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

Algoritmo de agrupamiento basado en Densidad Aplicado a la Búsqueda de Subestructuras en Cúmulos de Galaxias

Tesis que presenta:

Arturo Raymundo Avilés

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.

Algoritmo de agrupamiento basado en Densidad Aplicado a la Búsqueda de Subestructuras en Cúmulos de Galaxias

Tesis de Maestría en Ciencias
Ingeniería Eléctrica

Por:

Arturo Raymundo Avilés

Licenciado en Ciencias de la Computación

Universidad Autónoma de Yucatán 2002-2007

Becario de Conacyt, expediente no. 212737

Directores de Tesis

Dr. Mario Angel Siller González Pico

Dr. Ricardo Vilalta

CINVESTAV del IPN Unidad Guadalajara, Agosto de 2009.

Tesis de Maestría en Ciencias en Ingeniería Eléctrica

Presentada por:

Arturo Raymundo Avilés

Para obtener el grado de:

Maestro en Ciencias

Con especialidad en:

Ingeniería Eléctrica

Dr. Ricardo Vilalta López Dr. Mario Angel Siller González Pico Director de Tesis	Dr. Luis Ernesto López Mellado Sinodal
Dr. Félix Francisco Ramos Corchado Sinodal	Dr. Andrés Méndez Vásquez Sinodal

10 de Agosto de 2009

Agradecimientos

A Dios por acompañarme en cada momento de mi vida dándome Salud, Fortaleza, Esperanza y luz a mi vida.

A mis Padres, A mis Hermanas, A mis Familiares, y mis Amigos, Gracias por su comprensión, apoyo y amistad.

A mis Asesores de tesis Dr. Ricardo Vilalta López y Dr. Mario Angel Siller González Pico gracias por el apoyo, tiempo y dedicación en la elaboración de esta tesis.

También quiero agradecer al Dr. Heinz Andernach y el Dr. Cesar Careta del Departamento de Astronomía de la Universidad de Guanajuato por su valiosa participación en la primera etapa de este estudio y por preparar la serie de datos de los cúmulos de galaxias.

Al CINVESTAV por permitirme realizar mis estudios de maestría.

Al CONACYT por el apoyo económico con la beca número para la realización de esta tesis.

Becario de CONACYT No. 212737

Resumen

La *minería de datos espacial* es un subcampo del área de minería de datos que trabaja con datos que tienen una posición absoluta, una posición relativa y una figura geométrica que las representan. Esta tesis presenta el trabajo de investigación y desarrollo de un algoritmo de minería de datos espacial, partiendo de un algoritmo de clustering basado en densidad llamado DBSCAN.

El algoritmo base fue perfeccionado utilizando árboles R*, el cual es un método de acceso espacial para el manejo de los datos, también se propuso una modificación en la manera de realizar la expansión del cluster por medio de puntos borde.

Al concluir este trabajo se logró una mejora al momento de obtener el clustering generado por el algoritmo, en el tiempo de ejecución del algoritmo este se redujo en comparación con el algoritmo base, lo que al momento de utilizar este algoritmo con grandes bases de datos se obtiene una reducción en el tiempo de procesamiento.

Palabras clave: minería de datos espacial, DBSCAN, Cúmulos de Galaxias.

Abstract

Spatial data mining is a subfield of data mining that works with spatial data. These spatial data have an absolute position, a relative position, and a geometric shape that are uased for their representation.

This thesis describes the development of a data mining algorithm based on a density based clustering algorithm, DBSCAN.

The algorithm was developed using R* trees, which is a method for handling spatial data. We proposed a change in the way to perform the cluster expansion by means of border points.

As conclusion of this work, we show an improvement in performance time as compared to the DBSCAN algorithm.

Keywords: Spatial Data Mining, DBSCAN, galaxies clusters.

Indice general

Indice de tablas

Indice de figuras

Capítulo 1

Introducción

1.1. Minería de Datos

El término Minería de Datos se refiere a: procesar datos para identificar patrones, es una disciplina con base en la estadística, el reconocimiento de patrones y machine learning, entre otras.

Gracias a los avances en electrónica y computación, en las recientes décadas han surgido grandes bases de datos en diversas áreas: el comercio, la banca, astronomía, física de partículas, química, medicina, departamentos de gobierno, etc., la cuales contienen información que sólo puede ser extraída mediante las técnicas especiales que provee la minería de datos. Esta actividad puede parecer, a primera vista, un simple análisis exploratorio de datos, pero al observar detalladamente la composición de estos repositorios, surgen varias diferencias.

① **Tamaño de las Bases de Datos.**

La cantidad de datos a manejar en minería de datos llega a ser de muchos millones e incluso billones de registros. Hoy es común hablar de tamaños en gigabytes o incluso terabytes. Por ejemplo, se estima que el proyecto Earth Observing System de la NASA generará cerca de 50 gigabytes de datos por hora.

La necesidad de procesar una gran cantidad de datos acarrea algunos problemas: Una

Figura 1.1: Aplicación de la minería de datos en diferentes áreas

base de datos gigantesca no cabe en la memoria principal de una computadora, lo que hace necesario contar con técnicas de análisis secuenciales. En general, se cuenta con gran número de registros o de variables, aumentando la cantidad de dimensiones a considerar. Los datos pueden estar almacenados en distintos archivos interrelacionados y/o distribuidos, para cuyo procesamiento se necesitan métodos de clustering.

② **Datos contaminados.**

Tradicionalmente en estadística se trabaja con datos limpios, en donde se utilizan técnicas para identificar y eliminar datos outliers (ruido). En minería de datos, sin embargo, no es factible eliminar dichos datos. Teniendo grandes cantidades de datos, y considerando que provienen de análisis de marketing, transacciones financieras, recursos humanos, etc. siempre existen datos con cierta invalidez, que pueden ser vitales para descubrir patrones débiles.

③ **Observaciones dependientes y sesgo de selección.**

En grandes conjuntos de datos es bastante difícil cumplir con una de las premisas fundamentales del análisis estadístico: los datos deben ser muestreados independientemente y deben pertenecer a la misma distribución. En minería de datos es usual trabajar con población no-estacionaria, es decir, población que cambia en el tiempo. Para detectar los cambios en la población es necesario llevar un registro del tiempo en que se tomaron los datos. Incluso a veces es necesario procesar los datos en tiempo real. El sesgo casi siempre está presente en una selección, pero en grandes volúmenes de datos es casi

totalmente inevitable debido a que no se puede asegurar ni revisar que las mediciones sean aleatorias y en muchos casos sólo son muestreos convenientes o sólo de los casos posibles, como por ejemplo, un banco sólo puede tener los datos de las personas que aceptan abrir una cuenta corriente, no de personas que no son clientes, lo que produce un sesgo en el muestreo. Para poder realizar un buen análisis es necesario contar con modelos que tomen en cuenta el mecanismo de selección.

④ **Búsqueda de patrones.**
Aunque se cuenta con mecanismos computacionales para analizar datos, antes que todo es necesario encontrar una manera de enseñarle a la computadora qué es un patrón interesante o cómo identificar estructuras. Aquí es esencial estar familiarizado con el dominio del problema y generar acercamientos ad-hoc a los problemas.

⑤ **Falsas relaciones.**
La búsqueda de patrones puede arrojar falsos resultados debido a la gran cantidad de datos analizados. Para minimizar este hecho existen diversas estrategias: restringir la familia de modelos, optimizar la penalización de la función de bondad de ajuste, o imponer criterios de selección de patrones más fuertes. Aunque, después de aplicarlas, los patrones identificados como más interesantes siempre deben ser presentados al experto en el tema quien debe decidir su aplicabilidad.

⑥ **Datos no numéricos.**
Las bases de datos actuales no almacenan sólo números, sino también imágenes, audio, texto y datos geográficos. Los análisis descritos anteriormente deben aplicarse a este tipo de datos al igual que a los datos numéricos. Es por ello que el Web Mining se ha convertido en una sub-área de minería de datos.

En la actualidad existen varias definiciones para el concepto de minería de datos, pero la esencia de éstas se fundamenta en el concepto de escarbar en la información almacenada para descubrir elementos de utilidad desde grandes cantidades de datos almacenadas, con el objetivo de detectar de patrones de comportamiento consistentes, o relaciones entre los diferentes campos de una base de datos para aplicarlos a nuevos conjuntos de datos.

1.2. Machine Learning

El aprendizaje es un área clave para el desarrollo de una sociedad, que abarca una gama tan amplia de procesos que es difícil definir con precisión.

El diccionario de la real academia española define aprendizaje como:

1. Acción y efecto de aprender algún arte, oficio u otra cosa.
2. Tiempo que en ello se emplea.
3. Adquisición por la práctica de una conducta duradera

El ser humano está constantemente planificando, desde los razonamientos que hace sobre cómo se va a organizar el día, qué tareas tiene que hacer y cómo las va a acometer, hasta el establecimiento de planes específicos de cualquier actividad profesional, (como concluir los estudios de postgrado). Por ejemplo, las empresas elaboran su plan de acción en el que se define la programación de actividades para satisfacer las metas empresariales fijadas; en manufacturación es necesario determinar las acciones que transforman las materias primas en los productos elaborados; los abogados requieren de un plan de defensa de sus clientes, etc.

En general, planificar suele ser una tarea difícil debido principalmente a las siguientes cuestiones:

- Nuestra visión del mundo es incompleta.
- El mundo cambia constantemente.
- Nuestro modelo del mundo falla muchas veces.
- Las acciones tardan en ejecutarse.
- Algunas metas pueden tener efectos contrapuestos.

- Los planes no siempre son validos.

- No todos los planes tiene la calidad deseada.

Sin embargo, estas dificultades pueden ser solucionadas, gracias a la capacidad de aprendizaje del hombre que hace posible una continua adaptación al mundo. Hoy en día se ha considerado a la computadora como una máquina tonta, que solo realiza aquello que se haya programado previamente. Este modo de operar elimina una de las cualidades principales de la inteligencia, que consiste en un comportamiento que no ha sido programado, sino que se ha producido como consecuencia de una manipulación inteligente de los conocimientos y experiencias.

