Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

Determinación de Umbrales en Arboles Tipo CF Utilizando Aprendizaje Supervisado para Algoritmos de Clustering Jerárquico

Tesis que presenta:

Mayra Teresa Trejo Hernández

para obtener el grado de:

Maestro en Ciencias

en la especialidad de:

Ingeniería Eléctrica

Directores de Tesis

Dr. Mario Angel Siller González Pico
Dr. Ricardo Vilalta

Guadalajara, Jalisco, Agosto de 2009.

Determinación de Umbrales en Arboles Tipo CF Utilizando Aprendizaje Supervisado para Algoritmos de Clustering Jerárquico

Tesis de Maestría en Ciencias
Ingeniería Eléctrica

Por:
Mayra Teresa Trejo Hernández
Licenciada en Ciencias de la Computación
Universidad Autónoma de Yucatán 2002-2007

Becario de CONACYT, expediente no. 212734

Directores de Tesis
Dr. Mario Angel Siller González Pico
Dr. Ricardo Vilalta

CINVESTAV del IPN Unidad Guadalajara, Agosto de 2009.

Agradecimientos

A Dios por haberme guiado y cuidado hasta el día de hoy.

A mis padres Raúl N. Trejo Gómez y Guadalupe Hernández Tolentino por su amor, comprensión, paciencia y por creer en mi.

A mis hermanos Julio César y Raúl Ricardo por la confianza que depositan en mí y por inspirarme a ser mejor cada día.

A mis asesores Dr. Ricardo Vilalta López y Dr. Mario Ángel Siller González Pico por apoyarme con sus conocimientos y experiencias.

A los profesores Dr. Heinz Andernach y Dr. Cesar Caretta por sus aportes con respecto a los datos e información relacionada con el caso de estudio.

A mis compañeros, por su amistad y los momentos que compartimos a lo largo de la maestría.

Al conacyt por la beca.

Resumen

La importancia de la Minería de Datos en la actualidad se debe al desarrollo de las tecnologías de la información sobre diversas áreas del conocimiento, al crecimiento de la información como consecuencia de este desarrollo, a la necesidad del análisis y procesamiento de la información.

El objetivo de esta tesis es realizar un estudio sobre los métodos de Minería de Datos para encontrar la manera mas adecuada de analizar grandes bases de datos de cúmulos de galaxias conformadas de datos espaciales y correlacionados. En este trabajo se plantea la utilización del algoritmo BIRCH para el clustering de datos de cúmulos de galaxias, procurando mejorar los tiempos de ejecución.

Estos objetivos fueron alcanzados con la reducción del número de iteraciones demandadas por las reconstrucciones del algoritmo a partir de la elección de una nueva técnica aprendizaje supervisado la cual determina el umbral requerido por los árboles CF.

Como producto de esta investigación se realizó la implementación del algoritmo de clustering *BIRCH* en C++ bajo la plataforma de Linux, el análisis del desempeño de métodos de clasificación aplicados a nuestro problema; utilizando la herramienta de Minería de Datos Weka para realizar dicho análisis, en el cual seleccionamos el algoritmo de clasificación *K-nearest neighbours* ya que presentó resultados durante la comparación de ejecuciones que resultaron superiores a los demás métodos.

Como conclusiones presentamos una manera apropiada para la obtención del valor de umbral. logrando así la reducción hasta en un 100 % la reconstrucción del árbol en bases de datos similares a las conocidas previamente y por lo tanto mejoras significativas en los tiempos de ejecución.

Abstract

The popularity of Data Mining nowadays is due to the advanced development of information technologies on diverse areas of knowledge, the growth of the information as a result of this development, and the need for analysis and information processing.

The objective of this thesis is to carry out a study to find the best way to analyze large databases of galaxy clusters composed of spatial and correlated data. In this work we use the algorithm BIRCH for the clustering of galaxies, trying to improve on execution times.

This objective is achieved by reducing the number of iterations required by the algorithm by automating the choice of a new supervised learning technique which determines the threshold required by the CF tree.

The result of this research is the implementation of the BIRCH clustering algorithm in C ++ under the Linux platform, the analysis of the performance of classification methods applied to our problem, using the Weka Data Mining to perform this analysis (we used the algorithm K-nearest neighbors as the baseline algorithm).

To conclude we present an appropriate way for obtaining the threshold value, thus achieving a reduction of up to 100 % in the reconstruction of the tree in databases similar to those previously known and therefore significant improvements in execution times.

Índice general

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

El creciente desarrollo de la tecnología de la información, en áreas del conocimiento como: matemáticas, geografía y astronomía, entre muchas otras y tiene como consecuencia un crecimiento en la cantidad de información que cada una de estas áreas requiere. Es debido a esto que métodos para el manejo, análisis y procesamiento de la información están siendo requeridos.

Existen métodos convencionales utilizados por los astrofísicos para el manejo e interpretación de esta información, pero los algoritmos existentes resultan inadecuados debido a: el tamaño de las bases de datos, lo lento y generalidad de estos métodos. Es de estas inadecuaciones de donde nace la necesidad de dirigir la Minería de Datos hacia esta área con el objeto de maximizar el entendimiento de los datos.

Numerosas bases de datos de estrellas, galaxias, entre otras son resultado de técnicas utilizadas para observar el cielo, estas observaciones arrojan un equivalente de hasta 10^{12} bytes y los astrofísicos a menudo se enfrentan con situaciones en las que requieren de analizar esta información la cual resulta difícil de interpretar simplemente mediante la observación.

1.1. Antecedentes

Dirigiremos el trabajo al caso de estudio de Clasificación de Cúmulos de Galaxias y lo resolveremos mediante la aplicación de técnicas de Minería de Datos orientada al caso de estudio elegido, esto incluye la identificación de una metodología adecuada para el tipo y tamaño de los datos, así como la implementación computacional que permita el análisis de la aplicación de dicha metodología, se estudiarán elementos para la mejora del desempeño.

1.2. Motivación

La extracción de características importantes así como la interpretación de la información son aspectos importantes debido al gran tamaño de las actuales bases de datos pero el análisis de estas bases de datos de manera manual resulta ser un proceso lento y costoso, por lo que se requiere de métodos probados y específicos para llevar a cabo esta tarea.

La *Minería de Datos* es un proceso de análisis con el propósito de la identificación y extracción de patrones interesantes, relevantes con el propósito de encontrar rasgos previamente desconocidos contenidos de manera no obvia dentro de bases de datos con grandes volumenes de información. Permitiendo la interpretación y el entendimiento de las relaciones existentes entre los datos expresandolas de maneras que permitan un mayor y mejor entendimiento donde este resultado es de importancia para el poseedor de la información, brindando así la oportunidad de descubrir nuevo conocimiento.

Los parámetros utilizados para medir las similitudes son los *Atributos* y la representación producida son las relaciones encontradas son los denominados *Patrones*. De esta manera cada clase de objetos queda representado por un vector de atributos similares para cada clase. Un Patrón explica un diseño general, las relaciones existentes que son semejantes y que describen el modelo, es un tipo recurrente de eventos.

1.3. Objetivos

Los objetivos de esta tesis son:

Análisis del problema de clustering de grandes cantidades de datos (en nuestra aplicación trabajaremos con datos referentes a atributos de galaxias y serán descritos en secciones posteriores).

Elección de subárea de la Minería de Datos adecuada al problema y de un algoritmo dentro de esta área el cual defina el tipo adecuado de solución. Preprocesamiento de los datos para una correcta utilización de ellos dentro de la metodología.

Implementación de la solución planteada.

Realización de la propuesta que mejore los tiempos de ejecución de este algoritmo.

Estudio de los resultados obtenidos.

1.4. Estructura de la tesis

El resto de la tesis está organizada como sigue:

El capítulo 2 describe el marco de la investigación para el diseño metodológico, se encuentra dividido en dos secciones: en la primera se encontrarán los fundamentos, es decir un marco de antecedentes entre los cuales se presentan: definiciones, consideraciones, conceptos, etc. la segunda sección contiene el conjunto de trabajos hechos por otros investigadores en el área.

En el capítulo 3 se revisa un método denominado BIRCH sobre el cual basaremos nuestro análisis, lo estudiaremos, deduciremos la importancia de nuestro trabajo y definiremos la propuesta.

En el capítulo 4 se presenta el desarrollo e implementación de la metodología y el estudio de los experimentos.

En el capítulo 5 finalmente se presentan las conclusiones y el trabajo futuro.

Capítulo 2

Marco de Investigación

En este capítulo presentamos la serie de elementos denominados *conocimientos* considerados indispensables para el sustento del trabajo de investigación realizado, comprende dos secciónes, la primera *Marco Teórico* en la cual se describe el conocimiento que nos permitirá llevar a cabo los objetivos determinados y la segunda parte *Estado del Arte* en la cual evidenciaremos las relaciones existentes entre el marco teórico y nuestra investigación, se enunciará la información obtenida como producto de la revision bibliográfica.

2.1. Marco Teórico

2.1.1. Minería de Datos y Reconocimiento de Patrones

El término *Inteligencia Artifical* es definido de varias maneras, en [ ] podemos encontrar un resumen de algunas de las principales definiciones en esta área, aquí mencionamos algunos de estos enfoques:

- Sistemas que piensan como humanos.
- Sistemas que piensan racionalmente.
- Sistemas que actuan como humanos.
- Sistemas que actuan racionalmente.

Entre ellas encontramos como factor común el intento de igualar los pensamientos ó actitudes del hombre por una máquina.

El reconocimiento de Objetos de manera automática es una tarea del área de Inteligencia Artificial; incluso para un niño la identificación de un rostro antes visto es una tarea fácil, la tarea difícil consiste en explicar cómo es que esto fue hecho, esta es la parte que se desea automatizar y la cual está formando uno de los grandes retos del área de reconocimiento de patrones. Este es un tema muy importante en la actualidad y como este podemos mencionar muchos otros problemas; como la caracterización, modelado o reconocimiento de objetos.

Para decir a que denominamos Reconocimiento de Patrones primero necesitamos definir a los patrones como los objetos de nuestro interés los cuales cuentan con algunas características. De manera formal un *Patrón* es una descripción estructural ó cuantitativa de un objeto ó de alguna otra entidad de interés, una clase de patrones es una familia que comparte características comunes, ahora podemos decir que *Reconocimiento de Patrones* es la capacidad de un método automatizado de poder reconocer los patrones de la misma manera ó de la manera mas similar a la manera en que lo haría un humano, reconociendo el objeto a partir de mediciones de estas características, dotando así a las computadoras de la capacidad de observar el mundo de una manera similar a la que nosotros lo hacemos.

En la Figura 2.1 podemos ver un ejemplo muy sencillo de clasificación de flores en el que tenemos 3 clases (tipos de flores) y 2 descriptores (mediciones)

$$\vec{x} = [x_1 x_1 ... x_n]^T \tag{2.1}$$

Donde:
x_i es el i-ésimo descriptor y n es el número de descriptores, en este ejemplo simple podemos ver como cada flor está representada en un espacio de 2D [ ].

La Inteligencia Artificial y muchas áreas de esta, trabajan con cualquier tipo de datos. en todas las áreas de la sociedad se producen enormes cantidades con las cuales es necesario trabajar, para esto necesitamos de técnicas que permitan un manejo de los datos realmente relevantes, ya que como veremos mas adelante mucha de la información no es relevante para determinado estudio y únicamente causa ruido, pero más importante que encontrar esas técnicas es la necesidad de la automatización de esos métodos, la determinación de los patrones dentro de la información es determinante para la interpretación de la información.

Los datos con los que se trabajan deben estar completamente definidos, por ejemplo podríamos hablar de un cúmulo como una entidad la cual cuenta con características como tamaño, número de galaxias, masa, dispersión, color, forma, velocidad, pero cuando deseemos encontrar similitudes entre ellas tal vez no necesitamos de características como el color y la forma las cuales podrían clasificar como diferentes a galaxias con similitudes en dispersión, masa y velocidad (los cuales son mejores clasificadores).

