utiliza la media y la varianza.

3.1.6. Entrenamiento o prueba de datos, Aplicación del modelo

En esta sección se hace la ilustración de los resultados obtenidos al ejecutar los algoritmos EM, K-medias y CobWeb. Se inicia el análisis con el método EM el cual proporciona una K 'la cual determina el número de grupos a obtener'y con esta K, se usa para el método de K-medias en forma óptima.
En este caso sólo se dan tres de los resultados en forma resumida del algoritmo escogido pues se tienen 75 resultados sobre las simulaciones.

Prueba de datos con el Método EM

Se ejecuta el método EM del sistema WEKA sobre un archivo, parte de la base de datos. Los resultados se pueden visualizar de 2 formas diferentes por el tipo de algoritmo, es decir, obtener un gráfico en 2D en forma de plano cartesiano en coordenadas asumidas por el modelo; obtener una representación en un formato específico de texto, en el cual, se visualiza el contenido de cluster con la información de su media, desviación estándar y el total de instancias en cada cluster.

Primer Formato de salida después de la ejecución, en el sistema WEKA.

Al pulsar click derecho en el experimento EM de la lista de resultados, se encuentra el cuadro ***Result list*** y ahí aparece un segundo menú de opciones del cual elegimos la opción ***view in separate Windows***, o simplemente se visualiza del lado derecha en la sección denominada ***Clusterer output*** y obtenemos los siguientes resultados: en la Tabla 3.3.

=== Run information===		
Scheme:		
	weka.clusterers.EM -I 100 -N -1 -M 1.0E-6 -S 100	
Relation:	NormalEst_7_350	
Instances:	350	
Attributes:	7; V1, V2, V3, V4, V5, V6, V7	
Test mode:	evaluate on training data	
=== Model and evaluation on training set ===		
EM		
==		
Number of clusters selected by cross validation: 2		
		Cluster
Attribute	0	1
	(0.49)	(0.51)
==========		
V1		
mean	0.00096	0.1517
std. dev.	1.0468	1.0136
V2		
mean	-0.0137	0.0286
std. dev.	0.9862	1.0179
V3		
mean	-0.0671	-0.011
std. dev.	1.0584	1.0448
V4		
mean	-0.0805	-0.1216
std. dev.	0.9783	1.0133
V5		
mean	-0.0136	0.1166
std. dev.	1.0184	1.0217
V6		
mean	0.0263	0.1394
std. dev.	0.9641	0.9638
V7		
mean	-0.0746	-0.0464
std. dev.	1.0457	0.9777
Clustered	Instances	
0	169 (48 %)	
1	181 (52 %)	
Log likelihood:		-15.96423

Tabla 3.1: Corrida del método EM con datos que tienden a una distribución Normal.

La salida se muestra tal cual en la figura 3.3 del sistema WEKA:

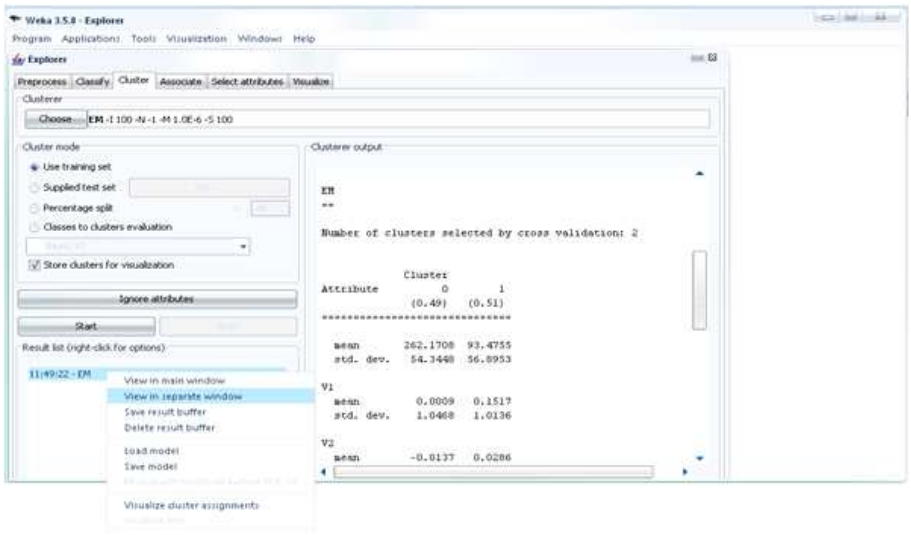

Figura 3.3: Selección de la opción de visualización de resultados en una ventana independiente.

Segundo formato de salida después de la ejecución, en el sistema WEKA. Al pulsar click derecho en el experimento EM de la lista de resultados se encuentra el cuadro ***Result list***, en este aparece un segundo menú de opciones y se elige la opción ***Visualize cluster assignments***, se obtienen los siguientes resultados, como se muestra en la figura 3.4:

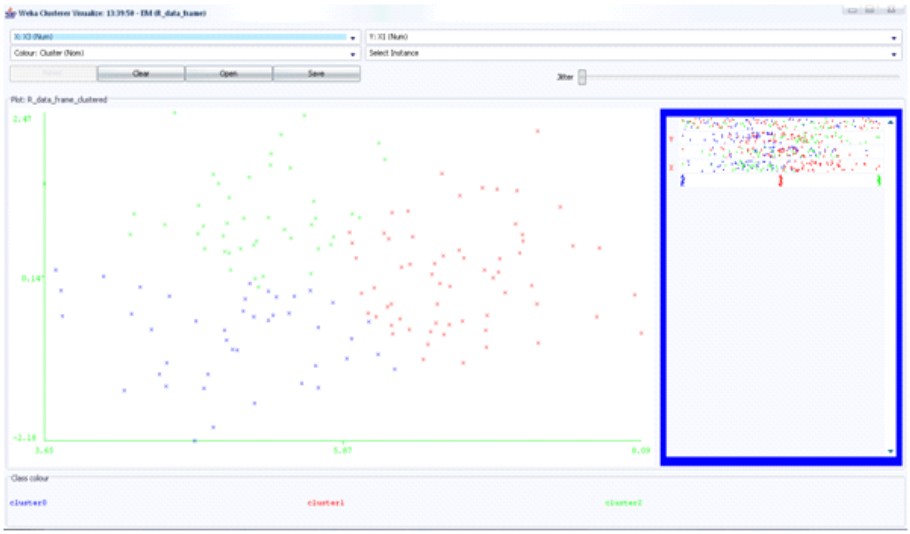

Figura 3.4: Selección de la opción de visualización de asignación de cluster.

Prueba de datos del Método K-Meads o K-medias

Se ejecuta el método K-medias del sistema WEKA sobre un archivo, parte de la base de datos. Los resultados se pueden visualizar de 2 formas diferentes por el tipo de algoritmo, es decir, obtener un gráfico en 2D en forma de plano cartesiano en coordenadas asumidas por el modelo; obtener una representación en un formato específico de texto, en el cual, se visualiza el contenido de cluster con la información de instancias en cada cluster.

Primer Formato de salida después de la ejecución, en el sistema WEKA.

Al pulsar click derecho en el experimento K-medias de la lista de resultados, se encuentra el cuadro ***Result list*** y ahí aparece un segundo menú de opciones del cual elegimos la opción ***view in separate Windows***, o simplemente se visualiza del lado derecha en la sección denominada ***Clusterer output*** y obtenemos los siguientes resultados: en la Tabla 3.2.

=== Run information===					
Scheme:	weka.clusterers.SimpleKMeans -N 4 -S 10				
Relation:	R_data_frame				
Instances:	150				
Attributes:	3; X1, X2, X3				
Test mode:	evaluate on training data				
=== Model and evaluation on training set ===					
kMeans					
====					
Number of iterations: 6					
Within cluster sum of squared errors: 17.389908450556554					
Missing values globally replaced with mean/mode					
Cluster centroids:					
	Cluster#				
Attribute	Full Data	0	1	2	3
	(150)	(33)	(40)	(39)	(38)
==========					
X1	0.568	0.2088	0.6817	0.7829	0.5399
X2	0.5181	0.548	0.3356	0.406	0.7994
X3	0.5191	0.7461	0.3219	0.8064	0.2347
		Clustered Instances			
0	33 (22 %)	2	39 (26 %)		
1	40 (27 %)	3	38 (25 %)		

Tabla 3.2: Corrida del método Kmedias con datos que tienden a una distribución Uniforme.

La salida se muestra tal cual en la figura 3.5 del sistema WEKA:

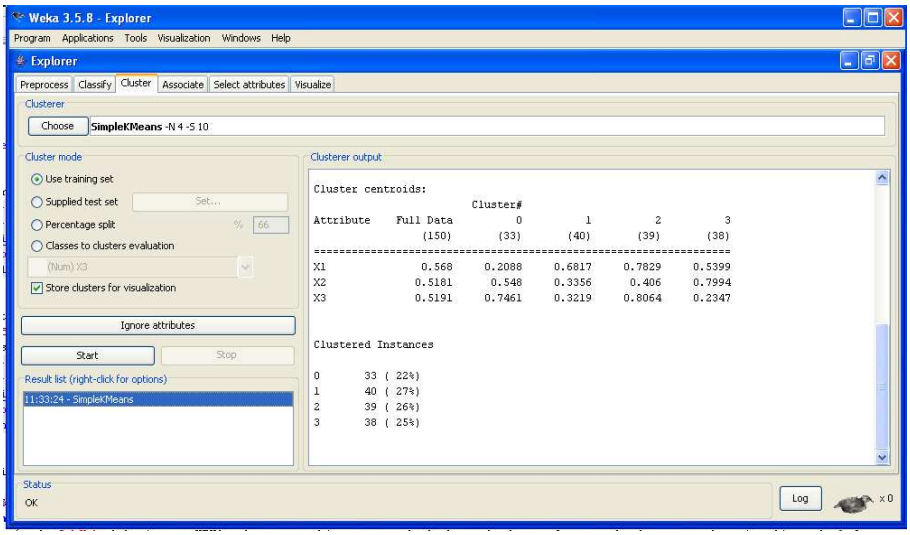

Figura 3.5: Selección de la opción de visualización de resultados en una ventana independiente.