El comportamiento inteligente se caracteriza por no producir siempre los mismos resultados. Según las circunstancias y factores objetivos y subjetivos, ante una misma situación, los seres inteligentes no toman la misma decisión. Machine Learning es un área de investigación que ha tratado de emular las actividades del aprendizaje humano con sistemas computacionales. Esto la ha llevado a ser un área importante de investigación por los últimos 25 años. Actualmente Machine learning se encuentra en la intersección de varios campos del conocimiento como la inteligencia artificial, la probabilidad y estadística, psicología y filosofía.

En muchas ocasiones el campo de actuación del Machine Learning se solapa con el de la Estadística, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el Machine Learning se centra más en el estudio de la Complejidad Computacional de los problemas.

La meta principal de Machine learning es lograr emular las habilidades cognitivas de los humanos con un sistema computacional. Entre ellas se encuentran la habilidad para generalizar reglas a partir de ejemplos específicos así como el abstraer características de una serie de objetos y determinar cuáles son los más importantes.

1.3. Clasificación de los sistemas de aprendizaje

Dependiendo de si durante el entrenamiento se utiliza o no la información que se pretende predecir:

Aprendizaje supervisado:
Los sistemas que aplican este tipo de aprendizaje conocen la categoría que quieren predecir y guían el aprendizaje en la dirección adecuada haciendo comprobaciones periódicas de la calidad (supervisando) de la solución calculada hasta el momento, es uno de los problemas más estudiados en Inteligencia Artificial. En particular, el objetivo de cualquier algoritmo de aprendizaje supervisado es construir un modelo de clasificación a partir de un conjunto de datos de entrada, denominado conjunto de entrenamiento, que contiene algunos ejemplos de cada una de las clases que pretendemos modelar.

Aprendizaje no supervisado:
En este caso, a los sistemas no se les suministra la categoría de cada ejemplo de entrenamiento y por tanto no pueden supervisar el conocimiento inferido.

Aprendizaje por refuerzo:
El algoritmo aprende observando el mundo que le rodea. Su información de entrada es el feedback o retroalimentación que obtiene del mundo exterior como respuesta a sus acciones.

Aprendizaje multi-tarea:
Métodos de aprendizaje que usan conocimiento previamente aprendido por el sistema de cara a enfrentarse a problemas parecidos a los ya vistos.

1.4. Organización de la tesis

Esta tesis está organizada en los siguientes capítulos como sigue:
Capítulo 2.- Estado del arte de trabajos relacionados del área de Astrofísica y conceptos utilizados en las secciones posteriores.

Capítulo 3.- Propuesta del algoritmo de clustering DBSCAN++.

Capítulo 4.- Resultados obtenidos.

Capítulo 5.- Conclusiones de la tesis y trabajo futuro.

Capítulo 2

Estado del Arte

2.1. Introducción

En la actualidad muchos estudios se llevan a cabo acerca del análisis de datos astronómicos, podemos decir que esta es un área nueva en la que muchos investigadores de diferentes áreas (desde astrónomos, astrofísicos, estadísticos, e inclusive investigadores del área de computación) se han interesado en ella.

Diferentes enfoques y necesidades han sido descubiertas en el área de la astrofísica, uno de los principales problemas que hoy en día encuentran en la astrofísica es la gran cantidad de información que se obtiene día con día, y la irregular forma en que los datos son obtenidos debido a los diferentes tipos de medición así como los diferentes tipos de instrumentos que no logran una adecuada estandarización en el área. Debido a eso el uso de técnicas de minería de datos son necesarias para maximizar la extracción de la información de la creciente cantidad de datos obtenidos en el área. Tal como menciona [ ]

"En el dominio de la Astrofísica, el papel de la minería de datos es ayudar a los investigadores a construir o verificar nuevos modelos físicos basados en los datos observados"

Pero no todo es tan sencillo ya que como era de esperarse surgen varios problemas en el camino, como el no poder a primera mano aplicar estos métodos debido a la diversidad de los componentes espaciales y temporales por la diversidad de los instrumentos proyectos que

no tienen comunicación alguna aunque estén enfocados en proyectos muy similares lo que produce datos no tan homogéneos.

En algunos casos todavía no se han desarrollado por completo la combinación de metodologías que los astrónomos necesitan. Pero hay algunos campos como por ejemplo: ***"An automated star-galaxy classification in complex and heterogeneous panoramic imaging data sets, y an automated, iterative, dynamical classification of transient events detected in synoptic sky surveys"***, en los cuales según [ ] son problemas que ofrecen una buena oportunidad para producir colaboraciones entre los astrónomos y científicos de la computación.

También una manera en la que han reaccionado los astrónomos son los observatorios virtuales (OV) los cuales son una colección de archivos de datos interactivos que utilizan la red para elaborar un ambiente de búsqueda científica para producir trabajos en conjunto, actualmente hay una gran cantidad de observatorios virtuales pero los principales son el Observatorio virtual Nacional en los Estados Unidos y el Observatorio virtual Astrofísico en Europa.

Algunos de los resultados que se han logrado con los OV son progresos significativos en la administración de datos, pero ha habido muy pocos progresos en el desarrollo de la exploración de datos altamente escalables y en herramientas de análisis para obtener resultados de las grandes cantidades de datos obtenidos. Aun cuando actualmente existen muchos programas de software en el área de la minera de datos casi no se tienen herramientas que verdaderamente soporten los conjuntos de datos del orden de los Terabytes o incluso del orden de Petabytes de información. Incluso según [ ] esto ha provocado la escasez de resultados dentro del área de la astronomía.

Se han dado avances y aunque los astrónomos ya tienen un conocimiento, necesitan poner más énfasis en conocer y aplicar principios establecidos de inferencia estadística como por ejemplo pruebas de hipótesis y estimación de parámetros, inferencia paramétrica, para poder tener más éxito en el desarrollo de sus proyectos y no seguir basándose en métodos como la transformada de Fourier desarrollada en 1807, regresión de mínimos cuadrados formulado en 1924, tal y como lo menciona [ ].

Los datos provistos por los observatorios virtuales son generalmente heterogéneos y distribuidos, es por eso que la minería de datos es de gran ayuda para resolver algunos de los problemas que se presentan al momento de analizar y buscar patrones y relaciones en los datos y a la hora de construir modelos.

2.2. La Minería de Datos en el Campo de la Astronomía

De acuerdo con [ ] actualmente en la minería de datos hay dos principales tipos de modelos, los descriptivos y los predictivos.

Los modelos descriptivos ***"detallan"*** patrones en los datos y son usados para crear Clusters ó grupos de entre los datos.

Los modelos predictivos son usados para pronosticar valores explícitos basados sobre determinados patrones sobre resultados conocidos. De igual manera [ ] llega a clasificar los escenarios de la minería basada en eventos en cuatro categorías:

- Acontecimiento conocidos / algoritmos conocidos.
- Acontecimiento conocidos / algoritmos Desconocidos.
- Acontecimiento desconocidos / algoritmos conocidos.
- Acontecimiento desconocidos / algoritmos desconocidos.

Se identifican 3 clases asociados a los escenarios

Asociación espacial: Identificación de objetos astronómicos en la misma locación en el cielo.

Asociación temporal: Identificar eventos que ocurran durante el mismo periodo de tiempo.

Asociación de coincidencia: Uso de técnicas de Clustering para identificar eventos que estén co-localizados dentro de espacio multidimensional.

Es conocido que una fracción significativa de todas las galaxias han estado involucrados en una interacción con otra galaxia en algún tiempo de su pasado nos menciona [ ]. La frecuencia de estas interacciones todavía no puede ser correctamente determinada empíricamente.

En los últimos tiempos de acuerdo con [ ] se ha visto una revolución en la manera en que los astrónomos usan los datos, se han creado centros de datos como el Astrophysics Data System y NASA/IPAC Extragalactic Database, y estos han transformado la manera como los astrónomos accesan a la literatura y llevan a cabo sus investigaciones, acceso a datos electrónicos y publicaciones ha traído capacidades de investigación de primera línea a diversos lugares en el mundo desarrollado, y un número creciente de archivos de los mayores telescopios está poniéndose en el dominio público.

El vigoroso desarrollo internacional de los observatorios virtuales, la revolucionaria liberación de datos de proyectos astronómicos y la rápida difusión de los resultados obtenidos hecho posible por los diversos journals, es uno de los mayores éxitos a últimas fechas dentro del campo de la astronomía.

Todos éstos posicionan a la astronomía como un modelo para otras ciencias para cómo puede usarse tecnología para acelerar la calidad y efectividad de la ciencia.

Pero no todos los puntos son positivos, también hay algunas negativos y entre ellos [ ] menciona que la administración de los datos astronómicos es aun inadecuada en perjuicio de la astronomía. Por ejemplo, existen presiones internacionales para hacer las bases de datos de acceso libre pero sin que todavía se solucionen los puntos legales.

Se habla según [ ] de 2 grupos de astrónomos, los que viven en un mundo desarrollado, los cuales tienen acceso a instantáneo a los datos y journals y en contra sus colegas en vías de desarrollo aun confían en fotocopias de primeras ediciones.

También [ ] habla de la existencia de un cuello de botella entre los journals y los centros de datos, es decir una gran parte de los más importantes datos publicados en la mayoría de

los Journals nunca aparece en los centros de datos.

Muchas veces incluso en algunas instituciones astronómicas la administración de los datos no es tomada de una manera seria, [ ] nos da el ejemplo de que si un astrónomo pone disponible una gran base de datos, no se le da el reconocimiento que es dado al autor de un paper.

Entre los principales tareas de la alianza internacional observatorios virtuales está definir la interoperabilidad entre los distintos estándares de los OV como por ejemplo los registros de los recursos, la capa de acceso a los datos, la descripción del contenido y el modelo de datos. La mayoría de los estándares para los OV están ya listas.

Aun en los observatorios virtuales hay todavía varios puntos críticos que necesitan solución como por ejemplo: Control de la calidad, certificación, control de versiones, pautas de propiedad la intelectual.