Los patrones forman parte fundamental del éxito de la clasificación, pero podemos encontrar una serie de problemas frecuentes [ ]:

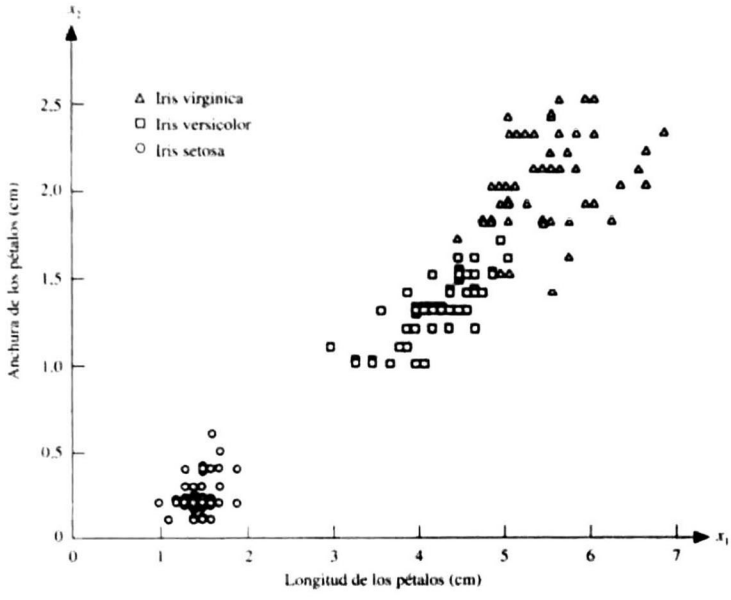

Figura 2.1: Reconocimiento de Patrones.

- Los patrones pueden ser inexactos (o falsos)
- Los descriptores pueden ser ilegibles o faltantes

Denominamos *Técnicas de Machine Learning* a todas aquellas que le permitan a la computadora aprender. es decir que permite a la computadora predecir un nuevo comportamiento a partir de la experiencia de ejemplos previamente presentados.

Estas técnicas son muy útiles en la Minería de Datos y son ejecutadas sobre los datos de muestreo. Un ejemplo de reunión de estas técnicas es el Software denominado *Weka*, el cual tiene como objetivo principal el preprocesamiento de los datos para una rápida ejecución de los algoritmos de Machine Learning.

La Minería de Datos es primordial en el proceso del Descubrimiento de la Información en Bases de Datos (Knowledge Discovery in Databases). Las fases principales que deben seguirse para llevar a cabo el proceso de Minería de Datos son las siguientes [ ] [ ] [ ]:

Selección: Eliminación de la información no relevante para el desarrollo de los experimentos ó que simplemente pueden no interesarnos para términos de la investigación.

Preprocesamiento: Una vez establecidos los atributos sobre los cuales aplicaremos la metodología, una depuración sobre la base de datos puede ser necesaria, esto puede ser alguna estandarización ó formato específico de los atributos.

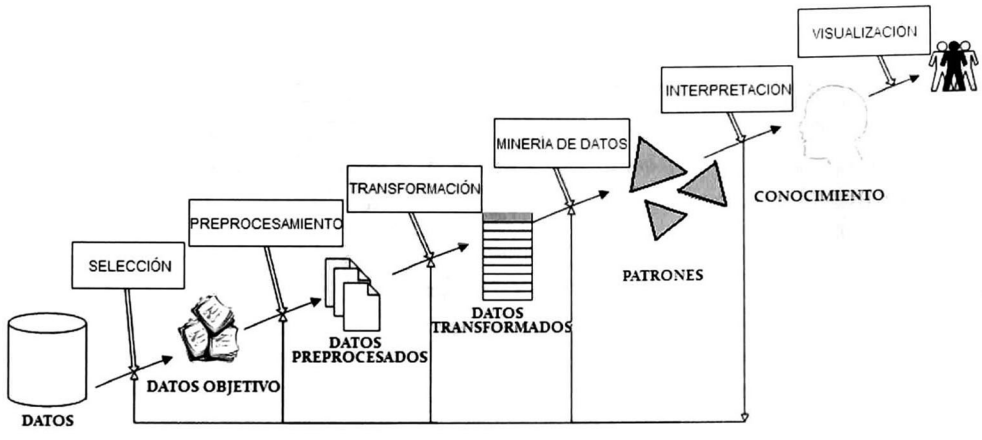

Figura 2.2: Minería de Datos.

Transformación: La base de datos es adecuada al formato requerido por el algoritmo que será ejecutado sobre ella.

Minería de Datos: Empleo de alguna técnica de Minería de Datos.

Interpretación: Análisis y explicación de los resultados derivados de la ejecución de la técnica seleccionada.

Visualización: Representaciones mediante imagenes podrían mejorar la interpretación de los resultados ó facilitar deducciones que de otra manera no saltan a la vista.

Ejemplos de técnicas de Machine Learning son los algoritmos de pre-procesamiento de datos, clasificación, regresión, clustering, reglas de asociacion y visualizacion; y ejemplos de áreas en las cuales son aplicables los métodos de Machine Learning [ ] son: el diagnóstico de fallas de máquinas, estudios de mercado, biología, química, selección automática de preferencias y astronomía, entre otras.

2.1.2. Representación del Conocimiento

Al tener tantos datos se necesita de algún tipo de representación que permita al usuario simbolizar las características de una manera que permita una rápida comprensión de la información mediante la interpretación de estructuras menos elaboradas. Cada uno de estos esquemas cuenta con sus propias ventajas y deficiencias y son específicos para resolver algún tipo de problema en especifico, es decir para obtener un mayor éxito en la utilización de alguna representación requeriremos la correcta representación dependiendo de los objetivos del

problema que deseamos resolver, un modelo de representación ó diagrama puede representar la diferencia al intentar resolver nuestro problema.

Las mas sobresalientes formas de representar la información obtenida son descritas a continuación [ ]:

Tablas de decisión Forma más simple de representar la información en la cual la representación está basada únicamente en las entradas de información; es capaz de representar las diferentes alternativas y las salidas de cada combinación de las entradas, será necesario llevar a cabo la determinación de las características más relevantes para incluir solo estas en la tabla.

Árboles de decisión Diagrama en el que las diferentes características son mostradas en el orden en el que son consideradas, habitualmente son formadas de acuerdo a una descripción del problema; la interpretación de la asignación de la clase de los elementos se va dando de acuerdo a los atributos recorridos para llegar al nodo determinado y cuando alcanzamos una hoja, esta es asignada con el valor de la clase asignada para esa hoja. Este modelo de representación se recomienda cuando el número de elementos y de atributos es pequeño; por la naturaleza de la representación los atributos deben ser discretos ó al menos deberán ser representados de esta manera.

Reglas de asociación Permiten predecir atributos partir de uno ó más combinaciones de atributos, esto con un porcentaje el cual representa el numero de ejemplos que cumple la regla denominado confianza, la cual nos permite encontrar aquellas asociaciones más probables.

Reglas con excepciones Es similar a las reglas de asociación pero permiten excepciones, logrando incorporar ejemplos necesarios pero que sabemos que no entran en el conjunto establecido por las reglas anteriores.

Reglas que implican relaciones Representación mas expresiva utilizada para reglas en las cuales los atributos se encuentran relacionados, realiza comparaciones entre los atributos.

Árboles para la predicción numérica Árbol de clasificación en el que utilizamos probabilidades para predecir casos; mediante esta representación podemos predecir atributos numéricos.

Representación basada en ejemplos Llamado aprendizaje basado en ejemplos ya que no se basa en la descripción de patrones explícitos sino en parecido con ejemplos similares que fueron clasificados anteriormente.

Cluster Esta forma de representacion consta de agrupaciones de datos de acuerdo a su similitud de datos.

2.1.3. Minería de Datos Espacial

DEFINICIÓN 1. La **Minería de Datos Espacial** es el proceso de extracción del conocimiento; el descubrimiento patrones interesantes y previamente desconocidos pero útiles en largas bases de datos que contienen datos espaciales.

La Minería de Datos Espacial demanda una integración de la Minería de Datos con las tecnologías de bases de datos espaciales; para el entendimiento de datos espaciales y el descubrimiento de relaciones entre datos espaciales y datos no espaciales [ ].

Las características que distinguen la Minería de Datos Espacial de la Minería Clásica [ ] son:

a. *Datos de Entrada:*
   -Utiliza grandes bases de datos.
   -Los datos de entrada incluyen dos tipos de atributos: Espaciales y no Espaciales.

b. *Fundamentos Estadísticos:*
   -Los datos tienden a estar altamente correlacionados.

c. *Patrones de Salida:*
   -La información de datos obtenida puede incluir datos que difieren de la mayoría.
   -Los resultados también suelen ser interpretados de manera espacial.
   -Las formas obtenidas podrían ser arbitrarias.

d. Proceso Computacional:
   -Se han creado adaptaciones de los métodos tradicionales para datos con estas características.
   -La autocorrelación espacial y la baja dimensionalidad en el espacio provee mayores oportunidades y eficiencia computacional.

Podemos clasificar la Minería Espacial en cuatro tipo principales [ ]:

I. *Modelos Predictivos:*
Predicen ubicaciones, dos técnicas representativas son: Modelo Espacial Autorregresivo SAR y Campos Aleatorios de Markov MRF

II. *Outliers Espaciales:*
Observaciones que presentan resultados desviados ó contradictorios con respecto al resto de la base de datos.

III. *Reglas de Colocacion Espacial:*
Predicen la presencia de características a partir de la aparición de otras características y con esto identifica reglas de co-ubicación.

IV *Algoritmos de Clustering:*
Agrupamiento de objetos los cuales poseen características similares y diferentes de objetos en otros grupos. Ahora los tres principales metodos de clustering son:

a) Particionales.- Divide los datos en un número determinado de grupos.

b) Jerárquicos.- Produce clusters reproduciendo la forma de árbol jerárquico definido en base a la similitud ó distancia de los objetos.

c) Basados en Densidad.- Encuentra los grupos basandose en la densidad de los objetos.

El modelo de datos representado por bases de datos de cúmulos de galaxias corresponde al término denominado Datos Espaciales.

DEFINICIÓN 2. **Datos Espaciales** son entidades los cuales los podemos describir mediante:

Una posición absoluta en un sistema de coordenadas.

Una posición relativa con respecto a los elementos entre sí.

Una representación mediante alguna figura geométrica que los represente.

Atributos los cuales describen la entidad.

2.2. Trabajos Relacionados

2.2.1. Minería de Datos

Algunas de las técnicas computacionales utilizadas para clasificación en bases de datos tenemos:

- Métodos de Agrupación.
- Métodos de Clustering.
- Métodos de Búsqueda de Información Automatizada.
- Técnicas de Visualización y Representación.

Estas técnicas permiten una vez ejecutadas llevar a cabo búsquedas parametrizadas sobre bases de datos en las cuales es posible consultar información de manera más precisa, es decir definiendo por sus parámetros más significativos.

Para hacer aplicaciones de Minería de Datos necesitamos que bases de datos se encuentren organizadas a su vez de arreglos multidimensionales, [ ] propone un método indexado multidimensional el cual utiliza un árbol R y una vez formado el árbol se utiliza un algoritmo incremental llamado `VAMSplit` para la construcción de un optimizado árbol R que conserve una proximidad espacial entre los nodos hermanos; también se estudia la manera adecuada de cómo dividir los datos para poder subir a memoria el número exacto tomando en cuenta la indexación.

De campos como el denominado *Estadística Astronómica* surgen preguntas interesantes para el área de Reconocimiento de Patrones, ejemplos de tales preguntas son: ¿cómo dividir o clasificar en subclases? ó ¿qué características son comunes entre elementos de una misma clase?, estas son algunas preguntas que resolveremos a lo largo de la investigación. Astrónomos poco a poco empiezan a aceptar la ayuda de otras áreas para una más rápida solución de problemas incluso cada vez existe un mayor número de herramientas utilizadas por astrónomos para la solución de problemas matemáticos.