Segundo formato de salida después de la ejecución, en el sistema WEKA. Al pulsar click derecho en el experimento Kmedias de la lista de resultados se encuentra el cuadro ***Result list***, en este aparece un segundo menú de opciones y se elige la opción ***Visualize cluster assignments***, se obtienen los siguientes resultados, como se muestra en la figura 3.6:

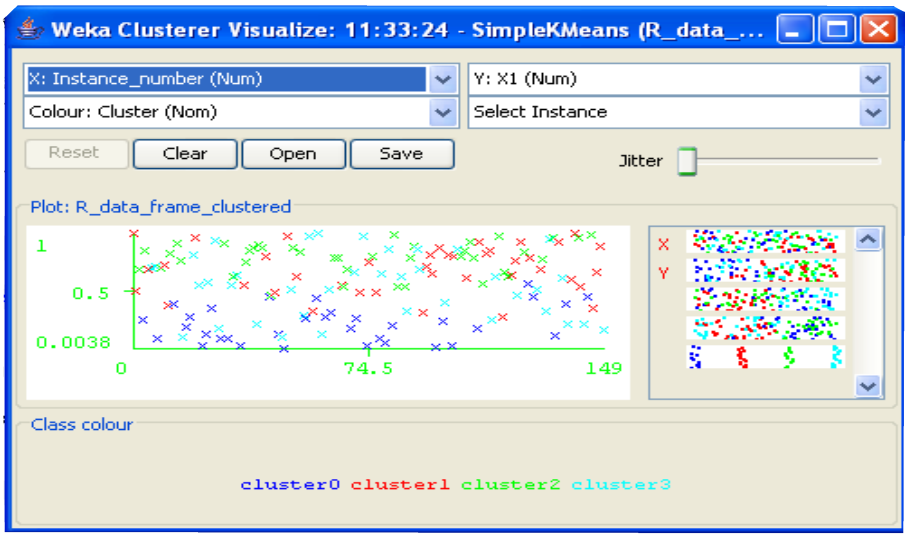

Figura 3.6: WEKA selección de la opción de visualización de asignación de cluster.

Prueba de datos con el Método CobWeb

Se ejecuta el método CobWeb del sistema WEKA sobre un archivo, parte de la base de datos. Los resultados se pueden visualizar de 3 formas diferentes, es decir, obtener un gráfico en 2D en forma de plano cartesiano en coordenadas asumidas por el modelo; obtener una representación en un formato específico de texto, en el cual, se visualiza el contenido de cluster con la información del total de instancias en cada cluster; y en el caso de COBWEB se obtiene un árbol, en donde, los hojas contienen el número total de instancias y el nodo la descripción de esa instancias.
Primer Formato de salida después de la ejecución, en el sistema WEKA.

Al pulsar click derecho en el experimento EM de la lista de resultados, se encuentra el cuadro ***Result list*** y ahí aparece un segundo menú de opciones del cual elegimos la opción ***view in separate Windows***, o simplemente se visualiza del lado derecha en la sección denominada ***Clusterer output*** y obtenemos los siguientes resultados: en la Tabla 3.3.

=== Run information===	
Scheme:	weka.clusterers.SimpleKMeans -N 4 -S 10
Relation:	R_data_frame
Instances:	150
Attributes:	3; X1, X2, X3
Test mode:	evaluate on training data
=== Model and evaluation on training set ===	
Number of merges: 1	
Number of splits: 1	
Number of clusters: 3	
node 0 [150]	
— leaf 1 [149]	
node 0 [150]	
— leaf 2 [1]	
Clustered Instances	
1	146 (97 %)
2	4 (3 %)

Tabla 3.3: Corrida del método CobWeb con datos que tienden a una distribución Exponencial.

La salida se muestra tal cual en la figura 3.7 del sistema WEKA:

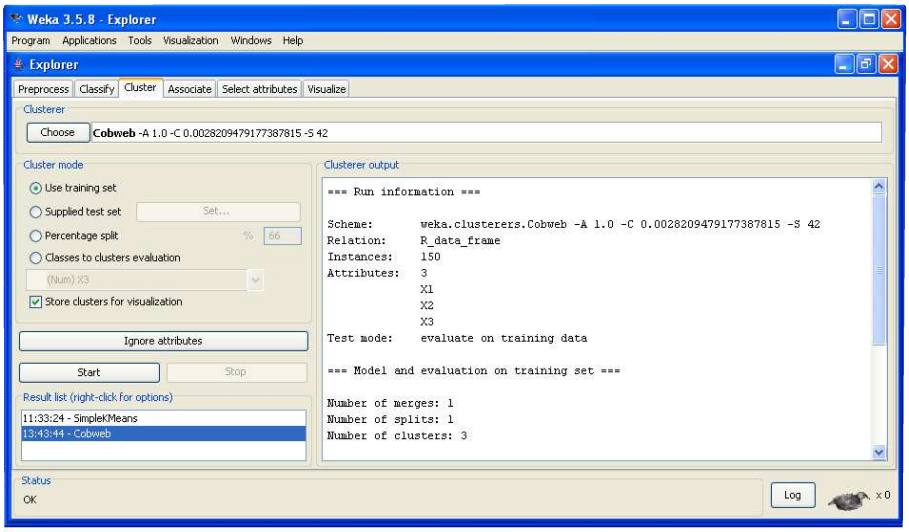

Figura 3.7: Selección de la opción de visualización de resultados en una ventana independiente.

Segundo formato de salida después de la ejecución, en el sistema WEKA. Al pulsar click derecho en el experimento EM de la lista de resultados se encuentra el cuadro ***Result list***, en este aparece un segundo menú de opciones y se elige la opción ***Visualize cluster assignments***, se obtienen los siguientes resultados, como se muestra en la figura 3.8:

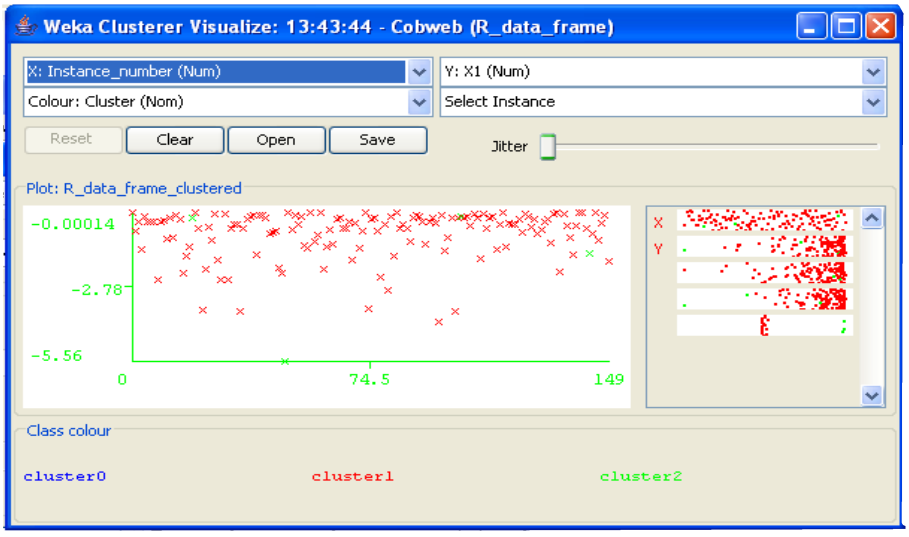

Figura 3.8: WEKA selección de la opción de visualización de asignación de cluster.

Cada método tiene una forma distinta de visualizar el cluster, como en el caso de CobWeb, en donde se visualiza un árbol, como se muestra a continuación en el siguiente gráfico.

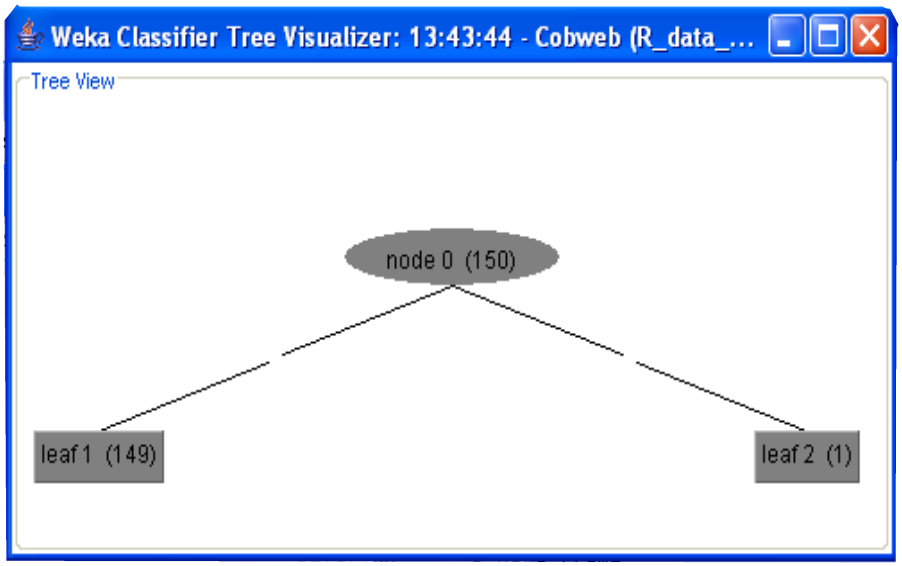

Figura 3.9: WEKA selección de la opción de visualize tree.

3.2. Proceso de Comparación de los Algoritmos con RWeka

En esta sección se presenta la ejecución de algoritmos de clasificación automática no supervisada (EM y Kmedias). No se utiliza CobWeb por la complejidad en la configuración de los parámetros (*acutty y cutoff*) necesarios para el funcionamiento del método, además de necesitar variables discretas que incluso deben ser nominales, al mismo tiempo el grado de confianza sobre el algoritmo en datos numéricos es mínimo como se puede observar en el trabajo [Fisher D. (1987)]. Pero, en nuestro caso los datos que se utilizan en el experimento son continuos.

La ejecución se lleva acabo sobre una plataforma Windows XP, con el software R herramienta hermana de S creada por AT&T, con la única diferencia de tener licencia libre, y como se ha dicho anteriormente la utilización de RWeka crea una interfaz de los algoritmos implementados en WEKA con entorno java a R con scripts.

De esta manera se evita hacer manualmente los siguientes pasos:

* Cargar el archivo en el paquete WEKA, al pulsar *Open file,*
* ir a la pestaña *Cluster,*
* elegir el método, en la parte *Clusterer* pulsando *Choose,*
* configurar el método seleccionado,
* pulsar el botón *start,*
* Repetir los casos cuantas veces sean necesarios, en nuestro caso se debe repetir 3500 veces,
* capturar los resultados en una hoja de datos,
* calcular o trasladar estos datos a un programa que calcule la tabla de ANOVA para poder estudiar el resultado final.