La definición de los centros de datos de acuerdo con [ ] puede ir desde ***"un lugar en el que se distribuyen datos observacionales"*** a un servicio en el cual aunque también se distribuye información, herramientas y servicios de valor agregado muchos equipos están tratando de proveer datos y servicios compatibles con otros OV en su dominio de experiencia.

Entre los principales servicios que están siendo interpretados por los OV se encuentran:

- Archivos de observaciones
- Servicios y herramientas de valor agregado
- Servicios teóricos
- Programas de software para el análisis de datos
- Servicios específicos
- Ambientes de Investigación

Uno de los principales objetivos a futuro de los proyectos de los Observatorios Virtuales según [ ] es crear una comunidad de observatorios virtuales proveedores de servicios.

Como en varios campos de la ciencia, la aparición del Internet ha dado un gran apoyo a la astronomía, esto según [ ] ha sido algo que revolucionó la manera en como los astrónomos trabajan, aun con aquellos que estaban acostumbrados a trabajar de una manera más aislada, por lo que ahora con el desarrollo de los Observatorios Virtuales se da un gran impulso a los observatorios y los equipos de astrónomos a hacer sus datos y servicios disponibles para la comunidad completa, es por eso que muchos grupos irán en cierto grado convirtiéndose en centros de datos, esto hará que se comparta conocimiento entre las diversas comunidades de astrónomos.

Entre los diversos métodos que se emplean en el reconocimiento de patrones en la astronomía van desde los simples divisores en el espacio de parámetros o aplicaciones de machine learning como redes neuronales artificiales o arboles de decisión, cuando se habla de imágenes normalmente cada imagen es tratada independiente de otras aun si existe información útil en alguna imagen vecina, la cual generalmente tiene propiedades muy similares también, en un estudio según [ ], la homogeneidad de la clasificación morfológica es importante, y tiene que ser logrado por la normalización de los atributos del objeto introducido en el clasificador. En los últimos años se han realizado algunos trabajos con métodos de clasificación no supervisados entre los que se encuentran Automated Source Classification Using a Kohonen Network por P. Maehoenen, P. Hakala, Star/galaxy classification using Kohonen self-organizing maps por A. Miller, M. Coe, Analysis of Digital POSS-II Catalogs Using Hierarchical Unsupervised Learning Algorithms por J. Yoo lo cual nos muestra el interés que hay con el uso de estas herramientas en el campo. [ ] también nos habla de que las mismas técnicas de clasificación son usadas para exploraciones más detalladas de inspecciones digitales grandes de cielo y otros conjuntos astronómicos de datos.

Otro punto es que actualmente con la diversidad de datos que existen, hay también problemas de inconsistencia que se dan cuando la misma fuente astronómica es usada varias veces en condiciones diferentes, aplicándole diferentes filtros y se obtienen muchas clasificaciones independientes los cuales no son consistentes unas con otras.

SKICAT es un sistema de análisis de datos astronómicos que incorpora técnicas de machine learning para clasificar objetos (principalmente estrellas y galaxias) en clases de una manera útil para los astrónomos es parte del estudio del cielo del observatorio del palomar.

El trabajo se hace con una serie de datos de entrenamiento los cuales ya se encuentran clasificados por astrónomos, y estos sirven para que aplicando los algoritmos de aprendizaje supervisado clasifique y otorgue clases a los nuevos datos que aun no han sido clasificados.

En cambio para el aprendizaje no supervisado no se requieren datos que hayan sido clasificados con anterioridad y este es el caso del *conceptual clustering* el cual clasifica los datos que no poseen una etiqueta previa.

Este [ ] es uno de los trabajos que pone especial énfasis en la separación de estrellas y galaxias utilizando machine learning que es un subproblema de la clasificación de objetos de en el cielo.

Aprendizaje no supervisado tiene la ventaja sobre el aprendizaje supervisado que no clasifica de acuerdo a pensamiento de un astrónomo en particular, dejando únicamente un claro y objetivo proceso, además puede identificar nuevas clases donde un astrónomo no encontraba ningún patrón en especial.

Varios algoritmos de aprendizaje no supervisado han sido aplicados a este tipo de problema pero uno de los que ha ofrecido mejores resultados es el Cobweb/95.

Cobweb/95 es un sistema de clustering conceptual que organiza las observaciones para maximizar la habilidad de conclusión utiliza el más simple esquema de codificación es decir los vectores característicos los cuales son descripciones de valor de los atributos.

Cuando los humanos observan objetos en un ambiente entonces normalmente agrupan los objetos en clases encontrando características en común y diferencias entre las clases, a través de estos métodos es como nos ayudamos a entender los objetos.

La manera en que Cobweb trabaja es organizando las observación incrementalmente dentro de un árbol de clasificación, cada nodo en un árbol de clasificación representa una clase

y es etiquetado en un concepto probabilístico que resume la distribución del valor de los atributos de los objetos clasificados debajo de un nodo.

Los nodos terminales son la clases más especifica que cubre ejemplos individuales que han sido previamente observados y la raíz representa los conceptos más generales que representan los datos por completo, tal y como lo explica en [ ].

En un ejemplo de la aplicación del algoritmo Cobweb/95 toman una base de datos de más de 33 mil datos, en la que cada dato se consideran 10 atributos, pero no se toman encuentra otros en [ ] como el color ya que sería un atributo que sesgaría considerablemente la clasificación.

Otro tipo de algoritmo de aprendizaje no supervisado utilizado para la clasificación que es usado en la astronomía son los mapas auto organizados (Self Organizing Map) [ ] un tipo de redes neuronales artificiales, los cuales consisten de una capa de entrada y neuronas ocultas en la capas intermedias, entonces la red aprende asociando diferentes tipos de patrones de entrada con diferentes agrupaciones de los nodos ocultos, las cuales son entrenadas en grupos que responden a diferentes patrones de entrada.

En especial los mapas auto organizados de Kohonen fueron utilizados en [ ] para la clasificación de estrellas y galaxias, en los cuales primero fue usado un subconjunto de datos de entrenamiento para la calibración del mapa, y en la cual el nodo más cercano el cual corresponde o se empareja con cada objeto entonces es clasificado. Y en el caso de que un nodo se emparejara con más de un objeto, se media el menor error asociado y a ese nodo se asocia el objeto. Es por eso que la habilidad de usar una serie de calibración, menor que el conjunto de entrenamiento, reduce el número de objetos que deben ser clasificados para el entrenamiento y solo son necesarios unos pocos cientos de objetos clasificados manualmente para lograr un buen rendimiento según [ ].

Para diversos propósitos cuando se desea separar los datos en clases [ ] define 2 puntos que son importantes a la hora de usar algoritmos de clasificación: la completez y la contaminación los cuales son definidos de la siguiente manera:

- **Completez:** Es el porcentaje de datos en este caso galaxias que son correctamente identificados.
- **Contaminación:** Es el porcentaje de datos que son clasificados de una manera incorrecta.

Un clasificador ideal debe de poseer un 100 porciento de completez y 0 porciento de contaminación.

2.3. Algoritmos de clustering en la Minería de Datos Espacial

Los métodos clustering de Minería de Datos Espacial son aplicados para extraer conocimiento interesante y regular. Estos métodos pueden ser usados para entender los datos espaciales, descubrir relaciones entre datos espaciales y no espaciales, reorganizar los datos en bases de datos espaciales y determinar sus características generales de manera simple y concisa[ ].

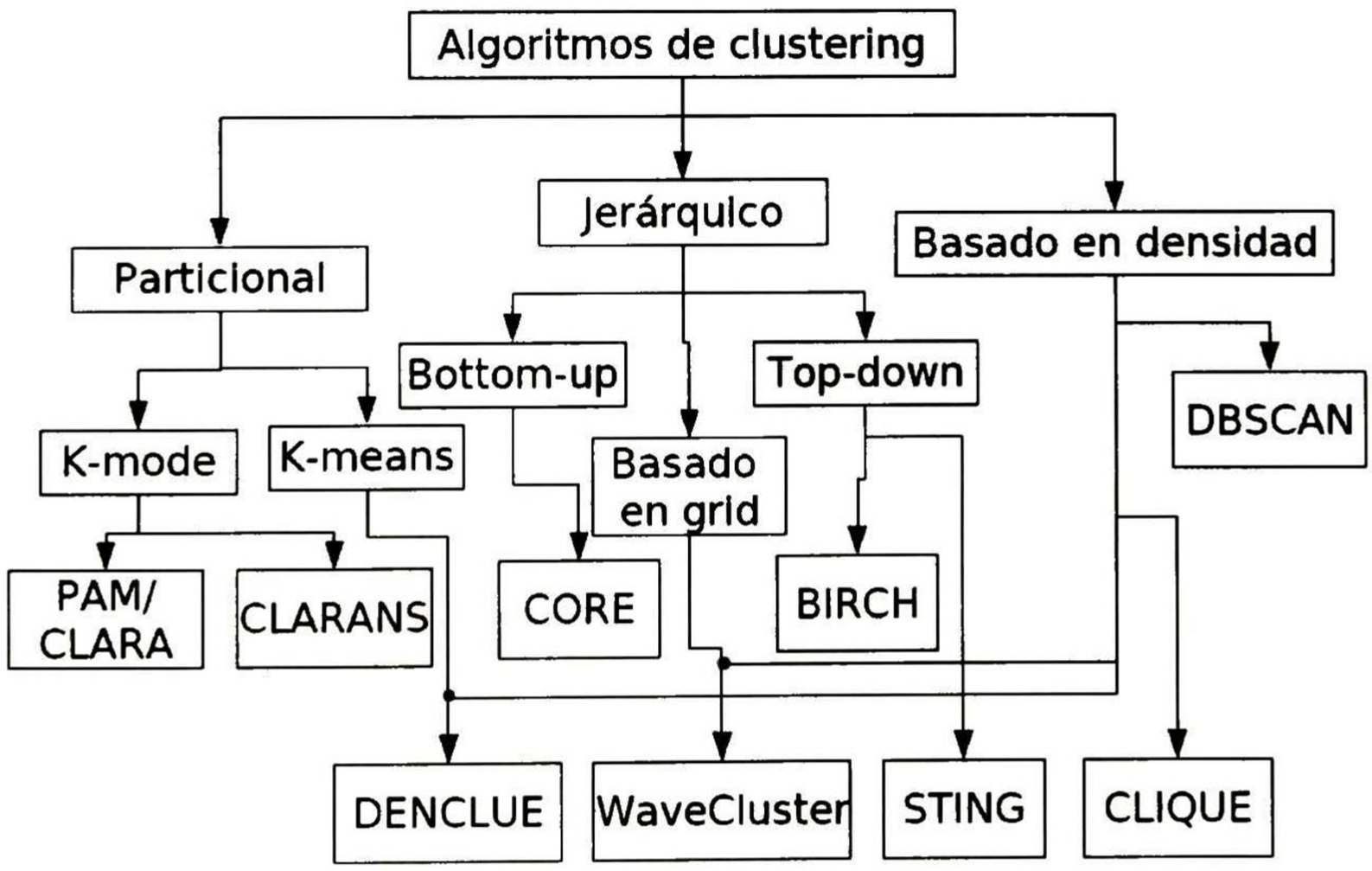

Figura 2.1: Clasificación de los diferentes métodos de agrupamiento

Como se muestra en la figura 2.1 los algoritmos de clustering se pueden agrupar en 3 principales corrientes:

- Particionales
- Basados en Densidad
- Jerárquicos

En las siguientes secciones se describirán estas 3 corrientes con los trabajos representativos de cada una.