Estos inicios de trabajo en conjunto han llevado a reconocer un gran potencial en agrupaciones formadas por grupos de colaboradores interdisciplinarios de grandes universidades como la Universidad de Stanford, California, Berkely entre otras y grandes centros de investigación entre los cuales podemos mencionar incluso a la NASA, toda esta investigación y colaboración está siendo llevada a cabo en Estados Unidos y Europa principalmente.

Regresando a las enormes bases de datos resultado de numerosos estudios e investigaciones la más importante unión de la comunidad astronómica ha surgido en forma de un repositorio,

producto de un gran esfuerzo mundial denominado Observatorio Virtual (OV) el cual es libremente accesible por usuarios interesados en esta información; para llevar a cabo dicho proyecto ha sido necesario un establecimiento de los datos que serán almacenados así como la estandarización de esta información, pero aún así mucho más trabajo es necesario, después de esta facilitación del trabajo para la obtención de los datos de interés, son necesarias herramientas para la extracción de características de estas base de datos, así como para la extracción de grupos con características de interés y la interpretación de ellos. Se necesita de software computacional, pero es de total comprensión la inexistencia de un único paquete el cual provea todos los métodos necesarios, en [ ] encontramos un sistema denominado VOStat el cual está siendo desarrollado para el área de estadística y es utilizado por astrónomos, este provee capacidades a los astrónomos del OV, en este sistema ha sido necesaria la utilización de diversos conceptos de servicios web y cómputo distribuido, utilizando así las bondades de la tecnología y la computación para la facilitación de la tarea de cómputos complejos dejando así a los astrónomos nada más que la interpretación de los resultados al no tener que realizar su propio software como es realizado en algunas ocasiones (cuestión que los aleja de su objetivo).

El hecho de que astrónomos hayan abierto las puertas a investigadores de otras disciplinas se ha debido a que están conscientes que investigadores de otras áreas pueden obtener conocimiento del área por medio de cursos de Astronomía ofrecidos por universidades pero estudiosos del cielo no tienen tan fácil acceso a la formación matemática o computacional.

Estas son algunas de las muchas de las razones por las que en la actualidad decimos que la Astronomía se ha vuelto un área interdisciplinaria en la que se ha reconocido un mejor entendimiento en la materia al crear grupos multidisciplinarios, formado por áreas como la estadística y la computación e incluso permitiendo la participación del software computacional y es ahí donde nos interesa hablar de algunos métodos y software utilizados para cómputos complejos, entre los más mencionados se encuentran el `Método de Monte Carlo` usando Cadenas de Markov, también es desconocido la variedad de posibles aplicaciones para maximizar el valor esperado que pueden ser llevadas a cabo por el algoritmo `EM`. Ambas son mencionadas por [ ].

La Minería de Datos es utilizada para encontrar patrones en datos, utilizando técnicas para la construcción de modelos. Un modelo bueno es una guía útil para entender la realidad y tomar decisiones; Minería de Datos en Bases de Datos astronómicas es un esfuerzo por explorar algunos observatorios virtuales como el Centro Astronómico de Datos (ADC, disponible a través de la NASA); dichas observaciones y datos fueron reforzados con imágenes obtenidas en el Estudio Digital del Cielo (DSS), en esta investigación se realizó una selección de 1800 galaxias las cuales fueron relacionadas tratando de identificarlas en diversas bases de datos, obteniendo una tasa aproximada de 8 % de interacción en el universo.

En los últimos años a cambiado la forma de acceso a los datos astronómicos, centros de datos como: ADS (Astrophysics Data System, http://adswww.harvard.edu/), CDS (Centre de Données astronomiques de Strasbourg, http://cdsweb.u-strasbg.fr/) y NED (NASA/IPAC Extragalactic Database, http://nedwww.ipac.caltech.edu/) han formado parte del porque de esta investigación ya que es la nueva forma en que las investigaciones y publicaciones astronómicas en el mundo son llevadas a cabo es mediante el acceso a estas bases de datos astronómicas disponibles en la red y esto es gracias al creciente número de datos que están haciéndose del dominio público. Este comienzo de libre distribución de los datos es posible gracias a que en el 2003 la asamblea general International Astronomical Union (IAU) formó una alianza con las recomendaciones del Consejo Internacional para la Ciencia (ICSU) y la Organización para el Desarrollo y Cooperación Económica (OECD) quienes consideraron que el momento de comenzar una política de libre acceso a los datos astronómicos, dicha compartición de información ya existía y era la forma en que los astrónomos llegaban a un consenso entre los datos que no eran consistentes en sus bases de datos pero la llegada del Internet y la formación de alianzas la distribución de datos no solo permitirá sino propiciará este intercambio de información entre las entidades relacionadas.

Adentrándonos mas a nuestro tema hablaremos más acerca del documento de [ ] el cual nos acerca al conocimiento de un crecimiento de los datos en volumen, complejidad y calidad esto dado los avances en la tecnología a lo que la comunidad astronómica ha respondido con el VO el cual es geográfica e institucionalmente distribuido, basado en un ambiente web para Astronomía con masivas y complejas series de datos, se unifican archivos de datos y otra información, computación y herramientas para el análisis y exploración de datos.

La comunidad del OV a tenido grandes avances en la administración de la información, se tienen archivos, estándares, protocolos e interoperabilidad, sin embargo se encuentran problemas cuando hablamos de la escalabilidad de los datos ya que en esa área no se encuentran muchos avances, pero existen más dificultades y retos que surgen de la alta dimensionalidad de los datos, como los estadísticos de alta dimensionalidad y alta complejidad; pero algunos desafios mayores como la falta de herramientas para tratar con estas dificultades ha demorado los avances y son estas cuestiones las que proveen oportunidad de colaboración entre astrónomos y científicos de computación, [ ] argumenta que nos encontramos entrando a una nueva generación de análisis de datos científicos y sistemas de exploración.

Mientras asociaciones como el observatorio virtual son creadas para ayudar a las comunidades científicas por otro lado tenemos los sistemas creados de acuerdo a las necesidades que surgen, en específico hablaremos de las necesidades en Astronomía y extensos datos existentes para los cuales son necesarios sistemas que ayuden al análisis de datos el cual podría beneficiar de gran manera el trabajo de estos científicos. Dichos sistemas tienen funciones de archivación e indexación de datos, herramientas de acceso y la más importante para nosotros es una amplia variedad de métodos de Minería de Datos y de visualización.

En el Instituto de Tecnología de California (Caltech) y su Centro para la investigación de Computo Avanzado (CACR), esta siendo desarrollado un sistema diseñado para el análisis de datos a una escala de los PetaBytes.

Otro recurso utilizado para el trabajo de tan extensas bases de datos es la utilización del Grid computacional del cual podemos mencionar como ejemplo al proyecto denominado GRIST. La meta de este y otros sistemas es que el usuario se sienta cómodo, haciendo transparente la forma en que trabaja y proporcionándole las herramientas para Minería de Datos así como los algoritmos que se requieren para la visualización de los datos; el propósito de este proyecto es ser capaz de interactuar en varias áreas científicas pero está pensado para trabajar en el área de Astronomía.

Sistemas con clasificación automatizada son soluciones para el problema de clustering que están siendo elaboradas ya que estos representan las soluciones a las nuevas necesidades de esta generación de grandes cantidades de información. En estos trabajos se presenta la aplicación de técnicas de Machine Learning e Inteligencia Artificial para la clasificación de estructuras astronómicas.

SKICAT es un sistema utilizado para el análisis y registro de datos astronómicos cuya tarea consiste en clasificar datos y fotografías como parte del Segundo *Palomar Observatory Sky Survey* (POSSII): SKICAT utiliza técnicas de Machine Learning para clasificar objetos del cielo, en especial estrellas y galaxias.

Se denomina *Aprendizaje Supervisado* en Astronomía al aprendizaje en el cual contamos con una serie de entrenamiento que consiste de una serie de objetos que han sido clasificados previamente por algún astrónomo experto y son introducidos al sistema como ejemplos para que el sistema aprenda a clasificar nuevos datos, a diferencia de un *Sistema de Aprendizaje No Supervisado* en el cual se clasifican nuevas entradas sin ejemplos de clasificación previas, en este tenemos la ventaja de poder encontrar nuevas características previamente desconocidas por los investigadores, el aprendizaje no supervisado es el área de nuestro interés y en esta podemos encontrar un concepto muy interesante en Astronomía denominado *Conceptual Clustering* el cual encuentra útiles categorías en datos no clasificados de una manera muy similar al método cotidiano utilizado por los humanos es decir separar objetos similares en la misma clase y en diferentes si no.

Un muy popular sistema de Conceptual Clustering es el **COBWEB** realizado en 1987 por Fisher el cual utilizaba una medida de heurística denominada utilidad categórica, esta fue creada por Gluck y Corter y trabaja con jerarquías simples determinadas por el nivel más bajo de clasificación realizada por los humanos, concluyendo que ciertas categorías son mas fácilmente reconocidas por los humanos que otras, pero esta medida de heurística unicamente era útil para clasificación de datos categóricos, esto cambió con la creación de **COBWEB/95** en el cual se pueden utilizar datos categóricos y numéricos utilizando una estructura de descripciones atributos-valor.

El problema de separación de estrellas y galaxias es tratado con Machine Learning por [ ] con un sistema que recibe los objetos producto de observaciones de manera incremental y los acomoda en un árbol de clasificación organizando de la manera más eficiente; donde cada nodo representa una clase en la cual resumimos sus características más relevantes y donde el nodo de nivel más alto representará el concepto más general y el nodo raíz el concepto mas específico. Este sistema ha sido aplicado a numerosos objetos en el cielo para determinar atributos o nuevos objetos los cuales han derivado del sistema de catalogación y análisis SKICAT; el experimento fue realizado con una base de datos de 33021 datos con 10 atributos fotográficos entre los cuales no se usa la categoría çolor"la cual sería firmemente decisiva en la clasificación de los objetos, no fue utilizada para demostrar el poder del método. Los resultados muestran clases inferiores en el caso donde la clase mayoritaria contenga una exactitud de 94 % y en los resultados podemos observar la efectividad del método en el cual para cada nivel llegamos a obtener resultados mayores a este umbral en niveles muy próximos a la raíz.

Otro ejemplo de experimentos realizados, fue el trabajo realizado por [ ] en el que se utilizó un programa llamado *AutoClass* el cual fue capaz de categorizar en cuatro clases usando unos pocos y muy simples atributos de imágenes de objetos utilizando las mismas precondiciones que en el ejemplo anterior.

En [ ] se presenta un trabajo realizado con redes neuronales para la caracterización de imágenes para la cual también fueron utilizados ejemplos de objetos clasificados como estrellas o galaxias, aunque la SOM no utiliza estos ejemplos para la modificación de los pesos; una parte de los datos de entrenamiento es utilizada para la transformación del mapa, el nodo que modifica de manera más cercana a cada objeto será el adecuado para representar a ese objeto y la habilidad con la que la modificación es llevada a cabo determinará el número de objetos necesario para ser clasificados durante el entrenamiento, concluyendo un mucho menor número de ejemplos de entrenamientos necesarios para la ejecución de un mapa de Kohonen que los requeridos para la ejecución de una red neuronal.

2.2.2. Minería de Datos Espacial

El desarrollo de este trabajo esta basado en los algoritmos de Minería de Datos Espacial, entre los métodos mas populares está el método de *K-MEANS*[ ], en este metodo un número determinado k es elegido aleatoriamente para representar los centroides de los clusters y los nodos restantes son asignados a su centroide mas cercano. una vez realizado esto los centroides son recalculados y el algoritmo realiza una nueva iteración hasta cumplir el total de iteraciones seleccionadas, k-medios pertenece a la clase de algoritmos particionales. Otros métodos particionales basados en el algoritmo de k-medios son: *PAM: Partitioning Around Medoids (1987)* y *CLARA: Clustering Large Aplications (1990)* los cuales fueron desarrollados por Kaufman Rousseeuw, en los cuales un elemento es tomado como medioide

de los clusters y los elementos restantes son asignados al cluster con el que presenten mayor similitud, la diferencia entre estos dos métodos es que el primero utiliza el total de la base de datos mientras que el segundo unicamente toma una porción de los datos basandose en la teoría de que de esta manera obtenemos una mejor representación de los datos. A partir de estos dos últimos métodos, surgió la creación de uno que ha llegado a una gran relevancia en la actualidad por ser el primer algoritmo diseñado para datos espaciales *CLARANS: Clustering Large Applications based on RANdomized Search (1994)*, este fue desarrollado para el análisis de agrupamientos mediante medioides.