3.2.1. Colección de Datos con RWeka

En este apartado se crea la base de datos mediante las instrucciones ***rnorm()***,***runif()*** y ***rexp()***, pertenecientes a R. Como en el capitulo anterior, sólo con la diferencia de la instrucción ***write.arff ()***, la cual se encarga de crear un archivo con formato arff, nativo de WEKA, ya explicado con anterioridad. A continuación se muestra el código en R para la creación de los archivos utilizando la librería RWeka, y crear la base de datos ya descrita en el capitulo 3 sub sección 3.1.3.

```
library(RWeka)
a < −(“E:/tesis/tesis_santiago/Capitulo_3/Experimento_2/NormalEst_”)
c < −(“_”);          valor< −150;
e < −(“.arff”);          y< −3;
a3< −(“E:/tesis/tesis_santiago/Capitulo_3/Experimento_2/Exponencial_”)
a2< −(“E:/tesis/tesis_santiago/Capitulo_3/Experimento_2/Uniforme_”)
for(k in 1:5) {#- - - - - - - - -for1- - -cantidades de 150 a 350
              for(i in 1:5) {#- - - - - - - - -for2- - -variables que van de 3 a 7
              b< −(as.character(i+2))
              d< −(as.character(valor))
              M< − matrix ( ,valor, y)
              M2< − matrix ( ,valor, y)
              M3< − matrix ( ,valor, y)
              media< −0
for(j in 1:y) {#- - - -for3 crea los datos en formato em arff
              y los deposita en la dirección especifica
              M [ ,j] < − rnorm(valor, media)
              M2[ ,j] < − runif(valor)
              M3[ ,j] < − Exp(M2[ ,j])
              write.arff(data.frame(M) ,file=paste(a[1], b[1],c[1],d[1],e[1],sep=“”))
              write.arff(data.frame(M2),file=paste(a2[1],b[1],c[1],d[1],e[1],sep=“”))
              write.arff(data.frame(M3),file=paste(a3[1],b[1],c[1],d[1],e[1],sep=“”))
              media< −media+3
              }#Fin_for3
              y< −y+1
              }#Fin_for2
              valor< −(150+(k*50))
              y< −3
}#Fin_for1
```

Tabla 3.4: Código en RWeka

3.2.2. Ejecución de Métodos

El siguiente código crea la base de datos para el análisis del ANOVA. La función **make_Weka_clusterer (name, class)**, crea una interface R para la existencia de un objeto WEKA, ya sea de aprendizaje, de filtrado de datos o simplemente para poder visualizar las propiedades del objeto WEKA en R. Utiliza dos parámetros:

name - es la representación en cadena de caracteres (String) de la ruta y el nombre exacto del objeto en notación **JNI** (Java Native Interface)

class - se refiere al nombre de la clase R para la función interface.

Ya con esta función creamos dos objetos ***EM*** y ***KMds***, sus nombres van acorde a sus métodos, si se quiere visualizar las propiedades de cada objeto, hay que ejecutar la función ***WOW*** **(*EM/KMds*)**, ésta función nos da la pauta para saber la configuración del algoritmos de clustering y no dejar la configuración de default. Cada objeto necesita el archivo del cual va a sustraer el conocimiento, por lo tanto se necesita la función ***read.arff*** **(*file*)**, ***file*** es la representación en cadena de caracteres (String) de la dirección exacta en donde se encuentra tal archivo junto con su nombre. También los objetos EM/KMds, necesitan una configuración necesaria y se hace mediante la función ***Weka_control*** **( )**, la cual depende de que método se use para saber cuáles son sus parámetros. En nuestro caso, nos interesa modificar el número de cluster a formar. Por lo cual, teniendo el conocimiento necesario para hacer la corrida con el objeto creado **EM(** ***ll, Weka_control*****(** ***N = 3*****))** y **KMds(*ll, Weka_control*(*N = 3*))**, el primer parámetro tiene en memoria los datos a ser minados y la función Weka_control sólo se le configura con en número de cluster a formar, que es necesario para el análisis. Ahora ya ejecutados los métodos de clustering estos nos dan un objeto, el cual tiene ahora la capacidad de visualizar la corrida como en WEKA (Formato). Como las observaciones obtenidas no proporcionan en que cluster pertenecen, entonces utilizamos la función ***predict*** **(*temp*)**, la cual proporciona un vector, que tiene una relación entre número de instancias con cada cluster, la relación se puede describir de la siguiente manera:

Instancias	1	2	3	4	5	6	7	8	$\cdots$	150
Cluster	1	0	2	0	1	2	1	1	$\cdots$	0

El 0, 1, 2 representan los cluster entonces la instancia 1 esta en el cluster 1, la instancia 2 esta en el cluster 0 y así sucesivamente (es la mejor forma de explicar la relación). El sistema R nos ofrece herramientas para el análisis de datos, una de estas herramientas es la función **table(*predict* (*temp*))**, al cual se le proporciona un vector como entrada y de salida se consigue un objeto que contiene en forma resumida la información contenida de la siguiente manera:

Instancias	0	1	2
Cluster	33	22	95

Las funciones importantes para hacer el análisis de forma automática, ya con la información obtenida, se crea un programa en donde se cargué el archivo, se analicen los datos y se pongan en un repositorio para su análisis. Tal y como se muestra a continuación, el código para la ejecución de los algoritmos EM y SimpleKmeans.

```
VectorEM      <−NULL;  VectorKm      <−NULL
VectorNvar    <−NULL;  VectorNdatos  <−NULL
VectorDist    <−NULL;  VectorMtdo    <−NULL
VectorNCtr    <−NULL
kk3i<− 3                    i  <− 1
ind<− 1#indice para el numero de datos 150 a 350
EM<− make_Weka_clusterer("weka/clusterers/EM", "weka_cluster")
KMds<− make_Weka_clusterer("weka/clusterers/SimpleKMeans", "weka_cluster")
ruta<− "C:/Documents and Settings/C&Bmini/Escritorio/tesis/tesis_santiago/
        Capitulo_3/Experimento_2/"
bisNum<−c("Exponencial_","NormalEst_","Uniforme_","150","200","250","300","350")
for(f in 1:3){
for(kk in 1:5){
for(kk2 in ind:5){ #distri #numero de variables #cantidad de datos
dv <− paste(ruta,bisNum[f],as.character(kk+2)"_",
as.character(bisNum[ind+3]),".arff",sep="")
<− read.arff(file=dv)
for(kk3 in 3:7){
temp <− EM(ll,Weka_control(N = kk3i))
temp2 <− KMds(ll,Weka_control(N = kk3i))
VectorResEM <− table(predict(temp))
VectorResKm <− table(predict(temp2))
VectorEM <− cbind(c(VectorEM,VectorResEM)) #Tamaño
VectorKm <− cbind(c(VectorKm,VectorResKm)) #" "
VectorNdatos <− cbind(c(VectorNdatos,as.integer(rep(bisNum[ind+3], times=kk3i))))
VectorNvar <− cbind(c(VectorNvar,rep(as.integer(kk+2),times=kk3i)))
VectorDist <− cbind(c(VectorDist,rep(bisNum[f], times=kk3i))
VectorNCtr <− cbind(c(VectorNCtr,(0:(kk3i-1))))
kk3i<−kk3i+1;
}
kk3i<− 3
ind <− ind+1
i <− i+1
}
ind<−1
```

Tabla 3.5: Código en RWeka para la ejecución de los métodos. Primera parte

```
}
}
VectorMtdo < − cbind(c(rep(as.integer(1),times=length(VectorEM))
,rep(as.integer(2),times=length(VectorKm))))
Metodo < − factor(VectorMtdo)
Distribucion < − factor(c(VectorDist,VectorDist))
VectorEMKm< −c(VectorEM,VectorKm)
indice< −1;    indice< −1;    conta< −4
Vresp< −NULL
leng< −length(VectorEMKm)
while(indice<leng){
    datoss< − VectorEMKm[indice:(indice2)]
    datoss< − VectorNCtr[indice:(indice2)]
    Vres < −pbinom(datoss, size=sum(datoss), prob=propp)
    Vresp< −cbind(Vresp,Vres)
    indice< −indice+3
}
VectorNdatos < − c(VectorNdatos,VectorNdatos)
VectorNvar < − c(VectorNvar,VectorNvar)
VectorDist < − c(VectorDist,VectorDist)
VectorNCtr < − c(VectorNCtr,VectorNCtr)
datos < − data.frame(Distribucion=Distribucion, Metodo=Metodo,
          Ndtos=VectorNdatos,Nvar=VectorNvar,Ncluster=
VectorNCtr,Tamaño=as.vector(VectorEMKm), Y=as.vector(Vresp))
```

Tabla 3.6: Código en RWeka para la ejecución de los métodos. Segunda parte

Obsérvese que este lenguaje de programación es sensible, en cuanto al cambio de mayúsculas y minúsculas, por tanto **a** es diferente de **A**, las palabras reservadas y funciones están en la misma situación. Si se cambia una letra de minúscula a mayúscula o viceversa no reconoce tal función o palabra reservada y envía un mensaje de error. A manera de ejemplo las palabras RWeka y Rweka son diferentes, así que hay que tener precaución. La ventaja es que como es un lenguaje interpretado y no compilado, se detiene en la línea que esta mal y por tanto, es mas fácil de corregir o al menos de darse cuenta en donde esta el error sintáctico.

3.3. Comparación de algoritmos de agrupamiento: un estudio de simulación

Para esclarecer el comportamiento de los métodos de agrupamiento EM y Kmedias, se realiza un estudio de simulación caracterizado de la siguiente manera:
La colección está formada por datos de tres densidades de probabilidad distintas, como son Normal, Uniforme, y Exponencial; se incrementa la cantidad de variables de 3 a 7 y también se incrementa el número de instancias de 150 a 350. Así, se obtienen 25 archivos con datos que tienen una densidad Uniforme, otros 25 archivos con datos de una densidad Normal y finalmente, 25 archivos con datos con densidad Exponencial. En total se trabaja con 101,250 datos. A continuación se muestra la tabla 3.7, en donde se muestran las especificaciones exactas para las simulaciones de los datos utilizados en la minería.