2.3.1. Particionales

En esta corriente se encuentra el k-means [ ] el cual es un método de análisis de clusters que apunta a la partición de n observaciones en k grupos en el que cada observación pertenece al cluster con la media más cercana .

El algoritmo PAM (Partitioning Around Medoides) fue desarrollado por Kaufman y Rousseeuw, en éste se asume que existen n objetos, y que para encontrar K clusters, se determina un objeto representativo para cada agrupación. Tal objeto representativo, es un punto centralmente localizado dentro de un grupo, denominado medoide. Después de seleccionar los medoides de K, el algoritmo intenta analizar todos los pares posibles de objetos, tales que, cada objeto no seleccionado es agrupado con el medoide más similar. La calidad de la medida de agrupamiento se calcula para cada combinación.

Otro método para la minería de datos espacial es el CLARANS: (Clustering Large Applications based on RANdomized Search)[ ], este fue desarrollado para el análisis de agrupamientos mediante medioides en 2002.

2.3.2. Basados en Densidad

A este siguió el algoritmo DBSCAN: Density Based Spatial Clustering of Applications with Noise [ ], este algoritmo está fundamentado en la localización de los objetos basándose 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.
Otro algoritmo dentro de esta corriente es: OPTICS: Ordering Points To Identify the Clustering Structure [ ]. El cual realiza un ordenamiento previo de la base de datos y realiza una representación gráfica para un mejor entendimiento, y por ultimo, GBSCAN: Generalized Density-Based Spatial Clustering of Applications with Noise[ ] el cual basado en DBSCAN hace una generalización de la noción de punto de densidad.

2.3.3. Jérarquicos

En esta sección se listan dos principales algoritmos: BIRCH: Balanced Iterative Reducing and Clustering using Hierarchies [ ] es un algoritmo incremental que utiliza estructuras específicas denominadas clusters de características para ejecutar el clustering.
CURE: An efficient clustering algorithm for large databases [ ] utiliza un conjunto representativo por cada cluster en lugar de un único punto, se caracteriza por ser de tipo aglomerativo y tambien ir añadiendo grupos hasta obtener k grupos.

2.4. Observatorios Virtuales Internacionales

Un observatorio virtual es una colección de archivos de datos interactivos y de herramientas software que utilizan Internet para elaborar un ambiente de búsqueda científica en el que

los programas de investigación astronómica puedan ser conducidos.

De la misma forma que un observatorio real es un conjunto de telescopios, cada uno con una colección única de instrumentos astronómicos; el observatorio virtual consiste en un conjunto de centros de datos, cada uno con una colección única de datos astronómicos, productos de software y capacidades de cálculo.

Existen diferentes proyectos de observatorios virtuales en el mundo, principalmente el Astrophysical Virtual Observatory (Europa) y el National Virtual Observatory (Estados Unidos). Éstos últimos están asociados en el seno del International Virtual Observatory Alliance con el fin de coordinar sus esfuerzos.

El observatorio virtual podrá automatizar el proceso, actualmente duro, de búsqueda y agrupación de datos astronómicos y recuperar la información para crear un todo superior a la suma de sus partes. Esto será posible por una parte, por un inmenso esfuerzo de estandarización tanto de los datos cómo de los métodos y herramientas utilizadas por los astrónomos, y de otra parte utilizando la tecnología GRID para acceder de manera transparente a la gran capacidad de cálculo repartida entre los distintos centros de datos del mundo, desde cualquier localidad.

Se trata de construir unos estándares de intercambio, de herramientas de demanda, de sistemas de extracción de la información, de manera que se globalicen los datos y más generalmente la información astronómica a nivel internacional. En este contexto, los principales centros de datos astronómicos internacionales se esfuerzan por dar más visibilidad a sus bases de datos y desarrollan descripciones detalladas de sus contenidos, los principales observatorios virtuales son:

I Astrophysical Virtual Observatory `http://www.euro-vo.org/`.

II National Virtual Observatory `http://www.us-vo.org/`

III International Virtual Observatory Alliance `http://www.ivoa.net/`

IV Australian Virtual Observatory `http://www.aus-vo.org/`

V Virtual Observatory of China http://www.china-vo.org/

VI Virtual Observatory-India http://vo.iucaa.ernet.in/

VII Canadian Virtual Observatory http://services.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/cvo/

VIII Spanish Virtual Observatory http://laeff.esa.es/svo/

IX DRACO http://wwwas.oat.ts.astro.it/draco/

X France Virtual Observatory http://www.france-ov.org/

XI German Astrophysical Virtual Observatory http://www.g-vo.org/gavo/index.html

XII Hungarian Virtual Observatory http://hvo.elte.hu/en/

XIII Japanese Virtual Observatory http://jvo.nao.ac.jp/index-e.html

XIV National Virtual Observatory (USA) http://www.us-vo.org/index.cfm

XV Korean Virtual Observatory http://kvo.kao.re.kr/

XVI Russian Virtual Observatory http://www.inasan.rssi.ru/eng/rvo/

Capítulo 3

Propuesta

3.1. Introducción

Como vimos en los capítulos anteriores, en la actualidad existen muchos trabajos en los campos de astronomía y astrofísica en los que es necesario el uso de herramientas que permitan a los astrofísicos un mejor manejo de los datos para optimizar tiempo y recursos, en esta tesis se presenta el uso de un algoritmo del área de minería de datos espacial *DBSCAN++*, el cual tiene su origen en el algoritmo DBSCAN [ ], sobre el cual se han realizado cambios para que pueda ser utilizado sobre datos de cúmulos de galaxias, los cuales se explicarán en detalle más adelante en la sección 3.3.

Uno de los principales inconveniente con el DBSCAN es el tiempo que tarda en realizar las operaciones de consultas de vecindad. En este capítulo tenemos la presentación del trabajo realizado a través de las mejoras que se le han hecho al algoritmo de DBSCAN y en resumen son las siguientes:

I Uso de Árboles R*[ ] los cuales son una variante de los árboles R[ ], usados para la indexación de datos espaciales.

II El uso de puntos de expansión: Al momento de realizar la expansión del cluster se utilizan puntos que se encuentran en los limites de la vecindad de el punto inicial, en vez de seguir

con el punto *i+1* de los datos, es decir el uso de los puntos borde.

3.2. Árboles R*

Los árboles R* [ ] son un método de acceso multidimensional, que se ha adoptado como uno de los métodos de acceso estándar para las bases de datos espaciales y es el elegido por la mayoría de los Sistemas de Administración de Bases de Datos. También es el más estudiado con respecto a tópicos tales como procesamiento y optimización de consultas, modelos de costo, paralelismo, control de concurrencia y recuperación, entre otros.

En un árbol R* no se almacenan los objetos espaciales en forma directa sino que se almacena su MBR (Minimum Bounding Rectangle), es decir el menor rectángulo que contiene al objeto en cuestión. El Rectángulo de Mínimo Acotamiento (MBR) se define como el rectángulo mínimo (p_1, p_2) que cubre completamente al objeto donde p_1 es la esquina superior derecha y p_2 es la esquina inferior izquierda. El MBR es una estrategia común de los métodos de acceso espaciales para almacenar aproximaciones del objeto así como utilizarlos para indexar los datos con el fin de tener una recuperación eficiente de ellos.

En la figura 3.1, podemos ver la estructura de datos espaciales que ocupan los árboles R^* Esta estructura fue introducida por [ ], la cual surge como respuesta al problema de búsquedas ineficientes cuando nodos del mismo nivel se solapan. Cuando los objetos se intersectan en más de un MBR en un nivel específico se recortan y se guardan en distintas páginas, en la figura 3.1 el objeto G intersecta dos paquetes. Los paquetes se delimitan por líneas discontinuas y están etiquetadas por las letras MBR_A, MBR_P, MBR_B y MBR_C. Entonces el objeto G quedará almacenado en MBR_A y en MBR_P.

La desventaja de este método es la cantidad de almacenamiento ya que al permitir almacenar paquetes en más de una hoja, el árbol ve incrementado su tamaño de forma mínima, pero con la ganancia de realizar consultas y encontrar información en un solo recorrido por el mismo. La representación de árbol se puede observar en la Figura 3.2.