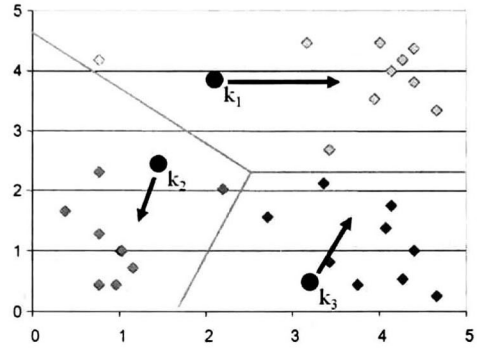

Figura 2.3: Ejemplo de Algoritmo de Clustering Particional: K-means. Este algoritmo representa cada una de sus particiones (clusters) por medio de su centroide.

A este siguió el algoritmo *DBSCAN: Density Based Spatial Clustering of Applications with Noise (1996)* [ ][ ] el cual forma parte de los algoritmos basados en densidad; este algoritmo está fundamentado en la localización de los objetos basandose en su cercanía y cantidad mínima de puntos que deben pertenecer al cluster para ser considerado como tal, el algoritmo inicia seleccionando un punto arbitrario p el cual designa a todos los objetos cercanos a el según los parámetros antes mencionados como densamente-alcanzables desde p, hasta recorrer todos los elementos, al terminar los elementos dentro de algún grupo son considerados centrales, los no asignados a ningún grupo son ruido y los restantes son puntos borde. De este algoritmo se derivan directamente otros como: *DENCLUE: DENsity-based CLUstEring(1998)* [ ] el cual define una función de densidad, gradiente y punto atractor; está conformado de dos fases: en la primera divide el hiper-rectangulo de datos en hipercubos y determina cuales de ellos son los mas poblados ó los conectados y en su segunda fase considera aquellos hipercubos determinados en la primera fase para especificar los grupos. *OPTICS: Ordering Points To Identify the Clustering Structure (1999)* [ ] el cual realiza un ordenamiento previo de la base de datos y representa graficamente para un mejor entendimiento; *GBSCAN: Generalized Density-Based Spatial Clustering of Applications with Noise(1998)*[ ] el cual basado en DBSCAN hace una generalización de la noción de punto de densidad.

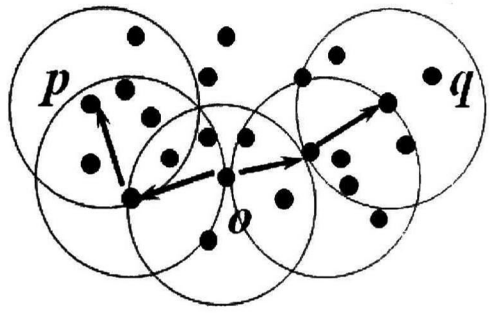

Figura 2.4: Ejemplo de Algoritmo de Clustering Basado en Densidad : DBSCAN.

BIRCH: Balanced Iterative Reducing and Clustering using Hierarchies (1996) [ ] [ ] [ ] es un algoritmo incremental que utiliza estructuras específicas denominadas clusters de características para ejecutar el clustering. *CURE: An efficient clustering algorithm for large databases (1998)* [ ] [ ] utiliza un conjunto representativo por cada cluster en lugar de un único punto, se caracteriza por ser de tipo aglomerativo e ir añadiendo grupos hasta obtener k grupos. *ROCK: RObust Clustering using linKs (1999)* [ ] en cambio es un algoritmo mas específico para datos cualitativos y categóricos que utiliza el concepto de enlaces en lugar de distancias para la asignación de los clusters, definiendo un determinado número de k grupos y uniendo los elementos al grupo que maximize la función de acuerdo al tamaño y la estimación del número de vecinos. Por último presentaremos a *CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling (1999)* [ ] trabaja con teoría de gráfos.

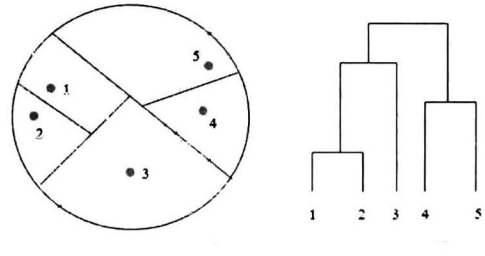

Figura 2.5: Ejemplo de Algoritmo de Clustering Jerárquico : BIRCH.

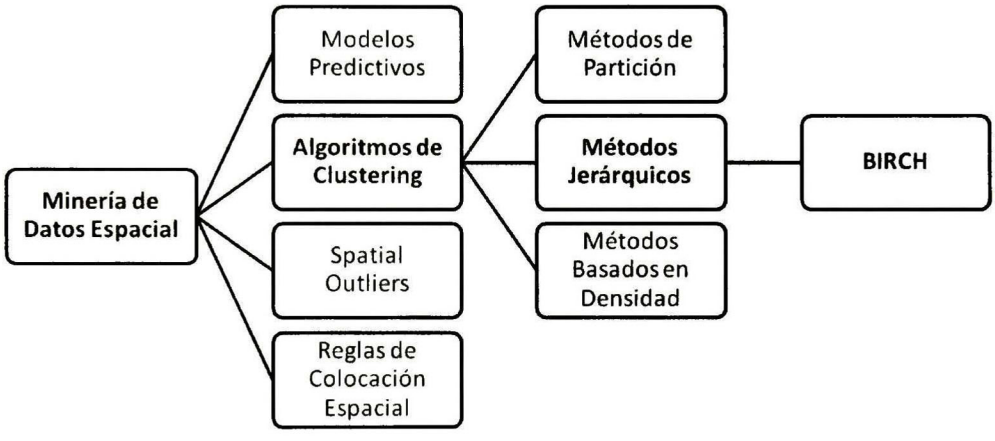

Figura 2.6: Estado del Arte del Área de Clustering en la Minería de Datos Espacial.

Capítulo 3

Metodología

En este capítulo revisaremos el método sobre el cual basaremos nuestro análisis, lo estudiaremos, deduciremos la importancia de nuestro trabajo y la propuesta será definida.

3.1. BIRCH

El algoritmo de *Reducción y Clustering Iterativamente Balanceado usando Jerarquías* BIRCH por sus siglas en inglés Balanced Iterative Reducing and Clustering using Hierarchies es un incremental y efectivo algoritmo para computar clusters en largas series de datos [ ] [ ]. Los tres puntos claves para el desarrollo del algoritmo son:

1. **Cluster de Características.** Representación resumida de la información del cluster correspondiente [ ].

   DEFINICIÓN 3. **Cluster de Características (CF).** Sea C un cluster con n Objetos $O_1, O_2, ..., O_n$. la estructura del correspondiente [ ] $CF = (N, \vec{LS}, SS)$ es:

   - N número de vectores.
   $$N = |C| \tag{3.1}$$
   - LS suma linear de vectores.
   $$\vec{LS} = \sum_{i=1}^{N} \vec{O_i} \tag{3.2}$$
   - SS suma de vectores cuadrados.
   $$SS = \sum_{i=1}^{N} \vec{O_i}^2 \tag{3.3}$$

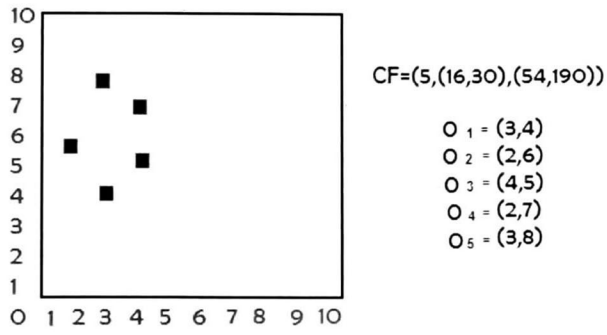

Figura 3.1: Ejemplo de un Cluster de Características CF

Un CF es incrementalmente mantenible cuando:

* Nuevos objetos son insertados en el cluster.
* Dos clusters son mezclados.

DEFINICIÓN 4. **Propiedad de Aditividad**
Sean $CF_1 = (N_1, \vec{LS}_1, SS_1)$ y $CF_2 = (N_2, \vec{LS}_2, SS_2)$ clusters disjuntos, entonces la suma de ellos es:
$CF_1 + CF_2 = (N_1 + N_1, \vec{LS}_1 + \vec{LS}_2, SS_1 + SS_2)$

2. **Árbol de características.** Representación resumida de los clusters, esto es ya que cada nodo no hoja contiene el CF del cluster de objetos del subárbol correspondiente.

   DEFINICIÓN 5. **Árbol de características (CF-TREE).** Representación del índice del cluster de características.

   Un CF-TREE está conformado por [ ] :

   * Factor de ramificación B.
   * Umbral T.

   Un CF-TREE contiene dos tipos de nodos [ ]:

   - **Nodo Hoja**
     - B entradas $[CF_i, hijo_i]$
     - $hijo_i$: Puntero al i-ésimo hijo
     - CF_i : Cluster de Características del subcluster representado por su hijo.
   - **Nodo No Hoja**
     - B entradas $[CF_i]$

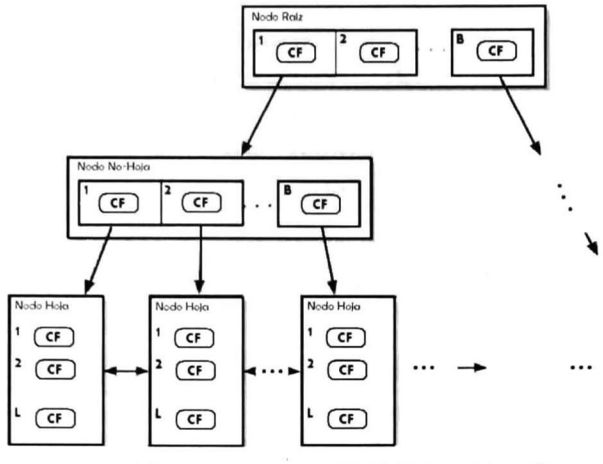

Figura 3.2: Estructura de un CF-TREE.

- $*previo$
- $*siguiente$

3. **Algoritmo de Reconstrucción.** El árbol debe ser modificado con respecto a las iteraciones y debe ser tranformado en uno mas pequeño; un nodo hoja es seleccionado por el cluster mas cercano para insertar el objeto: Si el umbral T es satisfecho, el objeto es absorbido por el cluster, de otra manera este forma un nuevo cluster en las hojas; el camino de la raíz hacia la hoja de inserción es modificado reflejando así dicho cambio [ ] [ ] [ ].

3.2. Algoritmo

BIRCH consta de cuatro fases principales las cuales serán explicadas a continuación [ ]:

Fase 1: Construcción del CF-TREE

Cuando un nuevo elemento debe ser insertado al CF-TREE, el elemento desciende recursivamente desde la raíz hasta los nodos hojas para elegir el Nodo Hoja mas apropiado de acuerdo al criterio de distancia elegido, tabla 3.1.

Una vez elegido el Nodo Hoja, si el elemento al ser añadido a la entrada correspondiente en el Nodo asignado cumple que el radio del *Cluster Feature CF* del nodo respectivo es menor que el *Umbral T*, el elemento podrá ser insertado. Las opciones posibles de este caso se muestran en la tabla 3.2.

	Distancia
D0	Euclideana
D1	Manhattan
D2	InterCluster
D3	IntraCluster
D4	Varianza

Tabla 3.1: Métricas de Distancia.