Número de datos (instancias)	Número de variables	Densidad Normal	Densidad exponencial	Densidad uniforme	Número total de archivos
150	3, 4, 5, 6, 7	5	5	5	15
200	3, 4, 5, 6, 7	5	5	5	15
250	3, 4, 5, 6, 7	5	5	5	15
300	3, 4, 5, 6, 7	5	5	5	15
350	3, 4, 5, 6, 7	5	5	5	15
Número total de archivos		25	25	25	75

Tabla 3.7: Descripción del Total de Datos

El diseño resultante es un 3(tipos de simulación) x 2(métodos de agrupamiento) x 4 covariables(Ndts, Nvar,Tamaño,Ncluster). Como variable respuesta del modelo se tomo la distribución binomial que describimos a continuación: sea P la probabilidad e ocurrencia de un evento, y $q = 1 - P$. Sea N el tamaño de la muestra y n la frecuencia observada de un evento. Entonces la probabilidad que n o un número mayor de casos sea observado bajo P es:

$$B(p) = \sum_{i=n}^{N} \binom{N}{i} p^i q^{N-i}$$

Estas probabilidades binomiales se obtuvieron del sistema R, se muestra a continuación el código de ejecución :

```
indice   < −1
indice2  < −3
conta    < −4
Vresp    < −NULL
leng     < −length(VectorEMKm)
         while(indice <leng){
         if(conta==7)
         conta < −3
         datoss < −VectorEMKm[indice:indice]
         propp < −VectorNCtr[indice:indice2]
         propp < −propp/10
         Vres < −pbinom(datoss, size=sum(datoss), prob=propp)
         Vresp < −cbind(Vresp,Vres)
         indice < −indice2+1
         indice2< −indice2+conta
         conta < −conta+1
         }
```

Tabla 3.8: Código en "R"para la ejecución de la variable respuesta

3.4. Resultados y discusión

A continuación se presenta la Tabla de Anova con todas las covariables e interacciones, posteriormente se muestran estos mismos resultados en forma gráfica. La primera presentación da a simple vista si hay diferencias significativas entre nuestra variable respuesta, con la simbología proporcionada del sistema R dentro del anova; la segunda muestra la evidencia de que tan significativa es esa diferencia, y esto nos lleva a saber si hay relación entre el dato (distribución perteneciente) y el método de agrupamiento. Se muestra el análisis de varianza (ANOVA) del modelo considerado anteriormente.

3.4.1. Resultados

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Distribucion	2	9.86	4.93	51.9256	<2.2e-16	***
Metodo	1	2.51	2.51	26.4415	2.86E-07	***
Ndtos	1	0.78	0.78	8.1923	0.004231	**
Nvar	1	0.01	0.01	0.0528	0.818234	
Ncluster	1	0.42	0.42	4.4608	0.034748	*
Tamaño	1	231.39	231.39	2436.8256	< 2.2e-16	***
Distribucion:Metodo	2	9.48	4.74	49.8954	< 2.2e-16	***
Distribucion:Ndtos	2	0.67	0.34	3.5329	0.02932	*
Metodo:Ndtos	1	0.03	0.03	0.2843	0.593943	
Distribucion:Nvar	2	0.01	0.003419	0.036	0.964638	
Metodo:Nvar	1	0.12	0.12	1.2161	0.270203	
Ndtos:Nvar	1	0.0042093	0.004209	0.0443	0.833251	
Distribucion:Ncluster	2	0.17	0.09	0.8995	0.406859	
Metodo:Ncluster	1	0.0001777	0.000178	0.0019	0.965499	
Ndtos:Ncluster	1	2.35	2.35	24.7101	6.97E-07	***
Nvar:Ncluster	1	0.18	0.18	1.9039	0.167728	
Distribucion:Tamaño	2	1.91	0.95	10.0384	4.49E-05	***
Metodo:Tamaño	1	1.84	1.84	19.3466	1.12E-05	***
Ndtos:Tamaño	1	12.74	12.74	134.1954	< 2.2e-16	***
Nvar:Tamaño	1	0.09	0.09	0.9951	0.318568	
Ncluster:Tamaño	1	8.23	8.23	86.6357	< 2.2e-16	***
Distribucion:Metodo:Ndtos	2	0.57	0.28	2.9987	0.049974	*
Distribucion:Metodo:Nvar	2	0.14	0.07	0.7607	0.467403	
Distribucion:Ndtos:Nvar	2	0.07	0.04	0.3766	0.686205	
Metodo:Ndtos:Nvar	1	0.01	0.01	0.1004	0.751399	
Distribucion:Metodo:Ncluster	2	0.17	0.08	0.8732	0.417696	
Distribucion:Ndtos:Ncluster	2	0.29	0.15	1.5515	0.212073	
Metodo:Ndtos:Ncluster	1	0.16	0.16	1.6714	0.196153	
Distribucion:Nvar:Ncluster	2	0.02	0.01	0.0992	0.905563	
Metodo:Nvar:Ncluster	1	0.01	0.01	0.0721	0.788308	
Ndtos:Nvar:Ncluster	1	0.002398	0.002398	0.0253	0.873744	
Distribucion:Metodo:Tamaño	2	2.24	1.12	11.8202	7.64E-06	***
Distribucion:Ndtos:Tamaño	2	0.16	0.08	0.8474	0.428618	
Metodo:Ndtos:Tamaño	1	0.07	0.07	0.7703	0.380179	
Distribucion:Nvar:Tamaño	2	0.72	0.36	3.7724	0.023087	*
Metodo:Nvar:Tamaño	1	0.76	0.76	8.0151	0.004664	**
Ndtos:Nvar:Tamaño	1	0.03	0.03	0.365	0.545775	
Distribucion:Ncluster:Tamaño	2	1.02	0.51	5.3955	0.004573	**

Tabla 3.9: Se obtiene la salida con el paquete "R". Primera parte del ANOVA

Metodo:Ncluster:Tamaño	1	0.29	0.29	3.0208	0.082289.
Ndtos:Ncluster:Tamaño	1	6.08E-05	6.08E-05	0.6E-03	0.979814
Nvar:Ncluster:Tamaño	1	0.0027557	0.002756	0.029	0.864739
Distribucion:Metodo:Ndtos:Nvar	2	0.09	0.05	0.4997	0.606759
Distribucion:Metodo:Ndtos:Ncluster	2	0.46	0.23	2.4437	0.086981.
Distribucion:Metodo:Nvar:Ncluster	2	0.02	0.01	0.0913	0.912761
Distribucion:Ndtos:Nvar:Ncluster	2	0.02	0.01	0.0812	0.921984
Metodo:Ndtos:Nvar:Ncluster	1	3.12E-02	3.106E-02	0.0327	0.856494
Distribucion:Metodo:Ndtos:Tamaño	2	0.45	0.23	2.3957	0.091251.
Distribucion:Metodo:Nvar:Tamaño	2	0.03	0.01	0.1457	0.864398
Distribucion:Ndtos:Nvar:Tamaño	2	0.21	0.11	1.132	0.322496
Metodo:Ndtos:Nvar:Tamaño	1	0.03	0.03	0.326	0.56806
Distribucion:Metodo:Ncluster:Tamaño	2	0.63	0.31	3.2983	0.037057*
Distribucion:Ndtos:Ncluster:Tamaño	2	0.09	0.05	0.4758	0.621409
Metodo:Ndtos:Ncluster:Tamaño	1	0.01	0.01	0.1175	0.731809
Distribucion:Nvar:Ncluster:Tamaño	2	0.01	0.004551	0.0479	0.953208
Metodo:Nvar:Ncluster:Tamaño	1	0.03	0.03	0.2748	0.600156
Ndtos:Nvar:Ncluster:Tamaño	1	0.06	0.06	0.6579	0.417351
Distribucion:Metodo:Ndtos:Nvar:Ncluster	2	0.12	0.06	0.6271	0.534197
Distribucion:Metodo:Ndtos:Nvar:Tamaño	2	0.1	0.05	0.5423	0.581442
Distribucion:Metodo:Ndtos:Ncluster:Tamaño	2	0.07	0.03	0.3461	0.707452
Distribucion:Metodo:Nvar:Ncluster:Tamaño	2	0.06	0.03	0.3372	0.713809
Distribucion:Ndtos:Nvar:Ncluster:Tamaño	2	0.01	0.004022	0.0424	0.95853
Metodo:Ndtos:Nvar:Ncluster:Tamaño	1	0.16	0.16	1.6481	0.1993
Distribucion:Metodo:Ndtos:Nvar:Ncluster:Tamaño	2	0.12	0.06	0.6476	0.523351
Residuals		3654	346.96	0.09	
—					
Signif. codes: 0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' ' 1					

Tabla 3.10: Se obtiene la salida con el paquete "R". Segunda parte del ANOVA

Con un nivel de significancia de un $\alpha = \mathbf{0.05}$, es importante el número de grupos, y las interacciones: distribución - número de datos, distribución-método-número de datos, distribución-número de variables-tamaño, distribución-método-número de cluster-tamaño. Si se toma un nivel de significancia más estricto $\alpha = \mathbf{0.01}$, son significativas : el número de datos y así como sus interacciones, con: método-número de variables-tamaño, distribución-número de cluster-tamaño. Si se toma un $\alpha = \mathbf{0.001}$, son significativas: distribución, método y tamaño, en el caso del α propuesto son importantes: distribución-método, número de datos-número de cluster, distribución-tamaño, método-tamaño, número de datos-tamaño, número de cluster-tamaño, y distribución-método-tamaño; pues tienen diferencias significativas. Las gráficas que aparecen a continuación muestran el comportamiento de la variable respuesta Y,(probabilidad promedio de los cluster resultantes). Como se observa en las tablas 3.9 y 3.10.