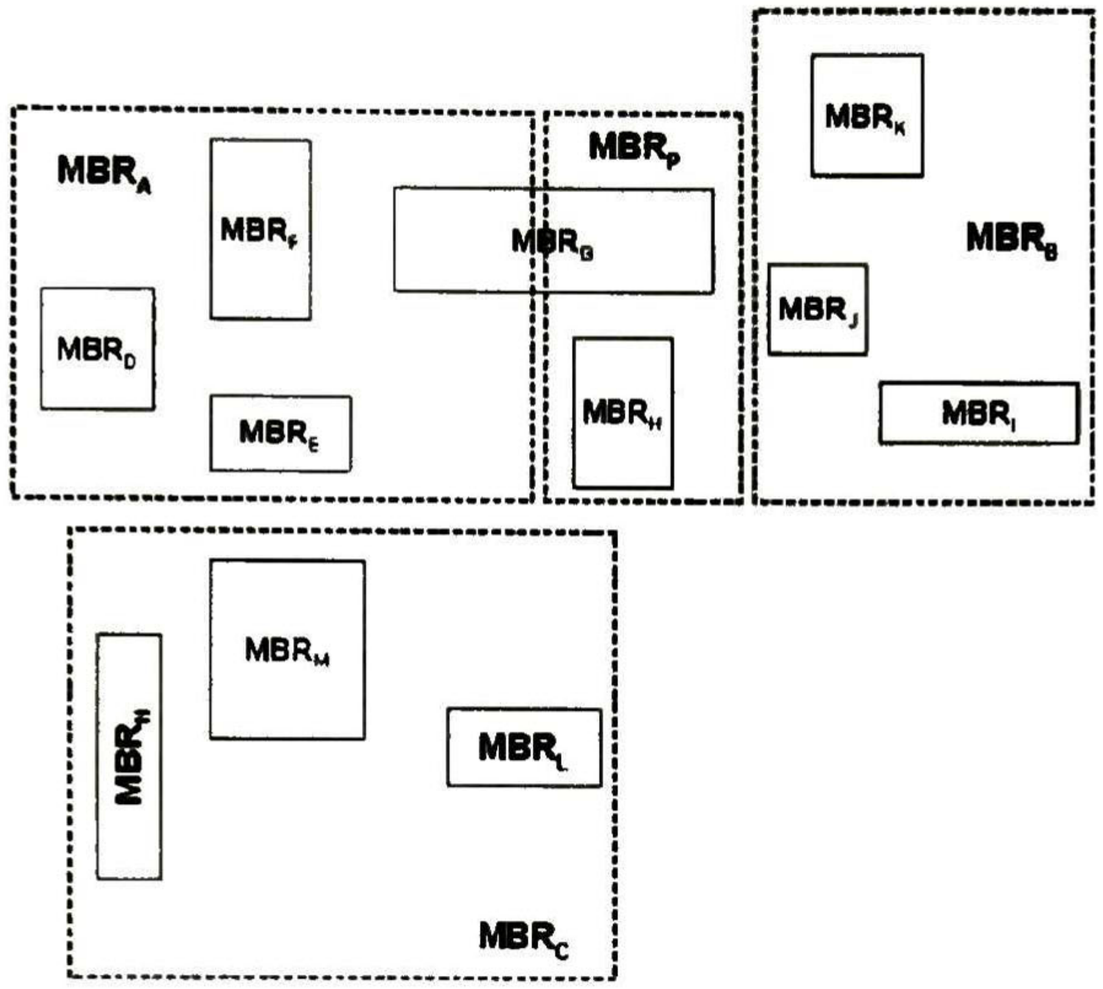

Figura 3.1: Ubicación de ciertos departamentos organizados en un árbol R^*

La generación del árbol mostrado en la figura 3.1 se realiza de la siguiente forma: Realizado el proceso de empaquetamiento y teniendo los MBR's la inserción de los objetos se produce de forma secuencial, en la cual el objeto que representa un departamento se compara con los MBR's para verificar si está contenido en él, en este caso la inserción se realiza en el nodo hijo (hoja). En este ejemplo podemos apreciar que el objeto que representa el departamento G está contenido tanto en el paquete MBR_A como MBR_P.

Para facilidad en la Figura 3.2 se omitió la palabra MBR, es decir por MBR_A solo escribimos A y así sucesivamente para todos los casos.

La forma de insertar y eliminar elementos de un árbol R* son uno de las principales ventajas de esta estructura.

La idea principal de la inserción es que dado un MBR, hallamos el nodo en dónde el MBR del objeto de la consulta pueda estar contenido en el MBR de un nodo. Puede suceder que este nodo sea una hoja, en tal caso añadiremos el MBR a la hoja, en caso contrario se añade un nodo el cual tendrá la función de servir como nodo interno que contendrá a las nuevas entradas que intersecten con ella.

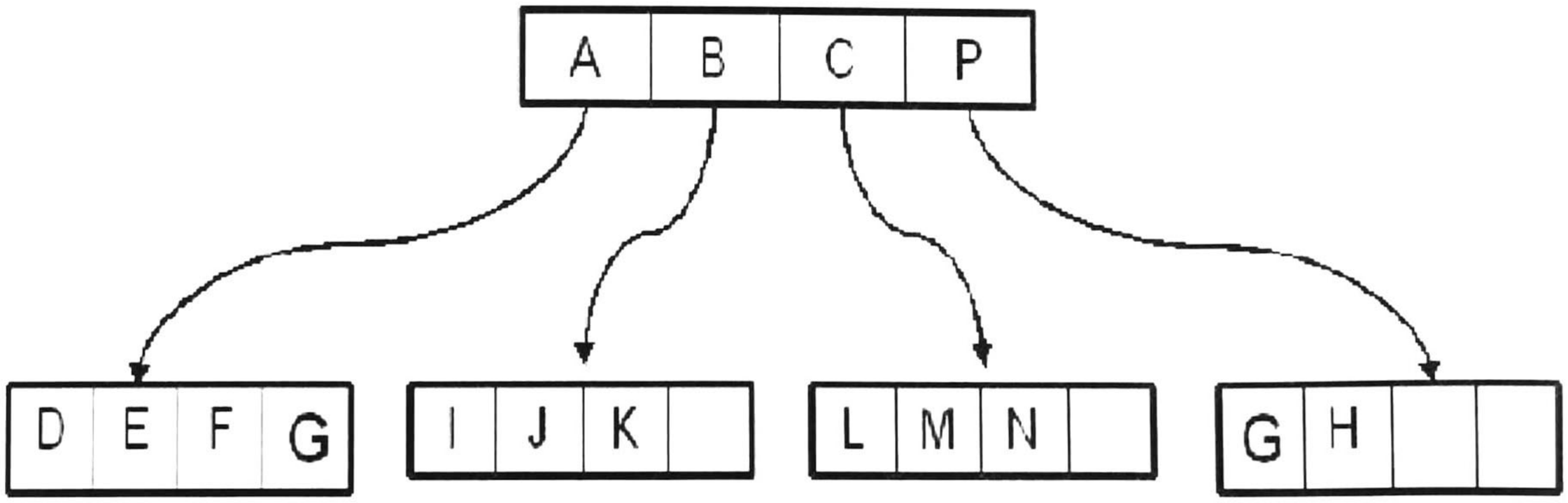

Figura 3.2: árbol R^* para la figura 3.1

Para el caso de la eliminación, el proceso consiste de dos etapas, primero realizar la búsqueda del elemento y eliminarlo, después realizar un reajuste de los nodos hojas si es el caso de incurrir en una cuota mínima para el nodo hoja.

Cada nodo en el árbol R* corresponde al MBR que contiene a sus hijos. Los nodos hoja del árbol R* contienen punteros a los objetos en la base de datos en vez de punteros a otros nodos. Cada nodo se almacena en una página de disco. Los nodos hoja del árbol R* contienen entradas de la forma $< I; oid >$ donde I es el menor rectángulo n-dimensional que contiene al objeto apuntado por oid; es decir, es de la forma $I = (I_0, I_1, ..., I_n)$. Aquí n es el número de dimensiones e I es un intervalo cerrado [a,b] que describe los límites del objeto en la dimensión i. En caso de que un objeto espacial se extienda más allá de los límites del espacio definido, entonces I_i puede tener uno o ambos puntos extremos igual a infinito.

Los nodos que no son de tipo hoja, contienen entradas de la forma $< I; pchild >$ donde *pchild* es un puntero a un hijo del nodo e I contiene a todos los MBR's del nodo apuntado por *p*child. En un árbol R*, cada nodo, con la posible excepción del nodo raíz, contiene entre m y M entradas donde $m \leq M/2$ y M es el número máximo de entradas por nodo; el nodo raíz tiene al menos dos hijos a menos que sea una hoja; y todas las hojas están al mismo nivel.

Debido a estas características es que se escogió como medio de indexación de los datos a

Figura 3.3: El cúmulo de galaxias HCG 87, a unos cuatrocientos millones de años luz de distancia, Copyright NASA.

los árboles R*

3.3. Cúmulos de galaxias

El tipo de datos con los cuales se trabajo en esta tesis son los cúmulos de galaxias, que fueron provistos por el Dr. Heinz Andernach y Dr. Cesar Careta del departamento de Astronomía de la Universidad de Guanajuato.

Los cúmulos de galaxias son entornos de alta densidad donde las galaxias interaccionan unas con otras y con el potencial gravitatorio del cúmulo según [ ] Estas interacciones hacen que la evolución de dichas galaxias sea muy diferente que la de las galaxias de campo, es decir las galaxias que se encuentran aisladas. Es conocido desde las primeras observaciones de cúmulos, que las propiedades de las galaxias que se encuentran en entornos de alta densidad son diferentes de las que se encuentran aisladas.

Así, la población de galaxias presentes en cúmulos está dominada por galaxias de tipo temprano [ ] , principalmente elípticas y S0.

Estas galaxias residen sobre todo en las partes centrales de los cúmulos donde la densidad de galaxias es mayor, formando una familia homogénea de objetos que siguen fuertes relaciones observacionales, tales como: plano fundamental o relación color-magnitud.

Por el contrario, las galaxias de tipo tardío son menos abundantes en los cúmulos, siendo dominantes en la población de campo. Aún así, no están ausentes en los cúmulos, encontrándose en las regiones más externas de los mismos, donde la densidad de galaxias no es muy alta.