Situación Inicial		**Situación Final**	
cf1|x	cf1<nuevoDato	cf1|nuevoDato	
cf1|x	cf1>nuevoDato	nuevoDato|cf1	

Tabla 3.2: Caso 1: Inserción cuando hay espacio en el Nodo Hoja.

En caso contrario, si no hay espacio en el Nodo Hoja para añadir un nuevo *Data Point*, el nuevo *Objeto* es añadido a esta nueva entrada, si no hay espacio para una nueva entrada por que el Nodo Hoja esta lleno, dicho nodo es dividido. Las opciones posibles de este caso se muestran en la tabla 3.3.

Cuando dividimos algun nodo hoja, debemos elegir el par de elementos mas lejanos como semillas para la reasignación de los demás elementos en la división.

Situación Inicial	**Situación Final**
cf1|cf2 ; nuevoDato>cf2	cf1|x nuevoDato|x ; sube cf2
cf1|cf2 ; nuevoDato<cf1	nuevoDato|x cf2|x ; sube cf1
cf1|cf2 ; cf1<nuevoDato<cf2	cf1|x cf2|x ; sube nuevoDato

Tabla 3.3: Caso 2: Inserción cuando no hay espacio en el Nodo Hoja.

Ya sea añadiendo el nuevo elemento a una nueva entrada ó a alguna ya existente, estos cambios se ven reflejados en la actualización de los caminos hacia el Nodo Hoja, de la misma manera el Nodo Padre debería prepararse para una posible división.

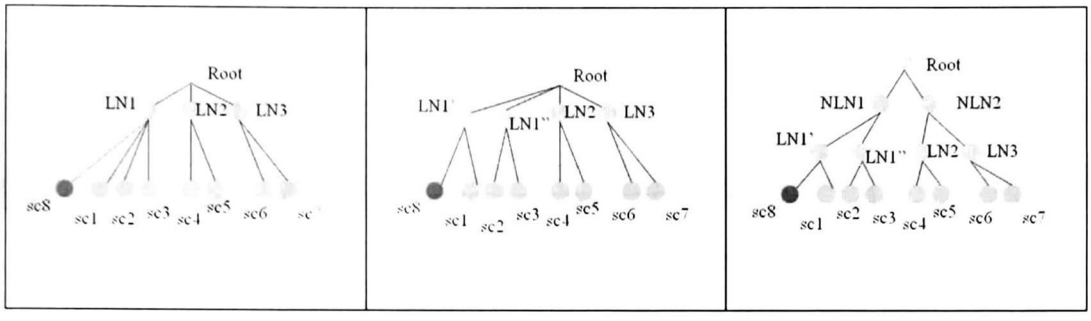

Figura 3.3: Actualización de camino del árbol hacia un Nodo Hoja, como consecuencia de la creación de una nueva entrada. B=3 y llega sc8, el cual tiene como entrada su cercana LN1, pero al ser añadido excede B=3, así que es necesario dividir LN1 en LN1 y LN1', lo cual provoca a su vez que el Nodo Raíz sea ahora quien exceda B=4 y produciendo la división de este en NLN1 y NLN2 provocando un crecimiento del árbol.

Algoritmo 3.1 Algoritmo de Clustering

Entrada: $D = O_1, O_2, ..., O_n, T$

Salida: K Clusters

1: **para** $O_i \in D$ **hacer**
2: determinar el correcto Nodo Hojapara insertar O_i.
3: **si** la condición de Umbral T es satisfecha **entonces**
4: O_i es añadido al Nodo Hoja y absorbido por el cluster, el CF actualizado.
5: **si no**
6: **si** hay espacio para una nueva entrada **entonces**
7: la nueva entrada es añadida a la hoja y en ella el nodo, el CF actualizado.
8: **si no**
9: dividiremos el Nodo Hoja y redistribuiremos los datos actualmente existentes de acuerdo al criterio de cercanía, los CFs son actualizados.
10: **fin si**
11: **fin si**
12: **fin para**

Figura 3.4: Paso 1, creación del Nodo Raíz con una única entrada $[CF_1]$.

Figura 3.5: Paso 2, el Nodo Raíz con el total de sus entradas permitidas, B=3.

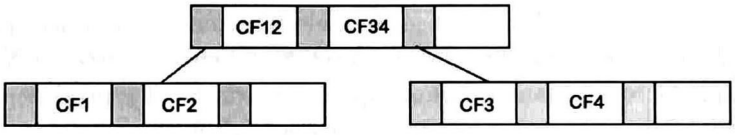

Figura 3.6: Paso 3, división del Nodo Hoja y actualización del árbol como consecuencia de la llegada de un nuevo objeto y la creación de una nueva entrada.

Si el árbol crece demasiado, podría llegar a ocupar mas espacio del asignado, el Umbral es incrementado y el árbol es reconstruido. En esta reconstrucción, los nodos existentes en el árbol anterior son insertadas al nuevo árbol de la misma manera que la inserción de objetos originales.

Fase 2: Condensación del árbol en el rango deseado (Fase Opcional)

Modificación del CF-TREE para lograr la reducción de su tamaño. Para el ajuste del árbol necesitamos primero hacer un ajuste del valor del *Umbral* para eso empezaremos definiendo algunos de los parámetros necesarios para el cálculo de este.

Factor de Expansión: es el factor utilizado como un parámetro para la función de crecimiento, y no debe ser menor que uno.
r es el radio promedio de los clusters y es calculado mediante regresión lineal.

$$f = Max(1{,}0, \frac{r_{i+1}}{ri}) \tag{3.4}$$

Distancia Mínima entre las dos entradas cercanas a la hoja.

T_{i+1} calculado mediante regresión lineal.

Nuevo Umbral es:

$$T_{i+1} = Max(D_{min}, T_{i+1} * f) \tag{3.5}$$

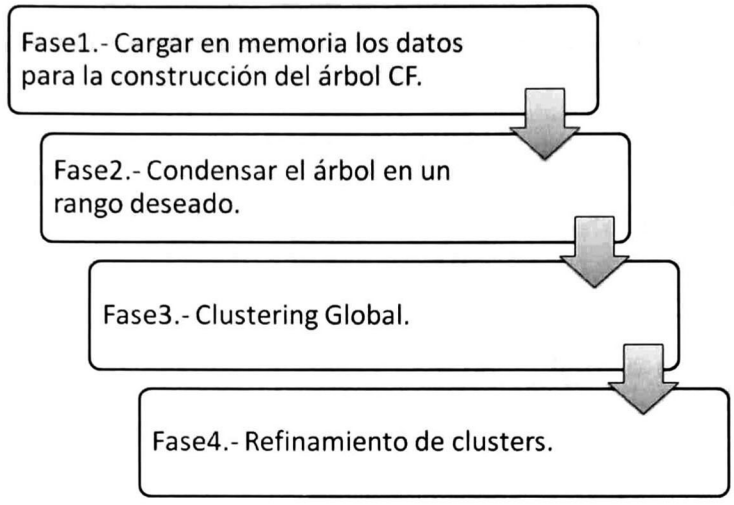

Figura 3.7: Reducción y Clustering Iterativamente Balanceado usando Jerarquías.

a menos que este valor resulte ser menor que el valor anterior del Umbral, en ese caso utilizariamos el valor de:

$$Aproximacion = T_i * (\frac{N_{i+1}}{Ni})^{\frac{1}{d}} \tag{3.6}$$

Es decir el Nuevo Umbral toma el valor entre T_{i+1} y la *Aproximacion*:

$$NuevoUmbral = Max(T_{i+1}, Aproximacion) \tag{3.7}$$

Fase 3: Clustering Global .

Cada Nodo Hoja es tomado como un elemento individual sobre el cual ejecutaremos el algoritmo de clustering **K-Means**.

Fase 4: Refinamiento del Software (Fase Opcional) .

Realiza un último Clustering en todos los datos, situándolos en el cluster mas cercano.

3.3. Complejidad

Consideramos el tiempo de la ejecución de un algoritmo como el tiempo que le toma a un algoritmo realizar el total de los calculos a partir de los datos de entrada.

La complejidad, en cambio permite darnos una representación del tiempo esperado para obtener una respuesta.

Fase 1 Sean: M memoria, P paginación.

El tamaño del árbol puede alcanzar un tamaño máximo de:

$$\frac{M}{P} \tag{3.8}$$

para insertar un dato al árbol, recorreremos un camino de la raíz a la hoja, lo cual corresponde a:

$$1 + \log_B \frac{M}{P} \tag{3.9}$$

además dentro de cada nodo debemos inspeccionar B entradas hasta encontrar el elemento mas cercano y corespondiente. Siendo así el costo de la entrada proporcional a la dimensión.

El costo de la inserción del total de los puntos es:

$$O(d * N * B(1 + \log_B \frac{M}{P})) \tag{3.10}$$

Si la reconstrucción del árbol es necesaria, tendremos:

$$C * d \tag{3.11}$$

con C constante mapeada a la dimensión d del árbol.

Así tendremos que la cota máxima de nodos para reconstrucción es:

$$\frac{M}{C * d} \tag{3.12}$$

por lo tanto el costo de la reconstrucción del árbol queda:

$$O(d * \frac{M}{C * d} * B(1 + \log_B \frac{M}{P})) \tag{3.13}$$

Una teoría importante para nuestra investigación señala que el número de reconstrucciones de un árbol depende de la heurística utilizada.

La heuristica utilizada originalmente es:

$$\log_2 \frac{N}{N_0} \tag{3.14}$$

el valor 2 es ya que el tamaño del árbol está limitado hasta la mitad de tamaño, N_0 es el número de puntos de datos cargados en memoria con el treshold inicial T_0.

basandonos en los hechos previos, podemos resumir el **costo de la primera fase:**

$$O(d * N * B(1 + \log_B \frac{M}{P}) + \log_2 \frac{N}{N_0} * d * \frac{M}{C * d} * B(1 + \log_B \frac{M}{P})) \tag{3.15}$$

Fase 2 El análisis del costo de la fase 2 es análoga a la de la Fase 1, por otro lado podemos decir que así como para la Fase 1 realizamos la lectura de la base de datos, para la fase dos tenemos un costo similar correspondiente a la inserción de los outliers y su lectura dentro de la re-inserción.

Dada la cantidad disponible de memoria en disco para la manipulación de los outliers, y las $\log_2 \frac{N}{N_0}$ reconstrucciones, podemos decir que el costo de las primeras fases no es mayor que el de la simple lectura de los datos.

Fase 3 El costo de esta fase se encuentra determinado por el tamaño máximo de rango de entrada y el algoritmo de clustering elegido.

El costo de las fases 1, 2 y 3 aumental linealmente con respecto a N.

Fase 4 Reacomoda cada punto en el respectivo cluster, por lo que nuevamente el tiempo que toma llevar a cabo dicho reacomodo es proporcional a $N * K$.

Por el análisis anterior podemos concluir que el algoritmo tiene un costo de $O(N)$.

3.4. Propuesta

El algoritmo de BIRCH cuenta con una conveniente complejidad lineal, a pesar de poseer la debilidad de requerir reconstrucciones cada vez que el umbral del árbol CF es excedido.

Es por eso que acontinuación estudiaremos las reconstrucciones producto de la heuristica para elegir el umbral.

La propuesta se basa en la elección de un algoritmo de aprendizaje para el cual requeriremos de un número menor o nulo de reconstrucciones, esto será llevado a cabo determinando el conjunto de reglas que definirán el modelo de tal manera que el valor del umbral encontrado será mas adecuado que el utilizado por la heuristica original.

Capítulo 4

Experimentos

En este capítulo se presenta el desarrollo e implementación de la metodología propuesta, la cual es descrita y justificada por medio de los experimentos, pruebas y comparaciones expuestos en el mismo.