3.4.2. Discusión

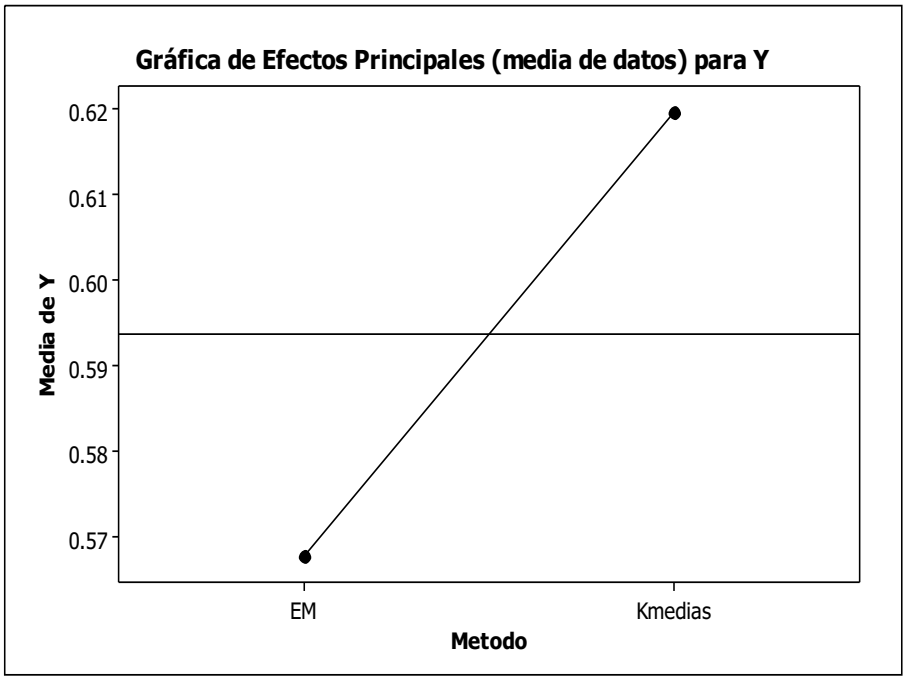

Figura 3.10: Comparación de los dos métodos EM, Kmedias con respecto de Y.

En la figura 3.10 puede apreciarse que, el método k-medias tiene una mayor probabilidad de agrupamiento. Obsérvese que para el método EM es del 57 % y para el método 2 es del 62 %, no se tienen una gran diferencia en la probabilidad, pero, como resultado de la comparación K-medias tiene una mejor probabilidad de agrupamiento.

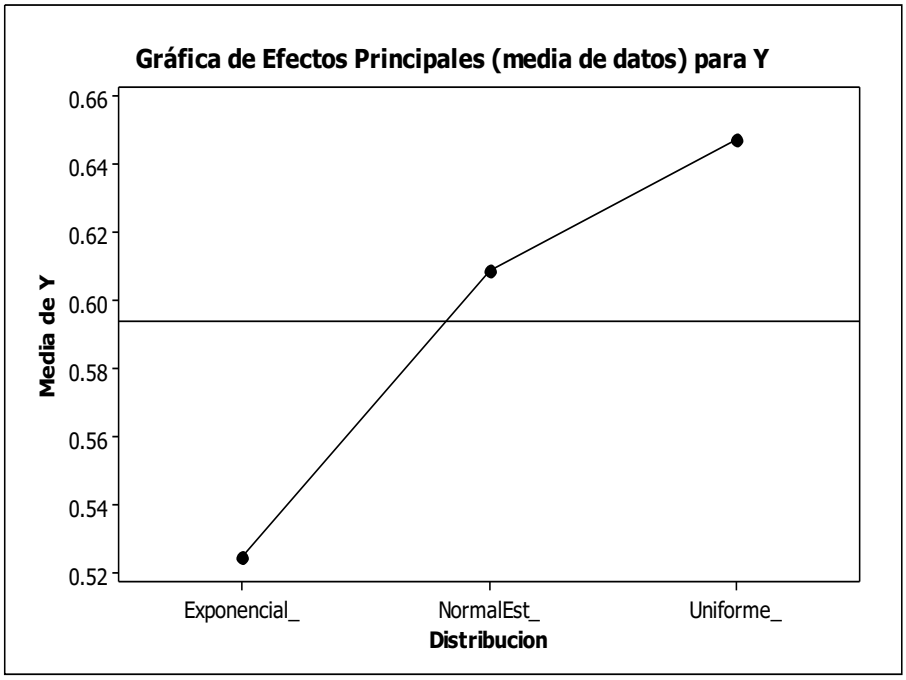

Figura 3.11: Comparación de las tres distribuciones.

La figura 3.11 muestra que el tipo de distribución tienen efecto sobre las probabilidades de los cluster. Obsérvese que la distribución Exponencial se comporta diferente a las distribuciones normal y uniforme. La distribución exponencial tiene una probabilidad del 52 %, la normal tiene una probabilidad del 62 % y la uniforme tienen la mayor probabilidad de 65 %, en consecuencia se puede decir, que la normal y la uniforme presentan mayor probabilidad de agrupamiento. Este resultado concuerda con lo esperado y puede concluirse que la probabilidad promedio de los cluster es la mas alta para las distribuciones normal y uniforme.

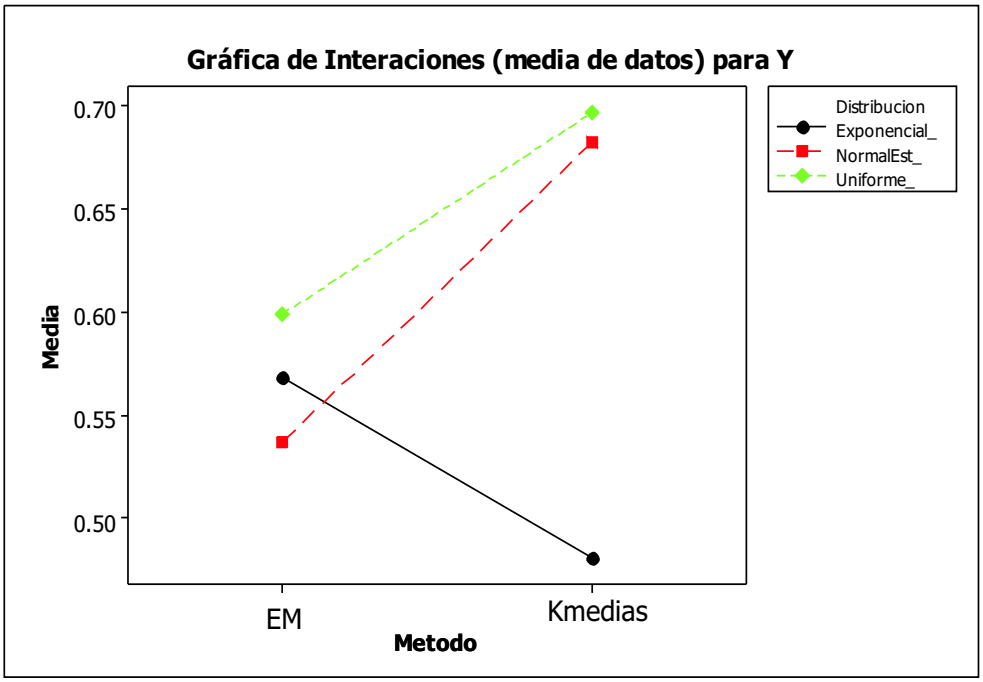

Figura 3.12: Comparación de las tres distribuciones junto con los métodos.

Al comparar simultáneamente las distribuciones y los métodos también se encontraron diferencias significativas. La gráfica 3.12, hacen evidente estas diferencias entre las medias.

Como primera observación a este gráfico, el método EM y el K-medias en cierto punto tienen la misma probabilidad, para lo cual se puede deducir que pueden dar un mismo resultado con respecto a la distribución normal y exponencial. El método EM a diferencia de lo que se pensaría con datos de distribución normal tiene menor probabilidad de agrupamiento, pero, al contrario con datos que tienden a una uniforme, pues tiene mayor probabilidad de agrupamiento; el método K-medias tiene una mala probabilidad con respecto a datos con distribución exponencial, pues es menor de 50 %, pero, funciona mejor con respecto a la uniforme, por tal motivo podemos llegar a la conclusión del anova que si hay diferencia significativa entre método y distribución, en el contexto usado en esta tesis.

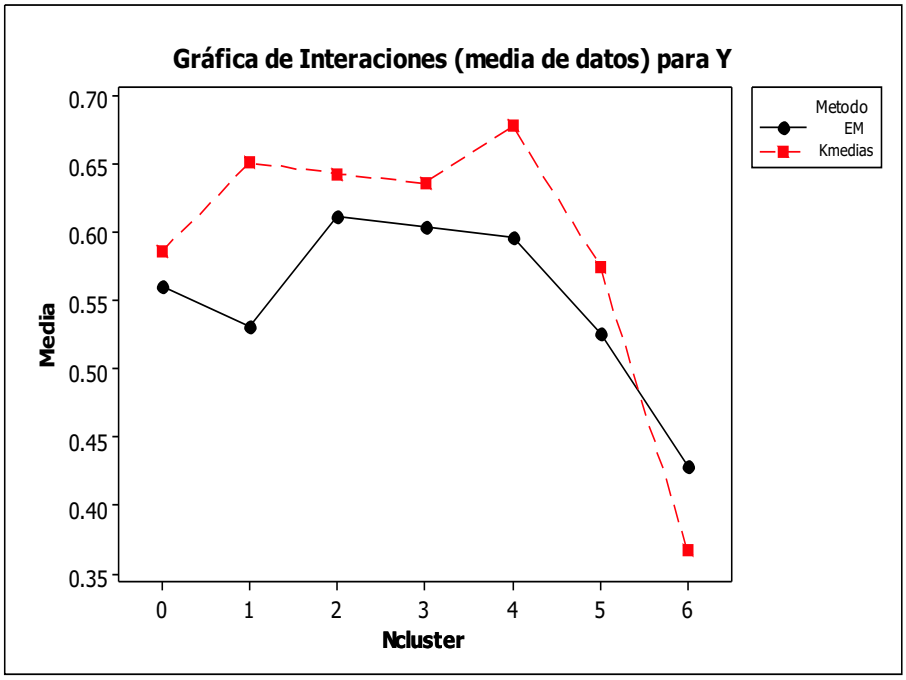

Figura 3.13: Interacción entre método y número de cluster.

En esta gráfica, no se refiere al tamaño del cluster, pues por indicio al menos un cluster debe haber. Como se observa en el capítulo 3, se habla del número de cluster que van de 3 a 7, por tanto se entiende que Ncluster hace referencia al "cluster0", "cluster1",$\cdots$,"cluster6", habiendo hecho esta aclaración, podemos entender que la mejor probabilidad esta en el cluster 4, ciertamente buena, pues alcanza 58 % para el EM y para el Kmedias 68 %, a diferencia del cluster 6 con una probabilidad muy baja que va del 42 % al 36 % para EM y K-medias respectivamente.