Estas diferencias observacionales entre las galaxias de campo y en cúmulos se conocen desde las primeras observaciones que se hicieron en cúmulos hacia la mitad del siglo XX según [ ]. Esto sugirió desde un principio que dichas diferencias entre galaxias de campo y cúmulos eran debidas a diferentes procesos de formación. Sin embargo, desde la aceptación general de las teorías jerárquicas como los modelos preferidos que explican la formación de estructuras, en los cuales las galaxias brillantes se forman por fusiones e interacciones, se ha puesto toda la atención en los mecanismos que pueden transformar galaxias de tipo tardío (dominantes en campo) en tipo temprano (dominantes en cúmulos).

Durante los últimos años, el desarrollo tanto de equipo de computo, como de herramientas de computación ha permitido simular la evolución de galaxias en cúmulos. Estas simulaciones obtienen que las galaxias evolucionan rápidamente debido a las interacciones que sufren unas con otras así como con el potencial global del cúmulo, produciendo drásticas transformaciones morfológicas en las galaxias presentes en cúmulos. Este escenario es el que se conoce como teoría de acoso galáctico de su nombre inglés "harassment"

Estas interacciones hacen que parte del material estelar de las galaxias sea arrancado de las mismas y quede ligado al potencial cumular formando la llamada luz difusa o luz intracumular. El estudio de la distribución de esta componente y su cinemática nos puede dar información directa sobre como se ha ensamblado la masa en los cúmulos de galaxias.

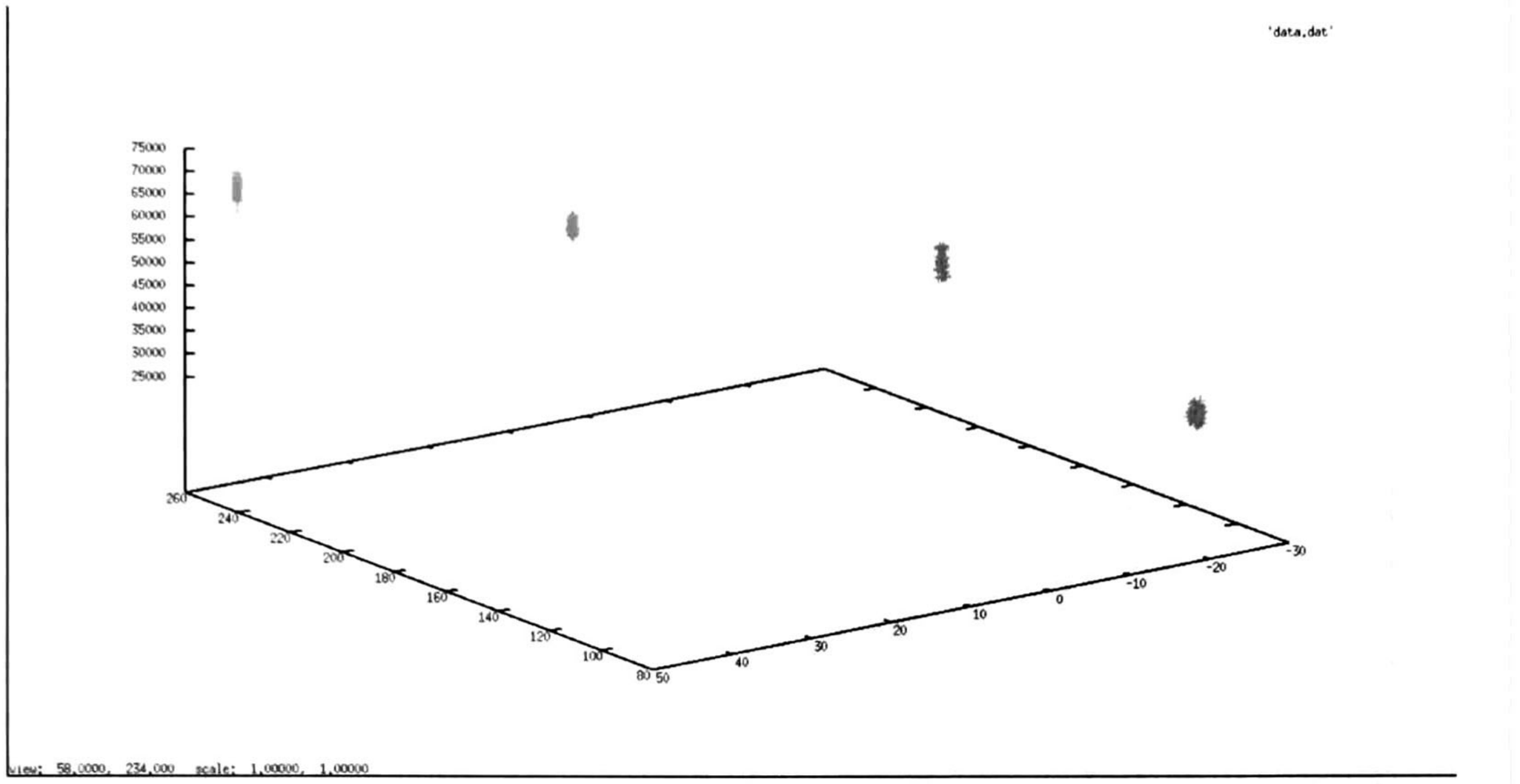

Figura 3.4: Vista de 4 cúmulos

Además, las interacciones de las galaxias con el medio intracumular caliente presente en los cúmulos producen que gran cantidad de gas galáctico sea arrancado de los discos de las galaxias espirales, produciendo una disminución de su formación estelar. Una evidencia observacional directa de este mecanismo es la deficiencia de HI que presentan los discos de las galaxias localizadas en cúmulos frente a las de campo.

Todos estos mecanismos transforman galaxias tardías en tempranas. Pero hay una serie de evidencias observacionales que pueden ser directamente contrastadas:

① Distribución morfológica de las galaxias de los cúmulos;

② Función de luminosidad;

③ Luz difusa (cantidad y distribución);

④ Presencia de subestructura.

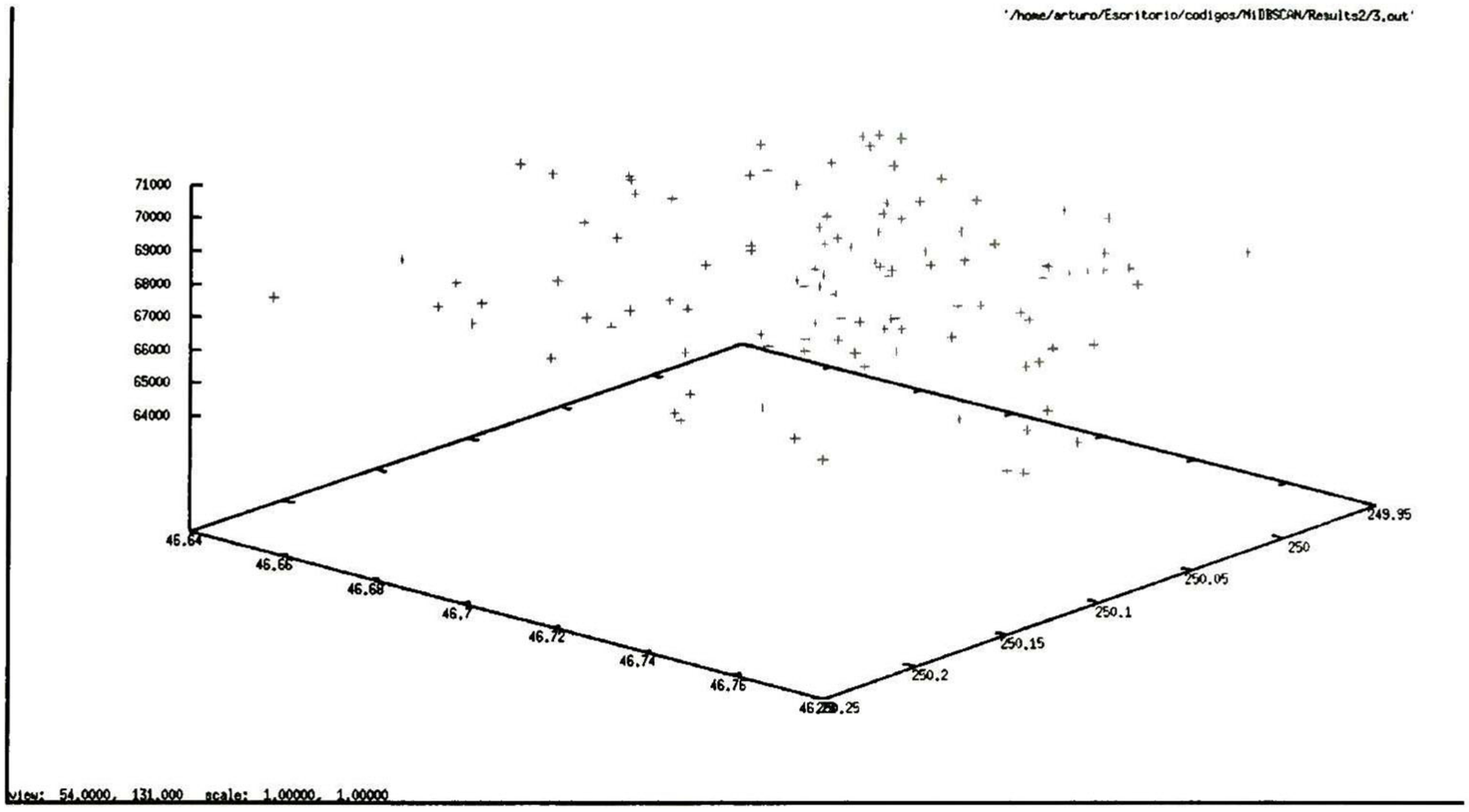

Figura 3.5: Gráfico del cúmulo A2219 con 117 miembros

3.4. DBSCAN

DBSCAN maneja 2 parámetros de entrada los cuales son *minpuntos* y *épsilon*, los cuales indican el número de puntos requeridos para que la agrupación sea considerada como un clúster y el radio épsilon, el cual es la distancia mínima cuando un punto se considera que esta dentro de la vecindad de otro.