4.1. Resultados

A continuación presentaremos la serie de datos con los que trabajaremos,los cuales corresponden a la siguiente descripción:

Atributo	Byte	Descripción
1	1	A ó S según el catálogo ACO 1989
2	2 – 5	Catálogo de Abell
3	6	Componente de desplazación en el cielo: E,W,N,S
4	7 – 15	Ascención Recta
5	16 – 24	Declinación
6	25 – 30	Velocidad Radial
7	32 – 34	Error de Velocidad
8	36 – 38	Código de Referencia

Tabla 4.1: Atributos de la Base de Datos.

cúmulo	número de miembros
A0550A	128
A2204A	111
A2219	117
A3542	113

Tabla 4.2: Cúmulos de Galaxias analizados en los experimentos. La letra A que se encuentra eal final del nombre de los dos primero cúmulos significa que se encuentran cerca del principal (A0550), pero no tanto como para formar parte del mismo; los cuatro cúmulos fueron seleccionados por contener mas de 100 elementos con al menos una Velocidad Radial.

A2219 163954.53+464016.4 67614 223 661
A2219 163954.53+464016.4 67614 223 661
A2219 163955.54+464101.4 68948 117 661
A2219 163956.69+464140.4 67403 97 661
A2219 163956.76+464335.4 68244 87 661
A2219 163959.26+464142.0 69556 248 661

Tabla 4.3: Muestra de la Base de Datos.

Donde algunos de los atributos no se encuentran en el formato apropiado que proporcionará información acerca de su ubicación, así que primero se necesita un preprocesamiento, en donde dos conversiones convenientes son la Ascención Recta y Declinación.

Ascención Recta:

$$\frac{RA}{deg} = [hh + \frac{mm}{60} + \frac{ss}{3600}] * 15; \tag{4.1}$$

Declinación:

$$\frac{DE}{deg} = signo * [dd + \frac{mm}{60} + \frac{ss}{3600}]; \tag{4.2}$$

Ahora, de los ocho atributos con los que contamos no todos son relevantes, para extraer las características primordiales se utilizaron algoritmos de selección de características, de los cuales se presenta un análisis de los resultados de dicha selección en la imagen 4.1.

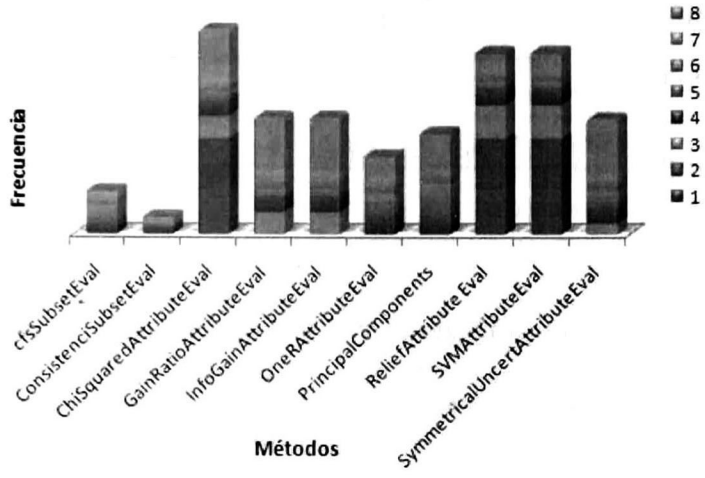

Figura 4.1: Selección de Atributos.

Los atributos 4, 5, 6 y 7 resultaron ser los principales; tabla 4.4.

Atributo	Byte	Descripción
4	7 – 15	Ascención Recta
5	16 – 24	Declinación
6	25 – 30	Velocidad Radial
7	32 – 34	Error de Velocidad

Tabla 4.4: Atributos Seleccionados.

La implementación del algoritmo *BIRCH* fue realizada en C++ bajo la plataforma de Linux.

Una descripción general, es la definición de los parámetros iniciales.

I. Máximo número de nodos permitidos en el árbol.

II. Medida de distancia.

III. Número de objetos.

IV. Número de dimensión (número de atributos).

V. Tamaño de página.

VI. Umbral inicial.

VII. Detector de Outliers.

new CFTree(umbral,tamañoPagina,dimension);

cfTree→insertDataPoint(dp);
Recorremos el total de los elementos en la base de datos y los insertamos al árbol de acuerdo al algoritmo establecido.

if(numeroNodos ≥ maximoNumeroNodos) { cfTree = cfTree → rebuildNewTree(NULL) }
Si el número de nodos en el árbol hasta ese momento es mayor que el número máximo de nodos permitido, el árbol debe ser reconstruido con un mayor umbral que produzca un menor número de nodos.

Cada uno de los CFs del nivel de las hojas es tomado como un elemento singular y el algoritmo de clustering elegido *k-Means* es ejecutado sobre los algoritmos;

A continuación se muestran los resultados obtenidos sobre cada una de las bases de datos de 111, 113, 117 y 128 galaxias con un umbral de 10 y un máximo de nodos de 40.

Al ejecutar el algoritmo con el cúmulo A2204A de 111 datos obtenemos un total de *14 reconstrucciones*, y obtenemos un árbol como el que muestra la figura *fig:111* con un único nodo en el nivel cero y cuatro nodos en el nivel 1, cada uno con: 7, 5, 4 y 5 nodos respectivamente.

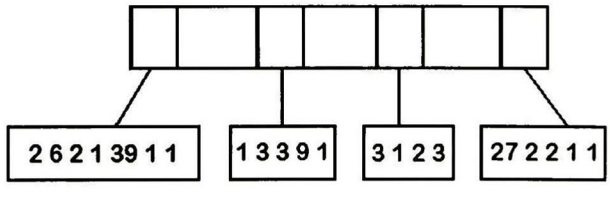

Figura 4.2: Clustering de árbol de 111 datos.

Al ejecutar el algoritmo con el cúmulo A3542 de 113 datos obtenemos un total de *15 reconstrucciones*, y obtenemos un árbol como el que muestra la figura *fig:113* un único nodo en el nivel cero y cuatro nodos en el nivel 1, cada uno con: 4, 6, 6 y 8 nodos respectivamente.

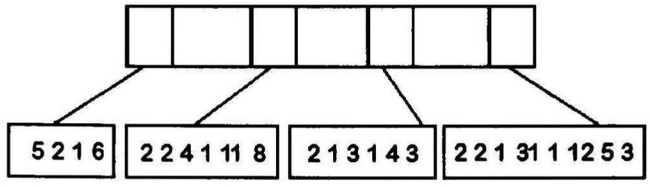

Figura 4.3: Clustering de árbol de 113 datos.

Al ejecutar el algoritmo con el cúmulo A2219 de 117 datos obtenemos un total de *35 reconstrucciones*, y obtenemos un árbol como el que muestra la figura *fig:117* un único nodo en el nivel cero y cuatro nodos en el nivel 1, y para el primer nodo del nivel 1, cuatro nodos más en el nivel 2 con 8, 8, 8 y 8 nodos, para el segundo nodo del nivel 1, tres nodos más en el nivel 2 con 7, 7, y 3 nodos, para el tercer nodo del nivel 1, un nodo más en el nivel 2 con 2 nodos, para el cuarto y último nodo del nivel 1, dos nodos más en el nivel 2 con 3 y 3 nodos respectivamente.

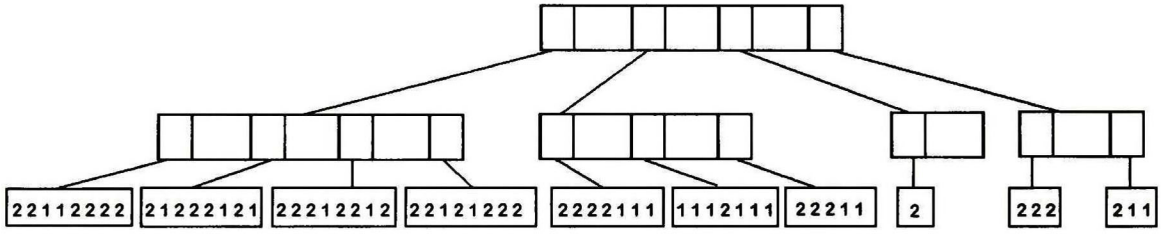

Figura 4.4: Clustering de árbol de 117 datos.

Al ejecutar el algoritmo con el cúmulo A0550A de 128 datos obtenemos un total de *16 reconstrucciones*, y obtenemos un árbol como el que muestra la figura *fig:128* un único nodo en el nivel cero y cinco nodos en el nivel 1, cada uno con: 8, 8, 3, 7 y 7 nodos respectivamente.

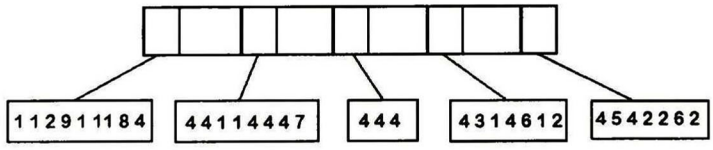

Figura 4.5: Clustering de árbol de 128 datos.

Problema: El algoritmo cuenta con deficiencias al momento de la inicialización de los parámetros; uno de ellos es el número de veces que el árbol será reconstruido, este número depende en gran medida del *umbral* T establecido como parámetro inicial.

Dentro del algoritmo de BIRCH, este problema es resuelto mediante una *Aproximación Lineal* que predice un valor mayor al de la iteración actual, sin embargo podríamos llegar al valor apropiado de umbral hasta despues de varias iteraciones, provocando así más de una reconstrucción.

Para solucionar este problema, trataremos el cálculo del umbral, como un subproblema de *Aprendizaje*.Como tal, el primer paso es la caracterización del problema y realizar el aprendizaje utilizandolas para el entrenamiento.

Descripción de las columnas: 1 = número de nodos en la base de datos, 2 = RA media, 3 = RA desviacion estándar, 4 = DE media, 5 = DE desviación estándar, 6 = RV media, 7 = RV desviación estándar, 8 = VE media, 9 = VE desviación estándar. 10 = numero máximo de nodos, 11 = umbral adecuado; las columnas de la 2 a la 8 son descriptores de las columnas (RA,DE,RV,VE) de la base de datos de entrenamiento que explican su distribución, la columna 10 fue obtenida mediante la experiencia de experimentos realizados.

1	2	3	4	5	6	7	8	9	10	11
128	88.135	0.413	-21.178	0.347	29405.164	1263.165	66.859	8.357	20	61
120	88.087	0.378	-21.193	0.345	29454.083	1269.485	66.83	7.47	18	76
110	88.036	0.352	-21.156	0.325	29503.918	1267.805	66.145	7.438	15	76
100	87989	0.335	-21.168	0.316	29462.97	1296.507	65.84	6.996	15	47
90	87.943	0.321	-21.165	0.327	29359.6333	1300.919	65.756	6.895	14	46
80	87.8999	0.313	-21.17	0.321	29361.975	1256.028	65.65	6.764	12	61
70	87.85	0.305	-21.178	0.338	29229.357	1219.404	65.857	6.744	11	47
60	87.793	0.292	-21.1999	0.355	29154.333	1272.306	66.167	7.247	10	47
50	87.726	0.275	-21.23	0.366	29106.26	1280.426	66.08	7.125	10	31
40	87.649	0.254	-21.196	0.368	29039.4	1311.883	65.95	6.935	5	47
30	87.556	0.225	-21.259	0.351	28827.067	1370.496	66.6	7.933	4	47
20	87.426	0.152	-21.265	0.259	28531.25	1425.45	66.6	8.003	4	16
10	87.3	0.07	-21.265	0.232	28819.1	1625.286	64	0.07	1	46

Tabla 4.5: Base de Datos de Entrenamiento.

Ahora ejecutamos los algoritmos de Clasificación Numérica; Lo que se busca es el algoritmo que de mejores resultados una vez realizado el entrenamiento, mostramos los resultados obtenidos al aplicar los diferentes algoritmos y a partir de los resultados obtenidos determinaremos cual es la mejor clasificación.