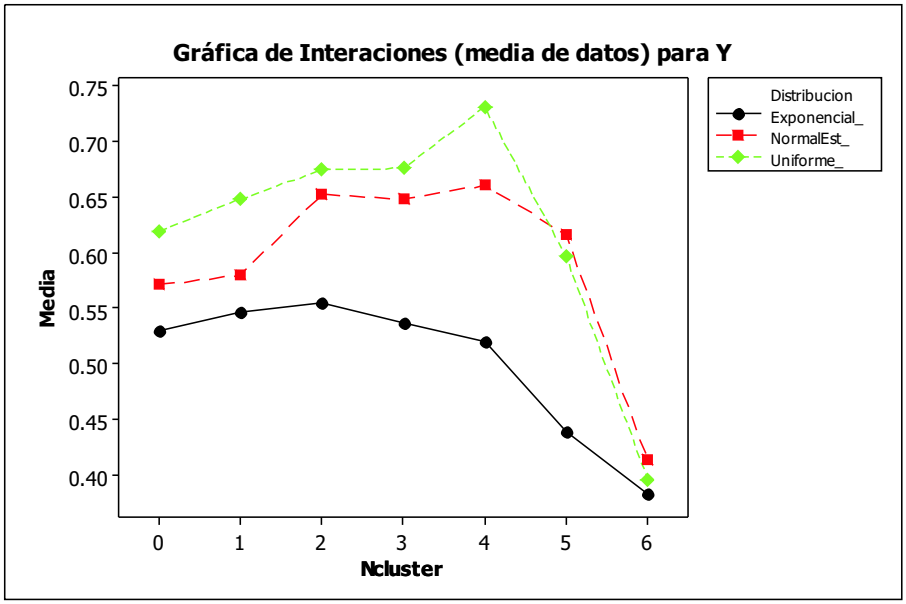

Figura 3.14: Comparación de los números de cluster y su interacción con las distribuciones.

He igualmente, como se ve en la tendencia, la uniforme tiene la mejor probabilidad en el cluster 4, pero para el cluster 6 cae aún 39 %.

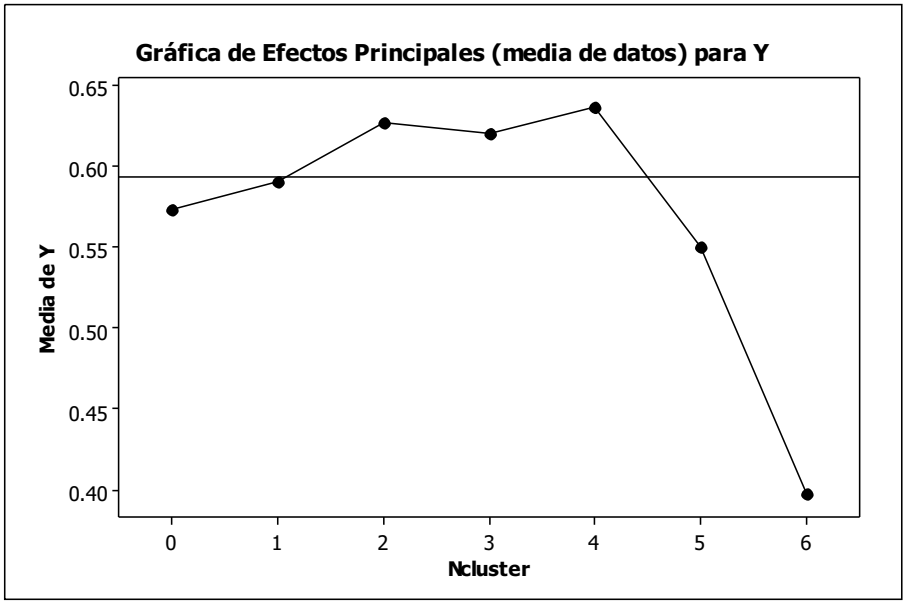

Figura 3.15: Comparación de los números de cluster.

Se muestra que los cluster 2,3 y 4 pasan la medía del conjunto de cluster. Se puede apreciar también como se ve en la figura 3.14 las tres distribuciones tienen el mejor agrupamiento en el cluster 4.

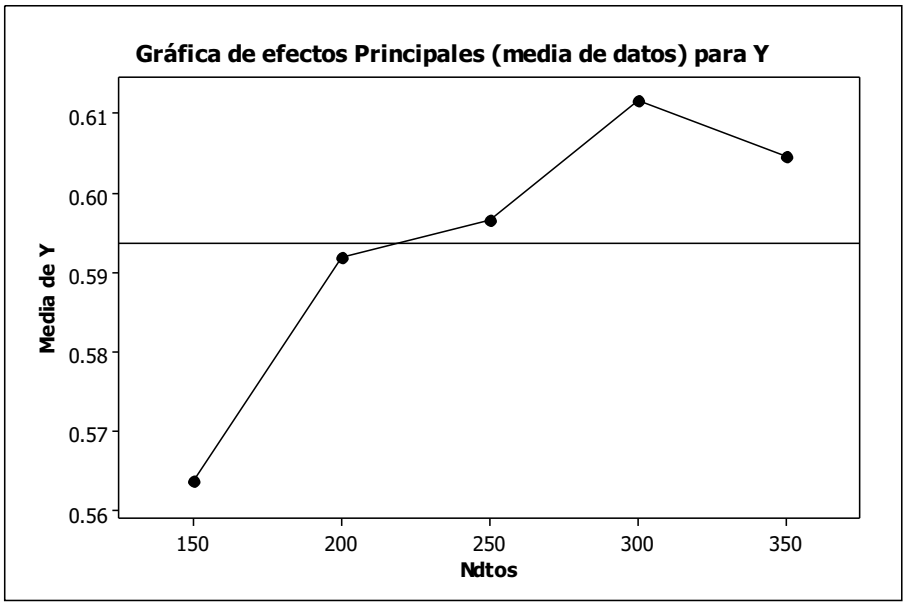

Figura 3.16: Tendencia del número de datos que van de 150 a 350.

Una aseveración es que entre mayor número de datos una mejor probabilidad, pero en este caso obtuvimos que el número óptimo es de 300 datos, pero el valor medio se encuentra entre 200 y 250 datos. Claro esto es conforme a la interacción con las variables Distribucion, Metodo, Ncluster, Tamaño y principalmente de la variable respuesta.

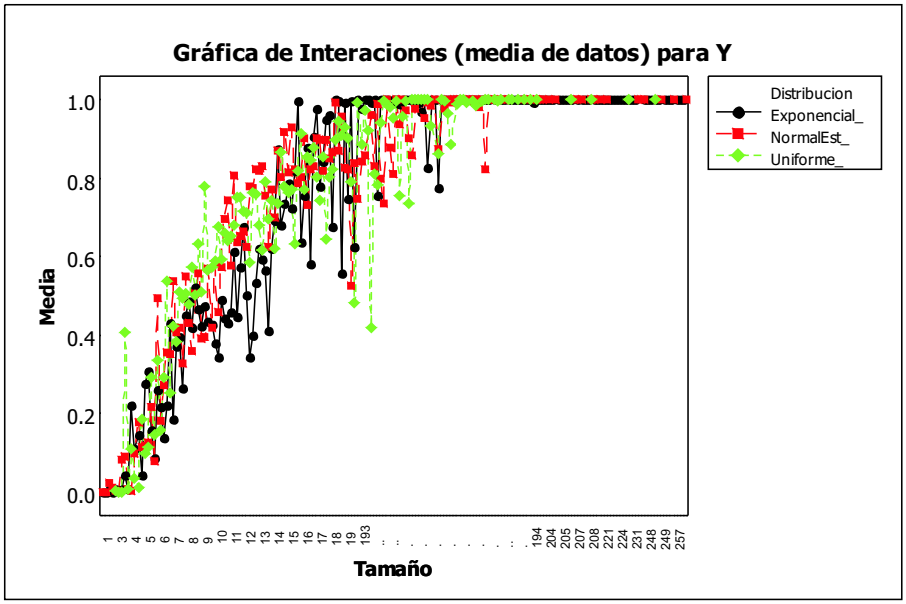

Figura 3.17: El tamaño de los cluster en interacción con las distribuciones.

En este caso nuestra afirmación es que cuanto más pequeño es el tamaño del cluster tienen una menor probabilidad, pero entre más grande el cluster tienden hasta probabilidad 1, no importando que tipo de distribución tengan los datos.

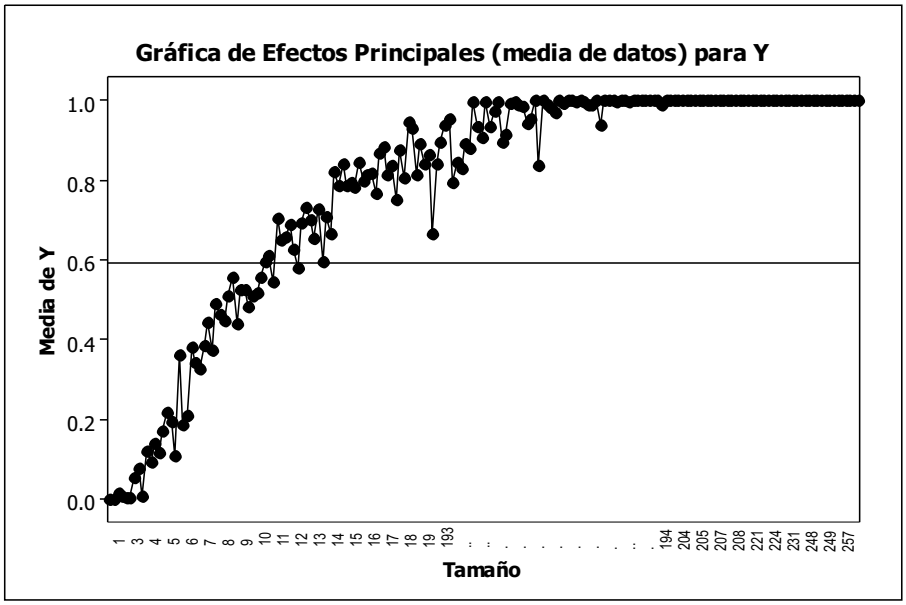

Figura 3.18: Comparación de los tamaños de cluster con respecto de Y.