Una de las principales razones por la cual DBSCAN[ ] puede reconocer los clúster es que dentro de cada clúster hay un típica densidad de puntos la cual es considerablemente mayor que los puntos afuera de los clusters, además la densidad dentro de las zonas de ruido es inferior a la densidad en cualquiera de los clusters. Ahora vamos a definir como es que el algoritmo DBSCAN trabaja:

Definición 1 (Epsilon-Vecindad de un punto). La Epsilon-Vecindad de un punto p, denotada por $N_{Eps}(p)$ es definida por:

$$N_Eps(p) = \{q \in D \mid dist(p,q) \leqslant Eps\}$$

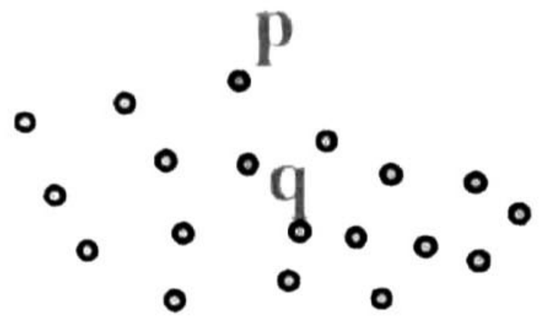

Figura 3.6: p es un punto borde y q es un punto central

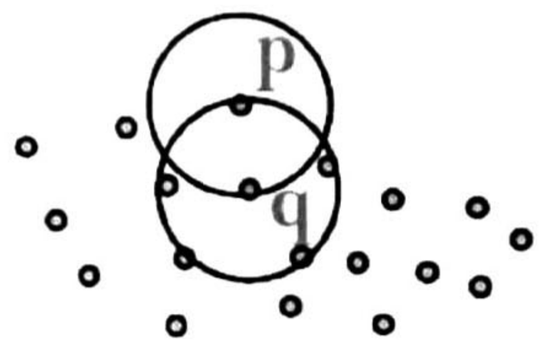

Figura 3.7: p es Directamente Densamente-Alcanzable desde q. q No es Directamente Densamente-Alcanzable desde p

Esta definición nos dice que un punto es *vecino* de otro sí, la distancia entre ellos es menor que un valor umbral *épsilon*.

Definición 2 (Directamente Densamente-Alcanzable). Un punto P es directamente densamente-alcanzable desde un punto q con respecto a *épsilon minpuntos* sí cumple las siguientes dos condiciones:

$$p \in N_{Eps}(q)$$

$$\mid N_{Eps}(q) \mid \geq minpuntos.$$

Este último punto es el que se conoce como *Condición de Punto Central*, obviamente la relación de *Directamente Densamente-Alcanzable* es simétrica para pares de puntos centrales. Pero en la figura 3.7 se muestra que cuando ocurre para un punto borde y otro punto central, esto no ocurre.

Definición 3 (Densamente Alcanzable). Un punto p es Densamente Alcanzable desde un punto q con respecto a épsilon y un Mínimo de Puntos si existe una cadena de puntos $p_1 \cdots p_n, p_1 = q, p_n$ tal que $p_{1+1} es$ Directamente Densamente-Alcanzable desde p_i.

En la definición anterior se define cuando un punto es densamente alcanzable desde otro

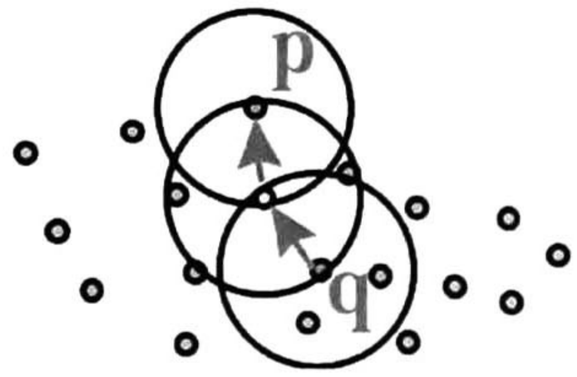

Figura 3.8: p es directamente densamente alcanzable desde q

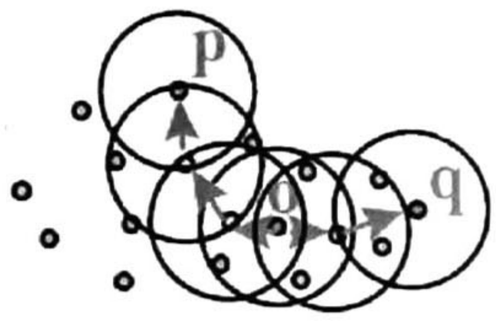

Figura 3.9: p y q estan densamente conectados entre sí por medio de o

punto por medio de medio de puntos intermedios, en la figura 3.8

Definición 4 (Densamente Conectado). Un punto p es Densamente conectado a un punto q con respecto a épsilon y un Mínimo de Puntos sí existe un punto o de tal manera que ambos, p y q son densamente alcanzables con respecto a épsilon y un Mínimo de Puntos.

Densamente Conectado es una relación simétrica, para puntos densamente alcanzables la relación es tambien reflexiva, como podemos ver en la figura 3.9.

Definición 5 (Cluster). Sea D una base de datos de puntos, un clúster C con respecto a épsilon y un Mínimo de Puntos es una subserie de D no vacía que satisface las siguientes condiciones:

1- $\forall p, q$: si $p \in C$ y q es densamente alcanzable con respecto a épsilon y un mínimo de Puntos $\Rightarrow q \in C$ y

2- $\forall p, q \in C : p$ es densamente conectado a q con respecto a *épsilon* y un *mínimo de Puntos*.

Esta es la definición que usamos en esta tesis para comprobar que un grupo de datos son o pertenecen a un cluster. Ahora es importante también definir otro concepto importante, recordemos que el algoritmo es sensible al ruido, es decir puede detectar que puntos no pertenecen a ningún clúster y esto lo definimos de la siguiente manera:

Definición 6 (Ruido). Sea $C_1 \cdots C_k$ Una serie de clusters de la base de datos D con respecto a épsilon y un Mínimo de Puntos Entonces definimos al ruido como la serie de puntos en la base de datos D que no pertenecen a ningún clúster C_i es decir:

$$ruido = p \in D \mid \forall i : p \notin C_i.$$

Lema 1. Sea p un punto en D y $\mid N_{Eps}(P) \mid \geq minpuntos$. Entonces $O = \{o \mid o \in D$ y o es densamente alcanzable desde p con respecto a *épsilon* y $MinPuntos\}$ es un clúster con respecto a *épsilon* y $MinPuntos$

Lema 2. Sea C un clúster con respecto *épsilon* y $MinPuntos$ y sea p un punto cualquiera dentro de C con $\mid N_{Eps}(P) \mid \geq minpuntos$ Entonces C que es igual a la serie $O = \{o \mid o$ es densamente alcanzable desde p con respecto a *épsilon* y $MinPuntos\}$.

3.4.1. Mejoras

DBSCAN tiene una complejidad promedio de $O(n\ log\ n)$ pero una buena parte de este tiempo es consumido por las operaciones de consulta, de hecho podemos decir que el proceso de clustering es un procedimiento iterativo de ejecutar operaciones de consultas. Y es en este punto de donde partimos para realizar una de las mejoras que se realizaron en el desarrollo de esta tesis, la cual fue reducir el numero de consultas de vecindad. De acuerdo con [ ] el valor default de requerimiento mínimo de puntos es igual a 4 ($K = 4$), esto para clusters reducidos es un buen requerimiento pero para clusters densamente poblados, el numero de vecinos es mayor a K. Ahora el procedimiento normal del DBSCAN es llevar a cabo operaciones de consulta para cada objeto contenido en la vecindad del objeto central.

Es normal que si existe cierta cantidad de vecinos de este objeto, entonces al momento de realizar las consultas de vecindad de los demás objetos contenidos, halla alguna intersección entre estos por lo que si se omite alguna de estas consultas el resultado es el mismo.

Por lo tanto el tiempo consumido por la operación de expansión para pj puede ser eliminado, y esto aplica aun más en clusters densamente poblados, eliminar ciertas consultas de vecindad dentro de la vecindad de los objetos centrales, puede ser ignorado.

Algorithm 3.1 Algoritmo de Clustering

Require: SetOfObjects, Epsilon, MinObjects
Ensure: Clusters

```
1: {DBSCAN (SetOfObjects, Eps, MinObjects)}
2: ClusterId=NextId(NOISE);
3: for i = 0 hasta SetOfObjects.size do
4:    Object= SetOfObjects.get(i);
5:    if Object.ClId = UNCLASSIFIED then
6:       if ExpandCluster(SetOfObjects, Object,ClusterId, Eps, MinObjects) then
7:          ClusterId := nextId(ClusterId)
8:       end if
9:    end if
10: end for
```

Entonces una manera de acelerar el algoritmo DBSCAN es tomar en cuenta puntos representativos dentro de la vecindad del objeto. Para esto se propuso que los puntos que están mas lejanos al punto central sean escogidos.

3.4.2. Pruebas

Durante las pruebas realizadas con el algoritmo propuesto se encontró que en algunas ocasiones los resultados varían un poco en cuestión con el clustering obtenido, es decir en algunas ocasiones se obtiene que datos que deberían formar parte de algún cluster son considerados como ruido, para eso fue necesario el uso de una función que verificara que los puntos que eran considerados como ruido.