1. functions : Métodos matemáticos.

   a) *functions.LinearRegression*

	LinearRegression
Correlation coefficient	0.04
Mean absolute error	1720.2331
Root mean squared error	11546.882
Relative absolute error	1438.73
Root relative squared error	4455.81

Tabla 4.6: LinearRegression. Regresion Lineal estándar para predecir valores.

   b) *functions.LeastMedSq*

	LeastMedSq
Correlation coefficient	0.3085
Mean absolute error	425.6369
Root mean squared error	2265.8999
Relative absolute error	355.9842
Root relative squared error	874.39

Tabla 4.7: LeastMedSq. Utiliza la regresión lineal media de cuadrados mínimos utilizando la regresión lineal para formar predicciones.

   c) *functions.MultilayerPerceptron*

	MultilayerPerceptron
Correlation coefficient	0.2824
Mean absolute error	117.9683
Root mean squared error	246.5362
Relative absolute error	98.66
Root relative squared error	95.14

Tabla 4.8: MultilayerPerceptron. Red neuronal de retropropagación.

d) functions.PaceRegression

	PaceRegression
Correlation coefficient	-0.092
Mean absolute error	5346.6856
Root mean squared error	32505.7067
Relative absolute error	4471.74
Root relative squared error	12543.59

Tabla 4.9: PaceRegression. Clase para construir modelos de regresión lineal y usarlos para predicciones.

e) functions.SimpleLinearRegression

	SimpleLinearRegression
Correlation coefficient	0.6224
Mean absolute error	92.7031
Root mean squared error	209.5564
Relative absolute error	77.53
Root relative squared error	80.87

Tabla 4.10: SimpleLinearRegression. Regresión Lineal Simple para aprendizaje.

f) functions.SMOreg

	SMOreg
Correlation coefficient	0.0994
Mean absolute error	309.7089
Root mean squared error	1812.7374
Relative absolute error	259.03
Root relative squared error	699.52

Tabla 4.11: SMOreg. Algoritmo de optimización de secuencial mínima en la formación de un vector de apoyo o de regresión usando polinomial o RBF nucleos.

II. lazy : Métodos que no construyen un modelo.

a) lazy.IBk

	Ibk
Correlation coefficient	1
Mean absolute error	0
Root mean squared error	0
Relative absolute error	0
Root relative squared error	0

Tabla 4.12: Ibk. k vecinos más cercanos.

b) lazy.KStar

	Kstar
Correlation coefficient	0.1514
Mean absolute error	78.1457
Root mean squared error	262.7136
Relative absolute error	65.3577
Root relative squared error	101.3782

Tabla 4.13: Kstar. Clasificador basado en ejemplos.

c) lazy.LWL

	LWL
Correlation coefficient	0.2188
Mean absolute error	76.647
Root mean squared error	251.5123
Relative absolute error	64.1
Root relative squared error	97.06

Tabla 4.14: LWL. Aprendizaje basado en pesos locales.

III. meta : Métodos que utilizan una combinación de métodos de aprendizaje.

a) meta.AdditiveRegression

	AdditiveRegression
Correlation coefficient	0.1917
Mean absolute error	90.6531
Root mean squared error	255.178
Relative absolute error	75.8183
Root relative squared error	98.4703

Tabla 4.15: AdditiveRegression. Meta clasificador que mejora el rendimiento de una clasificador base.

	Ibk	**KStar**	**LWL**
Correlation coefficient	1	1	0.9936
Mean absolute error	0	0	17.292
Root mean squared error	0	0.0002	28.7504
Relative absolute error	0	0	14.786
Root relative squared error	0	0.0001	11.2974

Tabla 4.16: AdditiveRegression Ibk, KStar, LWL sobre la base de datos de entrenamiento.

	Ibk	**KStar**	**LWL**
Correlation coefficient	-0.1757	-0.1805	-0.1435
Mean absolute error	70.2941	69.8428	62.6067
Root mean squared error	142.9498	141.1856	124.3978
Relative absolute error	72.0261	71.5636	64.1493
Root relative squared error	142.536	140.777	124.0378

Tabla 4.17: AdditiveRegression Ibk, KStar, LWL con un split de 66 %.

b) meta.Bagging

	Bagging
Correlation coefficient	-0.0154
Mean absolute error	102.0121
Root mean squared error	256.6027
Relative absolute error	85.32
Root relative squared error	99.02

Tabla 4.18: Bagging. Clase para empaquetar un clasificador.

	Ibk	**KStar**	**LWL**
Correlation coefficient	0.99	0.99	0.96
Mean absolute error	12.62	13.85	38.8
Root mean squared error	33.14	34.35	71.15
Relative absolute error	10.79	11.85	33.17
Root relative squared error	13.02	13.50	27.96

Tabla 4.19: Bagging Ibk, KStar, LWL sobre la base de datos de entrenamiento.

	Ibk	**KStar**	**LWL**
Correlation coefficient	0.01	0	-0.22
Mean absolute error	67.31	68.08	66.12
Root mean squared error	136.78	136.68	129.74
Relative absolute error	68.97	69.76	67.74
Root relative squared error	136.39	136.28	129.37

Tabla 4.20: Bagging Ibk, KStar, LWL con un split de 66 %.

c) *meta.CVParameterSelection*

	CVParameterSelection
Correlation coefficient	-0.3967
Mean absolute error	119.5662
Root mean squared error	259.142
Relative absolute error	100
Root relative squared error	100

Tabla 4.21: CVParameterSelection. Clase para realizar selección de parámetros por validación-cruzada para cualquier clasificador.

	Ibk	**KStar**	**LWL**
Correlation coefficient	1	1	0.97
Mean absolute error	0	0.66	32.52
Root mean squared error	0	2.57	64.83
Relative absolute error	0.00	0.57	27.81
Root relative squared error	0.00	1.01	25.47

Tabla 4.22: CVParameterSelection Ibk, KStar, LWL sobre la base de datos de entrenamiento.

	Ibk	**KStar**	**LWL**
Correlation coefficient	-0.18	-0.18	-0.27
Mean absolute error	70.29	69.83	65.11
Root mean squared error	142.95	141.18	123.04
Relative absolute error	72.03	71.55	66.72
Root relative squared error	142.54	140.77	122.68

Tabla 4.23: CVParameterSelection Ibk, KStar, LWL con un split de 66 %.

d) meta.MultiScheme

	DStump Ibk	**DStump KStar**	**DStump LWL**
Correlation coefficient	1	1	0.97
Mean absolute error	0	0.66	32.52
Root mean squared error	0	2.57	64.83
Relative absolute error	0.00	0.57	27.81
Root relative squared error	0.00	1.01	25.47

Tabla 4.24: MultiScheme DecisionStump y (Ibk, KStar, LWL) sobre la base de datos de entrenamiento.

	DStump Ibk	**DStump KStar**	**DStump LWL**
Correlation coefficient	0.19	0.13	0.13
Mean absolute error	77.92	85.53	85.53
Root mean squared error	261.36	258.57	258.57
Relative absolute error	64.74	71.06	71.06
Root relative squared error	100.04	98.97	98.97

Tabla 4.25: MultiScheme DecisionStump y (Ibk, KStar, LWL) con un split de 66 %.

En este estudio comparamos el desempeño de 13 algoritmos clasificadores para la predicción del valor del umbral del árbol CF Los resultados indican que el único clasificador fue el provisto por el algoritmo *Ibk:Instance Based Learning* IBk.

En una comparación de los resultados obtenidos con el algoritmo destacado, podremos ver el desempeño en las siguientes gráficas de comparación:

En la gráfica 4.11 se presentan los resultados del promedio de 10 ejecuciones con muestras de tamaño: 50, 100, 150, 200, 250, 300, 350 y 400 datos de el algoritmo BIRCH ejecutado con la aproximación usualmente utilizada, contra el algoritmo Ibk (K-nearest neighbours classifier).

Donde podemos ver una reducción en los tiempos de entre un 5-10 %. esto debido al pequeño número de elementos de datos con los que se cuentan por el momento.

	LeastMedSq	Linear Regression	Multilayer Perceptron	Pace Regression	RBFNetwork	SMOreg	Ibk	Kstar	LWL	Additive Regression	Bagging	CVParameterSelection	MultiScheme
Correlation coefficient	0.3085	0.04	0.2824	-0.092	-0.0162	0.0994		1	0.1514	0.2188	0.1917	-0.0154	-0.3967	-0.3967
Mean absolute error	425.6369	1720.2331	117.9683	5346.6856	121.3942	309.7089		0	78.1457	76.647	90.6331	102.0121	119.5662	119.5662
Root mean squared error	2265.8999	11546.882	246.5362	32505.7067	258.7827	1812.7374		0	262.7136	251.5123	255.178	256.6027	259.142	259.142
Relative absolute error	355.9842	1438.73	98.66	4471.74	101.53	259.03		0	65.3577	64.1	75.8183	85.32	100	100
Root relative squared error	874.39	4455.81	95.14	12543.59	99.86	699.52		0	101.3782	97.06	98.4703	99.02	100	100

Figura 1.6: Clustering de árbol de 111 datos.

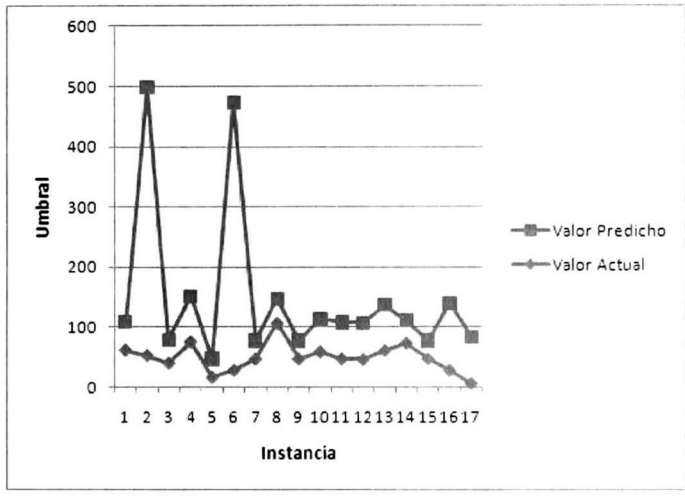

Figura 4.7: Comparación de Resultados de valores obtenidos con un 66%de la base de datos de entrenamiento.

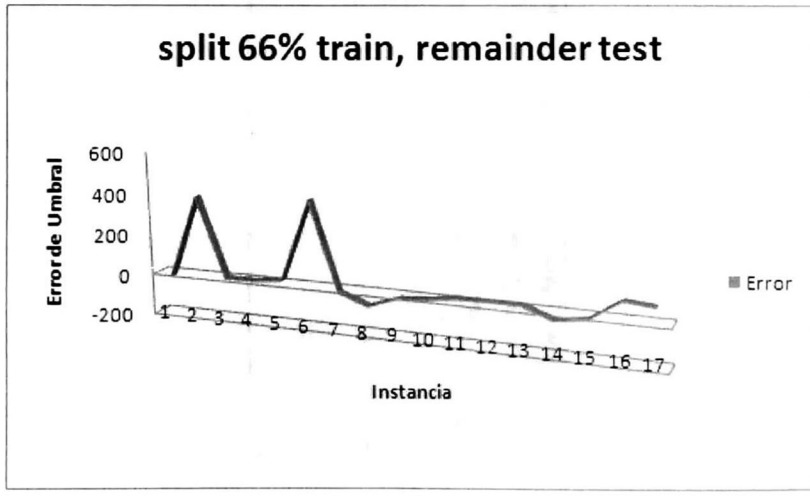

Figura 4.8: Errores en resultados sobre un 66%de la base de datos de entrenamiento.