Podemos observar que hay una gran variabilidad de los cluster, en función del tamaño del cluster, pero, la media es buena pues esta por el 60 % de probabilidad.

A continuación podemos ver la gráfica de barras, y la tendencia que tienen los métodos.

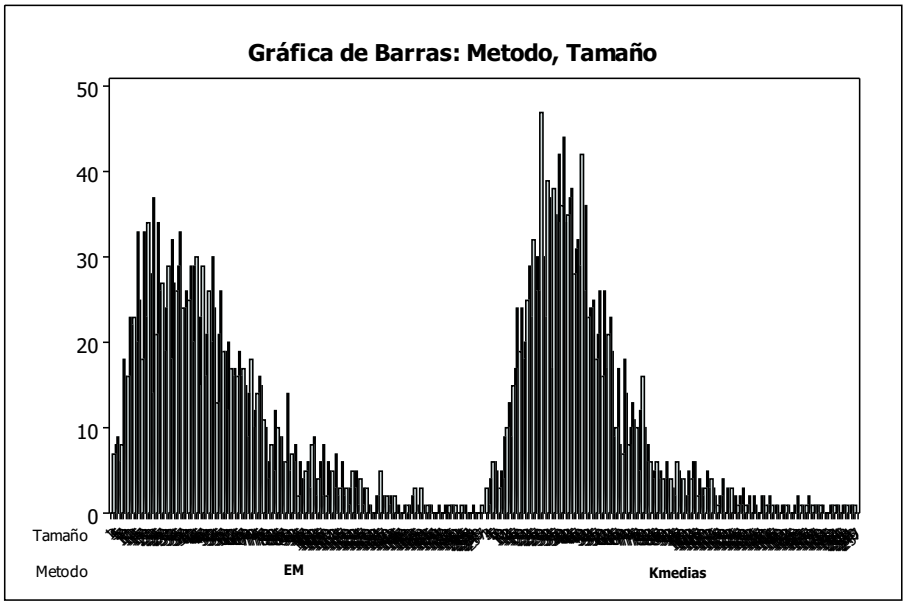

Figura 3.19: Interacción del gráfico de barras con la variable Tamaño y Metodo.

Observamos esta interacción con la tendencia del tamaño, que a su vez se compara al mismo tiempo con el método EM y el método K-medias. Y para finalizar podemos concluir que el método K-medias tiene un mejor comportamiento, con respecto al ambiente en donde se comparó.

Capítulo 4

Conclusiones

El trabajo que motivó a esta tesis es la metodología que maneja (Von A. y Mair P., 2008); el trabajo que se presenta, se realiza todo en R y RWeka. Se programó cada sección del proceso de minería de datos. La tecnología sobre RWeka, es casi nueva pues la librería se desarrolló en el 2007 y en el 2008 fue puesta en el cran de R, por tal motivo no hay trabajos de referencia, solo los expuestos por los desarrolladores de las librerías.

Se obtuvo una herramienta, que es un programa en R, que lleva a cabo la metodología propuesta por (Von A. y Mair P., 2008), para la comparación de métodos de agrupamiento, de diferente índole.

Es imprescindible hacer notar la importancia del trabajo, como una aportación a los usuarios de dos sistemas independientes, que son: WEKA y R. Ya que en su mayoría son usuarios, que no tienen una formación de programadores y por tanto es más difícil entender librerías sin un manual. Así que, este trabajo puede servir de manual para RWeka, una interfaz casi nueva.

Los resultados más relevantes obtenidos de la discusión con respecto a la simulación realizada son:

Primero Los métodos EM y k-medias se comportan en ciertos puntos de la misma forma, pero la conclusión del anova y las gráficas es irrefutable. Si hay diferencias significativas entre la tendencias del dato(Normal, Exponencial y uniforme) y el método, claro es, con respecto al contexto de este trabajo.

Segundo Los métodos comparados (EM y K-medias) parecen identificar cluster fundamentalmente en las distribuciones uniforme.

Tercero El tamaño de cada cluster tiene un efecto importante para la probabilidad de ser seleccionado ante un agrupamiento de mayor dimensión.

Y Finalmente los resultados obtenidos en la simulación brindan a los usuarios de los métodos de agrupamiento, una base para tomar decisiones sobre la existencia de clusters en las poblaciones bajo supuestos parecidos.

Referencias

Arriaza A., Fernández F., López M., Muñoz M., Pérez S. y Sánchez A. (2008). *Estadística básica con R y R-commander*. publicaciones de la universidad de Cádiz.

Bishop M. (2006). *Pattern Recognition and Machine Learning*. edit. Springer.

Bramer M. (2007). *Principles of Data Mining*. edit. Springer.

Carey V. (2007). *Arji: Another R-Java interface. R package versión 0.3.16.*.

Carlos J. (2003). *Introducción a WEKA*. Departamento de Informática Universidad de Valladolid.

Carmona L. (2006). *Minería de Datos usando SAS Enterprise Miner; una aplicación en datos forestales*. Proyecto Fin de Carrera, Colegio de Posgraduados,Texcoco, Estado de México.

Christen P. (2005). *A very short introduction to Data Mining*. Department of Computer Science. FEIT Australian National University, Diciembre.

Dapozo G., Porcel E., López M., Bogado V. y Bargiela R. (2007). *Aplicación de minería de datos con una herramienta de software libre en la evaluación del rendimiento académico de los alumnos de la carrera de Sistemas de la FACENA-UNNE*. Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura.

Delgadillo G., Ochoa A. y Muñoz J. (2006). Encuentro de investigación en ingeniería eléctrica. Una Aproximación a la Minería de Datos Laboral.

Docío L. y García C. (2005). Segmentación de locutor, detección y seguimiento de "presentadores habituales.[en] noticiarios de tv. III Congreso de la Sociedad,Española de Acústica Forense.

Fayyad U., Piatetsky-Shapiro G. y Smyth P. (1996). *From Data Mining to Knowledge Discovery in Data Base*. American Association for Artificial Intelligence.

Fisher D. (1987). *Knowledge Acquisition Via Incremental Conceptual Clustering*. Kluwer Academic Publishers, Boston - Manufactured in The Netherlands Irvine Computational Intelligence Project, Department of Information and Computer Science, University of California, Irvine, California 92717, U.S.A.

Gamboa A. (2009). *Distribuciones Continuas*. Universidad del Valle de México.

Giudici P. (2003). *Applied Data Mining statistical methods for Business and Industry*. edit. Wiley.

Hair J., Anderson R., Tatham R. y Black W. (1999). *Analisis Multivariante*. 5ta edit. Prentice Hall Iberia, Madrid.

Han J. y Kamber M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2nd edition.

Hand D., Mannila H. y Smyth P. (2001). *Principles of Data Mining, Editorial the MIT Press.*. Editorial the MIT Press.

Hartigan J. y Wong M. (1979). *Algorithm AS139: A k-means clustering algorithm*. Applied Statistics, Vol. 28.

Hernández J. (2003). Análisis y extracción de conocimiento en sistemas de información: Datawarehouse y datamining. Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia. Curso de Maestría, impartido en octubre del mismo año.

Hornik Kurt, Christian Buchta y Achim Zeileis (2008). *Open-Source Machine Learning: R Meets Weka, Wirtschaftsuniversität Wien*. Copyright © 2008 Springer-Verlag.

Infante S. y Zárate G. (1990). *Métodos Estadísticos: un enfoque interdisciplinario*. 2º edición. Editorial Trillas.

Jackson J. (2002). *Data Mining: A Conceptual Overview*. Communications of the Association for Information Systems Vol. 8.

John W. y John W. Jr. (2007). *Clustering Algorithms*. Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.3, No. 1, ISSN: 1885-4486 © ATI, 2007 22.

Kantardzic M. (2003). *Data Mining Concepts, Models, Methods, and Algorithms*. IEEE Press wiley- interscience Editorial.

Lang D. y Chambers J. (2005). *SJava, The Omegahat Interface for R and Java*. R package version 0.69-0.

Lehmer D. (1949). *Matematical methods in large-scale computing units*. Annals Computer Laboratory Harvard University, XXVI.

Lorrio A. (2009). *Clasificación automática de formas cerámicas completas: un estudio comparativo de diversos métodos multivariantes*. Departamento de Prehistoria, Universidad Complutense, Madrid.

Marín J. (2008). *Análisis de Clúster y Multidimensional Scaling*. Universidad de Carlos III de Madrid, departamento de Estadística.

Martínez J. (2001). *Herramientas para la Estructuración Conceptual, Computación y sistemas*. México, Vol. 4 N°. 3.

Maya C. (2001). *Desarrollo de una Algoritmo para la Caracterización y Clasificación de granos de café empleando Técnicas de Visión Artificial*. tesis Ing. Electrónica UNC.

Medina J. (2000). *2do Taller Internacional de Minería de Datos*. Colegio de Postgraduados,Septiembre del mismo año.

Mendenhall D. y Scheaffer W. (1986). *Estadística Matemática con Aplicaciones*. grupo editorial Iberoamericana.

Meneses A., Garrido S., Bazán D. y Florentino M. (2004). Seminario cocoa, programación orientada a objetos. Centro de Investigación y Estudios Avanzados (CINVESTAV).México, D.F.

Montes N. (2001). *Desarrollo de Algoritmos de Segmentación de frutos maduros y verdes de café en imágenes tomadas en condiciones controladas, basados en las propiedades de color*. Universidad Nacional de Colombia Facultad de Ingeniería y Arquitectura Manizales.

Morales E. (2005). curso de kdd, ciencias computacionales. Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE).

Pascual D., Pla F. y Sánchez S. (2007). *Algoritmos de Agrupamiento*. Departamento de Computación Universidad de Oriente & Departamento de Lenguajes y Sistemas Informáticos Universidad Jaume I.

Piatestsky y Shapiro G. (2006). *KDnuggets: Data Mining, Knowledge Discovery, Text Mining Web Mining,*. Springer Science + Business Media B.V., 2006. ISSN: 1573-756X.

Reese S. (1996). Searching for the mother lode: tales of the first data miner. Intelligent System & their Applications. *IEEE Expert, Vol. 11 Nº. 5.*.

Reyes H. (2008). *Estimación de Tendencias en Niveles Máximos de Contaminación usando Regresión por Cuantiles ajustando el efecto por variables meteorológicas*. tesis de Doctorado C. P., Estado de México.