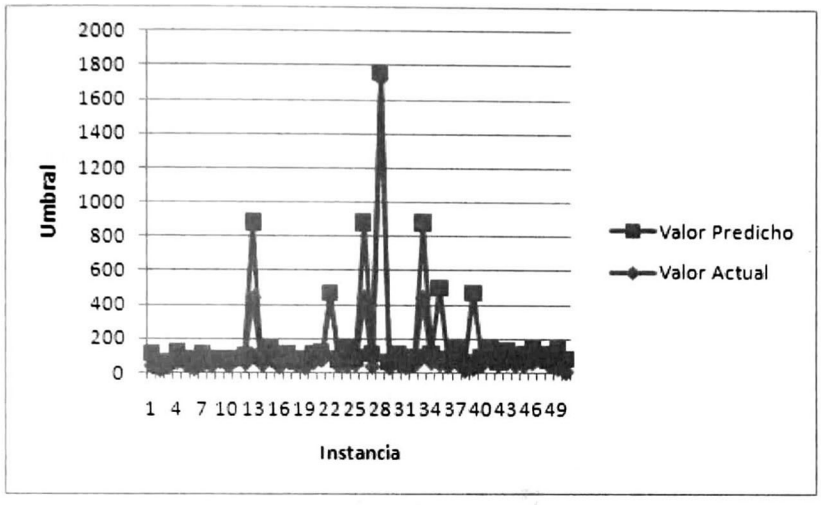

Figura 4.9: Comparación de Resultados de valores obtenidos con una Validación cruzada usando 10 particiones de datos.

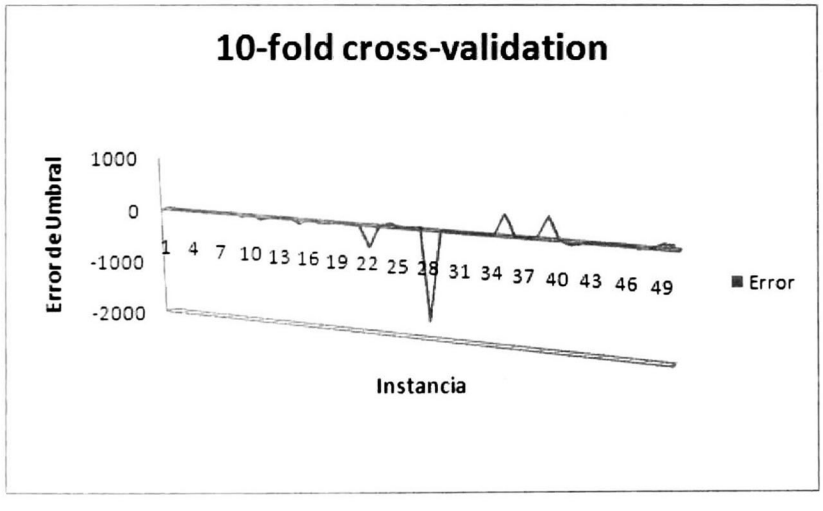

Figura 4.10: Errores en resultados sobre resultados obtenidos con una Validación cruzada usando 10 particiones de datos.

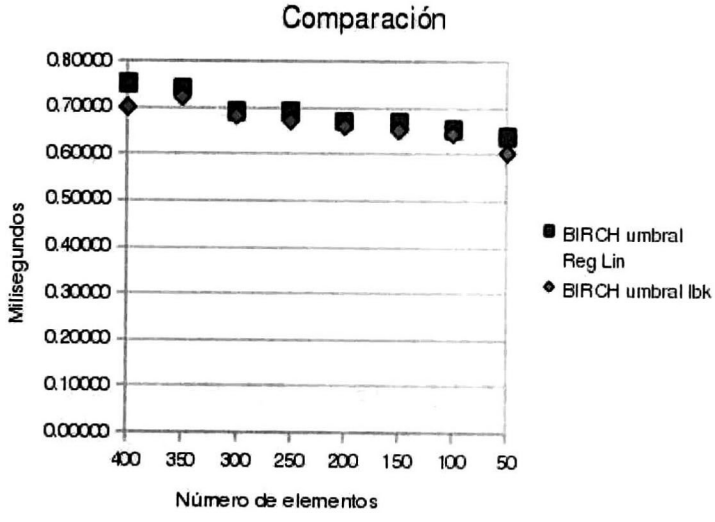

Figura 4.11: Comparación de Resultados.

Capítulo 5

Conclusiones

En esta sección resumiremos la interpretación de los resultados obtenidos, el manifiesto de los objetivos logrados, el aporte y los conocimientos obtenidos de esta investigación. Las preguntas que quedaron sin resolver son presentadas en el Trabajo Futuro.

5.1. Conclusiones

En esta tesis se presenta la solución al problema de clustering de bases de datos espaciales, mediante un algoritmo de clustering elaborado con un árbol de Cluster de Características.

La reconstrucción del árbol tiene varios parámetros iniciales entre ellos el umbral el cual es considerado importante por depender de este el número de veces que el árbol será reconstruido.

Este problema fue abordado tratándolo con una técnica de aprendizaje, en el que el valor del umbral fue caracterizado por variables como el número máximo de nodos permitido en el árbol.

Permitiendo así al algoritmo el poder ejecutarse de manera óptima, reduciendo el tiempo de ejecución al evitar las reconstrucciones.

La implementación fue llevada a cabo en C++ bajo la plataforma de Linux, además se utilizó la herramienta de Minería de Datos Weka para llevar a cabo algunos estudios sobre la caracterización de las variables involucradas.

Este trabajo constituye un primer paso para la mejora del algoritmo de Reducción y Clustering Iterativamente Balanceado utilizando Jerarquías en estudio de bases de datos de cumulos de galaxias, contribuyendo al estudio de la inicialización de parámetros.

El tiempo de ejecución del algoritmo fue reducido, aunque no así la complejidad del

algoritmo.

La principal contribución de este trabajo de tesis es una nuevo método para identificar automaticamente el valor del umbral. Esta aportación es considerada importante ya que BIRCH es en sí un eficiente método para clustering en grandes Bases de Datos, pero no tiene una metodología lógica para estimar el umbral óptimo de absorción de entradas en las hojas.

5.2. Trabajo futuro

En futuras investigaciónes se puede trabajar sobre esta nueva aproximación para continuar proveyendo soluciones a otras áreas de oportunidad en este algoritmo.

Extensión del estudio de los parámetros iniciales de BIRCH, que incluya la integración de más algoritmos de Minería de Datos.

La implementación del método Ibk y la incorporación al código de BIRCH.

Bibliografía

[1] Rafael C. Gonzalez, Richard E. Woods, *Tratamiento digital de imagenes* , Ediciones Diaz de Santos,1996.

[2] I. H. Witten, E. Frank, *Data Mining Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2005.

[3] Hillol Kargupta, Anupam Joshi, Krishnamoorthy Sivakumar, Yelena Yesha,*Trends in Spatial Data Mining capitulo 3 de Libro Data Mining: Next Generation Challenges and Future Directions*,Publicado por AAAI Press, 2004 Universidad de Michigan, 558 paginas.

[4] Martin Ester, Hans-Peter Kriegel, JÃ¶rg Sander*Spatial Data Mining: A Database Approach*,Proc. of the Fifth Int. Symposium on Large Spatial Databases (SSD 97), Berlin, Germany, Lecture Notes in Computer Science Vol. 1262 Pag 47 66, Springer, 1997.

[5] Raghu Ramakrishnan, Johannes Gehrke,*Database Management Systems*,McGraw-Hill Professional, Ed 2003.

[6] Arun K. Pujari *Data Mining Techniques* , Publicado por Orient Blackswan, 2001.

[7] Jiawei Han, Micheline Kamber, *Data Mining: Concepts and Techniques* , Edition 2, Publicado por Morgan Kaufmann ,2006.

[8] Hernandez Valadez Edna, director: Dra. Xiaoou Li Zhang, co-director: Dr. Luis E. Rocha Mier *Algoritmo de clustering basado en entropia para descubrir grupos en atributos de tipo mixto* , Tesis de Maestria en Ciencias en la Especialidad de Ingenieria Electrica del Centro de Investigacion y de Estudios Avanzados del IPN Cinvestav Agosto de 2006.

[9] Tian Zhang, Raghu Ramakrishnan, Miron Livny,*BIRCH: A New Data Clustering Algorithm and Its Applications*,Publisher Springer, Data Mining and Knowledge Discovery, Volume 1, Number 2, 1997 , pp. 141-182(42).

[10] Tian Zhang, Raghu Ramakrishnan, Miron Livny,*BIRCH: An Efficient Data Clustering Method for Very Large Databases*,Publisher Springer, SIGMOD Conference 1996: 103-114.

[11] Marco Frailis, Alessandro De Angelis, Vito Roberto, *Data Management and Mining in Astrophysical Databases*, EURASIP journal on Applied Signal Processing, 2005.

[12] E.D. Feigelson, G.J. Babu, *Statistical Challenges in Modern Astronomy*, PHYSTAT2003,SLAC,Stanford,California,September 8-11,2003.

[13] Ray Norris, Heinz Andernach, Guenther Eichhorn, Francoise Genova, Elizabeth Griffin, Robert Hanisch, Ajit Kembhavi, Robert Kennicutt, Anita Richards, *Astronomical Data Management*, Highlights of Astronomy, 2006.

[14] S.G.Djorgovski,C.Donalek,A.Mahabal,R.Williams,A.J.Drake,M.J.Graham,E.Glikman, *Some Pattern Recognition Challenges in Data-Intensive Astronomy*, Proceedings of the 18th International Conference on Pattern Recognition (ICPRÅ106).

[15] G. Jacoby, J. Barnes, A.S.P.,*Analysis of Digital POSS-II Catalogs Using Hierarchical Unsupervised Learning Algorithms*, in Astronomical Data Analysis Software and Systems V, eds.

[16] R. Shaw,*Clustering Analysis Algorithms and Their Applications to Digital POSS-II Catalogs*, in Astronomical Data Analysis Software and Systems IV, eds. et al., A.S.P Conf. Ser., 77, 272.

[17] A. Miller, M. Coe,*Star/galaxy classification using Kohonen self-organizing maps*,Monthy Notices Royal Astron. Soc., 279, 293, 1996.

[18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu,*A density-based algorithm for discovering clusters in large spatial databases*, KDD'96.

[19] M. Ankerst, M. Breunig, H. -P. Kriegel, and J. Sander,*Optics: Ordering points to identify the clustering structure*, SIGMOD,1999.

[20] A. Hinneburg,*D.l A. Keim: An Efficient Approach to Clustering in Large Multimedia Databases with Noise*, KDD,1998.

[21] W. Wang, Yang, R. Muntz,*STING: A Statistical Information grid Approach to Spatial Data Mining*, VLDB,1997.

[22] S. Guha, R. Rastogi, and K. Shim,*Rock: A robust clustering algorithm for categorical attributes*, In Proceedings of IEEE Conference on Data Engineering, 1999.

[23] S. Guha, R. Rastogi, and K. Shim,*Cure: An efficient clustering algorithm for large databases*, SIGMOD,98.

[24] G. Karypis, E.-H. Han, and V. Kumar,*CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling*. COMPUTER, 32(8): 68-75, 1999.

[25] Jorg Sander, Martin Ester, Hans-Peter Kriegel, Xiaowei Xu,*D*ensity-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications, Data Mining and Knowledge Discovery, Vol. 2, No. 2, Kluwer Academic Publishers, pp. 169-194, 1998.

[26] M. W Berry, M. Browne, *Lecture notes in data mining*, World Scientific, 2006.

[27] P. Rob, C. Coronel, *Database Systems: Design, Implementation And Management*, Cengage Learning, 2003.

[28] S. Mitra, T. Acharya, *Data mining: multimedia, soft computing, and bioinformatics*, John Wiley and Sons, 2003.

[29] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern classification*, Wiley, 2001.

[30] D. J. Hand, H. Mannila, P Smyth, *Principles of data mining*, MIT Press, 2001.

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Determinación de Umbrales en Arboles Tipo CF Utilizando
Aprendizaje Supervisado para Algoritmos de Clustering Jerárquico

del (la) C.

Mayra Teresa TREJO HERNÁNDEZ

el día 28 de Agosto de 2009.

Dr. Luis Ernesto López Mellado
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Félix Francisco Ramos Corchado
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Mario Angel Siller González
Pico
Investigador CINVESTAV 2A
CINVESTAV Unidad Guadalajara

Dr. Andrés Méndez Vásquez
Investigador CINVESTAV 2A
CINVESTAV