Ross S. (2000). *Introduction to probability and statistics for engineers and scientists*. Academic press.

Rui X. y Donald W. II (2005). Survey of Clustering Algorithms. *IEEE. Transactions on Neural Networks, Vol. 16, No. 3*.

Schauerhuber M., Zeileis A., Meyer D. y Hornik K. (2007). Benchmarking open-source tree learners in r/rweka. in data analysis, machine learning, and applications. (Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation e.., March 7-9, Freiburg), Forthcoming.

Serrano J. (2009). *Tema3: Generación de números aleatorios; asignatura: Computación Estadística*. Departamento de Informática de la Universidad de Jaén.

Sevillano G. (2005). *Circuitos Digitales basados en FPGAs para Generación de Números Aleatorio*. s, Tutores, Inés del Campo Hagelstrom y Javier Echanobe Arias.

Sáez G. (2000). El azar de la criptografía. *artículo publicado en Criptomicón; servicio web ofrecido por el Instituto de Física Aplicada del CSIC*.

Tarifa E. (2003). *Teoría de modelos y Simulación*. Facultad de ingeniería, Universidad de Jujuy.

Thomson W. (1958). *A modified congruence method of generating pseudo-random numbers*.

Triola M. (1999). *Estadística Elemental*. capitulo 11, edit. Pearson.

Two Crows Corporation (1999). *Introduction to Data Mining and Knowledge Discovery*. Third Edition, Web: www.twocrows.com.

Urbanek S. (2007). *RJava: Low-Level R to Java Interface*. R package version 0.4-16.

versión 3.5.8, W. (2008).

Von A. y Mair P. (2008). *Evaluating cluster solutions with reference to data generation processes: A simulation study*. Memorias del XXII Foro Nacional de Estadística, Aguascalientes, México: Instituto Nacional de Estadística, Geografía e Informática.

Vásquez B. (2009). *Estudio comparativo de Métodos de clasificación Automática en la Zonificación Agro Ecológica del sur del estado de Puebla*. Proyecto Fin de Carrera, Facultad de Ciencias de la Compuación - BUAP., Ciudad de Puebla.

Weinner R. (2001). *Estadística, Capitulo 13*. Edit. Continental.

Zhengxin C. (2001). *Data mining and uncertain reasoning: an integrated approach*. New York. Edit. Wiley.

Anexo A

Algoritmo Jerárquico

Ejemplo de uso de un algoritmo Aglomerativo Jerárquico:
Supongamos 6 variables

	1	2	3	4	5	6
1	0					
2	10	0				
3	9	6	0			
4	8	7	3	0		
5	4	16	11	9	0	
6	10	15	21	17	25	0

$d_{4,3} = 3$

$d_{3\ 4,1}\{9, 8\} = 8$

$d_{1\ 5,3\ 4}\{8, 9\} = 8$

$d_{3\ 4,2}\{6, 7\} = 6$

$d_{3\ 4,5}\{11, 9\} = 9$

	3 4	1	2	5	6
3 4					
1	8	0			
2	6	10	0		
5	9	4	16	0	
6	17	10	15	25	0

$d_{5,1} = 4$

$d_{15,2}\{10, 16\} = 10$

$d_{15,6}\{10, 25\} = 10$

$d_{3\ 4,5}\{11, 9\} = 9$

	1 5	3 5	2	6
1 5	0			
3 4	8	0		
2	10	6	0	
6	10	17	15	0

$d_{2,3\ 4} = 6$

$d_{3\ 4\ 2,15}\{10, 8\} = 8$

$d_{3\ 4\ 2,6}\{15, 17\} = 15$

	3 4 2	1 5	6
3 4 2	0		
3 4	8	0	
6	15	10	0

$d_{1\ 5,3\ 4\ 2} = 8$

	3 4 2 1 5	6
3 4 2 1 5	0	
6	10	0

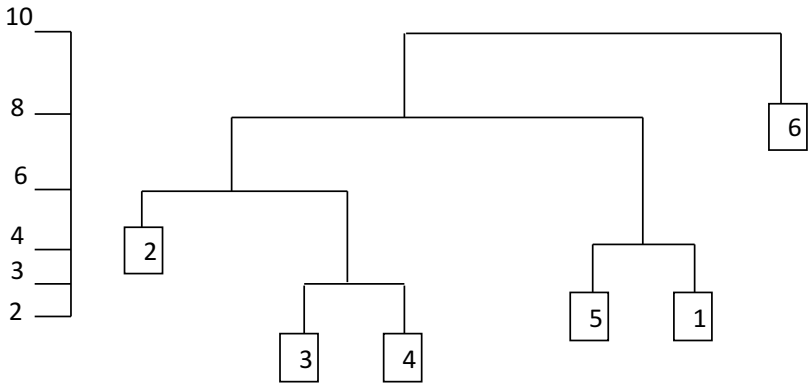

Figura A.1: Vista del diagrama, una corrida a papel y lápiz.

Anexo B

Código Completo, para comparar métodos de agrupamiento no supervisado(clustering)

```
library(RWeka)
#######################################################################
a<-("E:/tesis/tesis_santiago/Capitulo_3/Experimento_2/NormalEst_")
c<-("_")
e<-(".arff")
a3<-("E:/tesis/tesis_santiago/Capitulo_3/Experimento_2/Exponencial_")
a2<-("E:/tesis/tesis_santiago/Capitulo_3/Experimento_2/Uniforme_")
#######################################################################
valor<-150
y<-3
for(k in 1:5) {#for1---cantidades de 150 a 350
#################print(c("Variables con valor=",valor))
    for(i in 1:5) {#for2-----variables que van de 3 a 7
        ##########print(c("Archivo con variable =",i+2))
        b<-(as.character(i+2))
        d<-(as.character(valor))
        M<- matrix ( ,valor,y)
        M2<- matrix ( ,valor,y)
        M3<- matrix ( ,valor,y)
        media<-0
        for(j in 1:y) {#for3
            M[ ,j]<-rnorm(valor,media)
            M2[ ,j]<-runif(valor)
            M3[ ,j]<-log(M2[ ,j])
```

```
            write.arff(data.frame(M),file=paste(a[1],b[1],c[1],d[1],e[1],sep=""))
            write.arff(data.frame(M2),file=paste(a2[1],b[1],c[1],d[1],e[1],sep=""))
            write.arff(data.frame(M3),file=paste(a3[1],b[1],c[1],d[1],e[1],sep=""))
            media<-media+3
            }#for3
        y<-y+1
        }#for2
valor<-(150+(k*50))
    y<-3
}#for1
ind<-1  #indice para el numero de datos 150 a 350
VectorEM    <-NULL
VectorKm    <-NULL
VectorNvar  <-NULL
VectorNdatos<-NULL
VectorDist  <-NULL
VectorMtdo  <-NULL
VectorNCtr  <-NULL
i<-1
kk3i<-3
#Cobw <- make_Weka_clusterer("weka/clusterers/Cobweb", "weka_cluster")
EM   <- make_Weka_clusterer("weka/clusterers/EM", "weka_cluster")
KMds <- make_Weka_clusterer("weka/clusterers/SimpleKMeans", "weka_cluster")
ruta<-"C:/Documents and Settings/C&Bmini/Escritorio/tesis
        /tesis_santiago/Capitulo_3/Experimento_2/"
bisNum<-c("Exponencial_","NormalEst_","Uniforme_
            ","150","200","250","300","350")
for(f in 1:3){
for(kk in 1:5){
    for(kk2 in ind:5){  #distri   #numero de variables    #cantidad de datos
        dv<-paste(ruta,bisNum[f],as.character(kk+2),"_",as.character(bisNum[
                    ind+3]),".arff",sep="")
        ll<-read.arff(file=dv)
        for(kk3 in 3:7){
        temp  <-   EM(ll,Weka_control(N = kk3i))
        temp2 <- KMds(ll,Weka_control(N = kk3i))
        VectorResEM <- table(predict(temp ))
        VectorResKm <- table(predict(temp2))
        VectorEM     <- cbind(c(VectorEM,VectorResEM)) #Tamaño
        VectorKm     <- cbind(c(VectorKm,VectorResKm)) #"   "
        VectorNdatos <- cbind(c(VectorNdatos,as.integer(rep(bisNum[
                        ind+3], times=kk3i))))#N° Datos
        VectorNvar   <- cbind(c(VectorNvar,rep(as.integer(kk+2),times=kk3i)))#N° Var
        VectorDist   <- cbind(c(VectorDist,rep(bisNum[f], times=kk3i)))
        VectorNCtr   <- cbind(c(VectorNCtr,(0:(kk3i-1)))) #estaal
        kk3i<-kk3i+1;
        }
```

```
            kk3i<-3; ind<-ind+1; i<-i+1
        }
    ind<-1
}
}
VectorMtdo   <- cbind(c(rep(as.integer(1),times=length(VectorEM)
                  ),rep(as.integer(2),times=length(VectorKm))))#Metodo
Metodo       <- factor(VectorMtdo)
Distribucion <- factor(c(VectorDist,VectorDist))
VectorEMKm<-c(VectorEM,VectorKm)
indice  <-1
indice2 <-3
conta   <-4
Vresp   <-NULL
leng    <-length(VectorEMKm)
while(indice $<$leng){
        if(conta==7)
        conta  <-3
        datoss <-VectorEMKm[indice:indice]
        propp  <-VectorNCtr[indice:indice2]
        propp  <-propp/10
        Vres   <-pbinom(datoss, size=sum(datoss), prob=propp)
        Vresp  <-cbind(Vresp,Vres)
        indice <-indice2+1
        indice2<-indice2+conta
        conta  <-conta+1
}
VectorNdatos<-c(VectorNdatos,VectorNdatos)
VectorNvar  <-c(VectorNvar,VectorNvar)
VectorDist  <-c(VectorDist,VectorDist)
VectorNCtr  <-c(VectorNCtr,VectorNCtr)
datos <- data.frame(Distribucion=Distribucion,
                    Metodo=Metodo, Ndtos=VectorNdatos,Nvar=
                    VectorNvar,Ncluster=VectorNCtr,Tamaño=as.vector(
                    VectorEMKm), Y=as.vector(Vresp))
modelo=lm(Y~Distribucion*Metodo* Ndtos*Nvar*Ncluster*Tamaño,data=datos)
Ruselt<-anova(modelo)
Ruselt
```