Algorithm 3.2 Algoritmo de Selección de Puntos Borde

Require: SetOfObjects, Epsilon, MinObjects
Ensure: Clusters

```
1: {ObjetosExpansion(Vecindad, ObjetosBorde,Objeto)}
2: ObjetosBorde=0;
3: for i = 0 hasta Vecindad.size do
4:    ObjetoActual=Vecindad.get(i);
5:    if i=0 then
6:       BordeXpos=ObjetoActual;
7:       BordeXneg=ObjetoActual;
8:       BordeYpos=ObjetoActual;
9:       BordeYneg=ObjetoActual;
10:      BordeZpos=ObjetoActual;
11:      BordeZneg=ObjetoActual;
12:   end if
13:   if ObjetoActual.x¿BordeXpos.x then
14:      BordeXpos=ObjetoActual;
15:   end if
16:   if ObjetoActual.¡xBordeXneg.x then
17:      BordeXneg=ObjetoActual;
18:   end if
19:   if ObjetoActual.y¿BordeYpos.y then
20:      BordeYpos=ObjetoActual;
21:   end if
22:   if ObjetoActual.y¡BordeYneg.y then
23:      BordeYneg=ObjetoActual;
24:   end if
25:   if ObjetoActual.z¿BordeZpos.z then
26:      BordeZpos=ObjetoActual;
27:   end if
28:   if ObjetoActual.z¡BordeZneg.z then
29:      BordeZneg=ObjetoActual;
30:   end if
31: end for
```

Capítulo 4

Resultados

4.1. Introducción

En este capítulo se describe y se profundiza en la propuesta general del capítulo anterior, también se describen los experimentos realizados, Se realiza un análisis de los datos obtenidos y de esta manera se comprueba la eficiencia del algoritmo propuesto.

Los datos con los que trabaja el algoritmo son cúmulos de galaxias, donde cada cúmulo contiene entre 200 y 2000 galaxias, con el fin de encontrar subestructura como grupos de planetas, sistemas planetarios, por citar algunos ejemplos.

Example of Data

```
El formato de la tabla de datos es la siguiente:
Abell hhmmss.ss+dd''"" " vhelio err ref
A2219 163954.53+464016.4 67614 223 661
byte    fmt     description
-------------------------------------------------------------------
1       A1      A or S for A-or S-cluster from ACO 1989;
                sorted by number, first all A- then all S-clusters
```

```
2-5     I4      Abell number
6       A1      cluster component: E,W,N,S if displaced on the sky;
or A,B,C.
in order of distance, if displacement is not obvious;
a minus sign in this column indicates that the cluster
is a duplication of another entry in the ACO catalogue
and should be deleted for statistical studies.
7-15    A9      RA_J2000 of galaxy, hhmmss.ss (2I2,F5.2).
16-24   A9      DE_J2000 of galaxy, +dd''"" " (A1,2I2,F4.1)
25-30   I6      Radial velocity in km/s.
32-34   I3      Velocity error in km/s.
36-38   I3      Reference code.
```

Cada dato contiene:

① Un identificador del cúmulo al que pertenece,

② La orientación o la distancia aproximada,

③ La Ascensión Recta al año 2000,

④ La DEclinación al año 2000,

⑤ La Velocidad Radial,

⑥ Error de Velocidad Radial,

⑦ Código de referencia.

La ascensión recta (AR) y la declinación (DE) son las medidas utilizadas por los astrónomos para especificar lugares en el cielo. Son muy similares a la latitud y la longitud en la Tierra. La declinación de un punto en el cielo, como la latitud en la Tierra, es un número entre -90 y +90 grados. La ascensión recta de un punto del cielo es muy similar a la longitud,

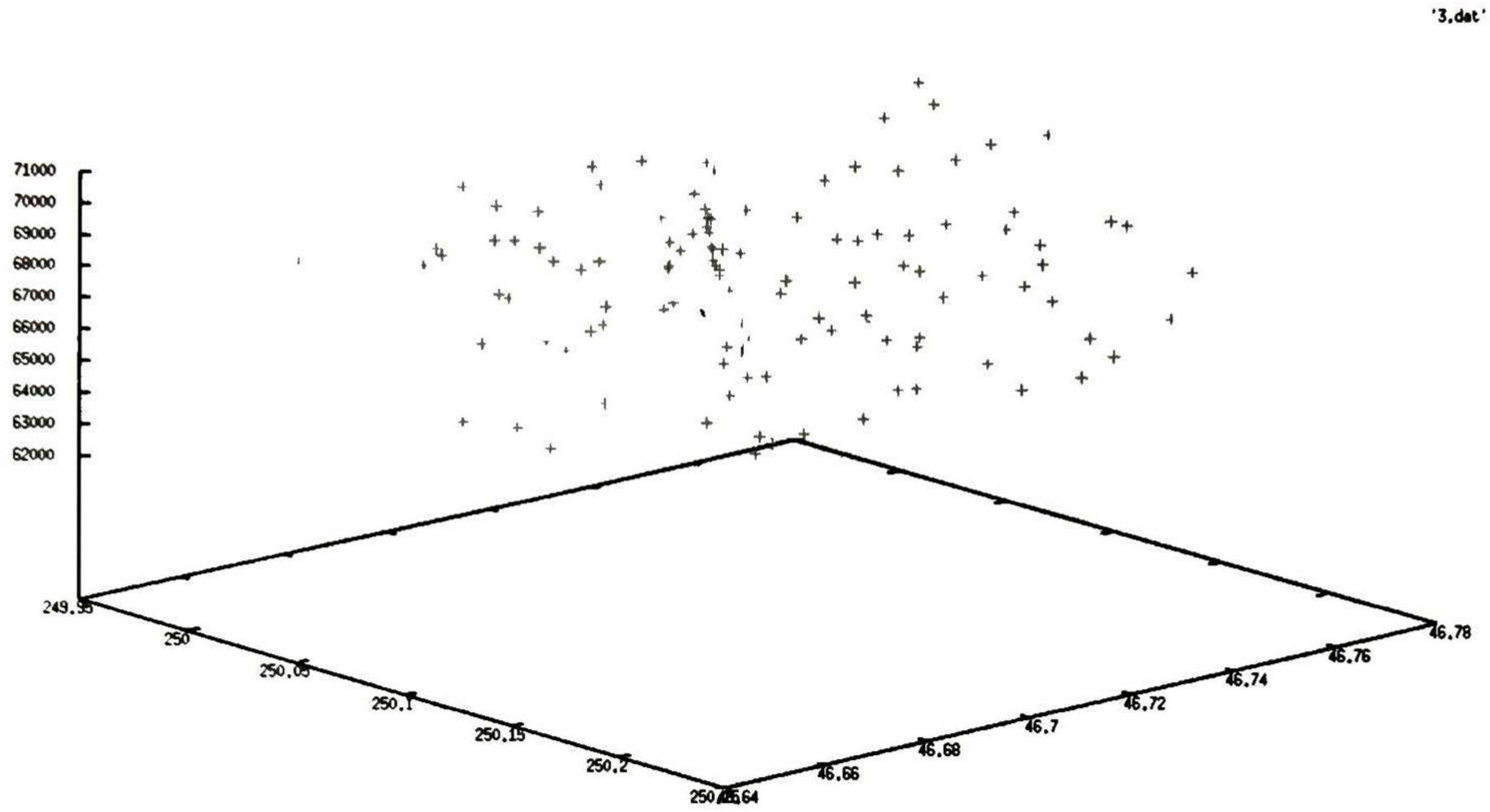

Figura 4.1: En este gráfico se muestra la distribución de los datos de las tablas 4.1, 4.2, 4.3, 4.4.

salvo que en vez de estar expresada en grados, se expresa en horas, minutos y segundos, y puede ir desde 0h 0m 0s hasta 24h 0m 0s. Dando una AR y una declinación, cualquier punto del cielo puede ser especificado con precisión.

En las tablas 4.1, 4.2, 4.3, 4.4 se puede ver un ejemplo del cúmulo A550A en el cual se ve detallado los 128 datos de entrada cada uno con su Ascensión Recta, Declinación, Velocidad Radial y la ultima columna es el error de la velocidad radial.

4.2. Resultados Obtenidos

Los resultados presentados son obtenidos del promedio de tres ejecuciones de ambos algoritmos, cada una con un valor de minpuntos=40 y un valor de épsilon=800. En la figura 4.2, vemos que cuando crece el tamaño de los datos a considerar por los algoritmos de clustering, el tiempo del DBSCAN crece de manera considerable, en cambio con el DBSCAN++ el tiempo se mantiene mas constante.

Tabla 4.1: Datos Cúmulo A550A

ID	RA_J2000	DE_J2000	Velocidad Radial	Error
550	87.194583	-21.106389	27911.00	64.00
550	87.241250	-21.409472	29230.00	64.00
550	87.246667	-21.107806	27701.00	64.00
550	87.265000	-21.421361	29979.00	64.00
550	87.277083	-20.920000	27881.00	64.00
550	87.327083	-20.910833	26921.00	64.00
550	87.335833	-21.379167	29859.00	64.00
550	87.336667	-21.449528	32228.00	64.00
550	87.337083	-21.399500	29320.00	64.00
550	87.441667	-21.550417	27161.00	64.00
550	87.447917	-21.190278	27491.00	64.00
550	87.458333	-20.726417	27431.00	64.00
550	87.470417	-21.380639	27161.00	64.00
550	87.472500	-21.174111	27611.00	90.00
550	87.522083	-21.625972	28540.00	64.00
550	87.557500	-21.430167	27521.00	64.00
550	87.608333	-21.582222	29769.00	64.00
550	87.621667	-20.963139	30729.00	90.00
550	87.650417	-21.089306	27311.00	64.00
550	87.710417	-21.563806	28870.00	64.00
550	87.717500	-21.220944	30789.00	64.00
550	87.781250	-21.919028	29560.00	90.00
550	87.807917	-21.350694	27581.00	64.00
550	87.809583	-21.451667	29769.00	64.00
550	87.810417	-21.506028	29470.00	64.00
550	87.814167	-21.626583	29680.00	64.00
550	87.825833	-20.605278	30249.00	64.00
550	87.857083	-21.512556	29739.00	64.00
550	87.867917	-20.248417	29949.00	64.00
550	87.870000	-20.942361	27401.00	64.00
550	87.884167	-21.121472	29859.00	64.00
550	87.898333	-21.145778	29020.00	64.00
550	87.902917	-21.419528	29859.00	64.00
550	87.905000	-20.672167	30249.00	64.00
550	87.919167	-20.491028	30099.00	64.00
550	87.939583	-21.119333	29979.00	64.00
550	87.944167	-21.043056	27431.00	64.00
550	87.953750	-21.573722	29590.00	64.00
550	87.961250	-20.448306	30009.00	64.00

Tabla 4.2: Datos Cúmulo A550A

ID	RA_J2000	DE_J2000	Velocidad Radial	Error
550	87.974167	-21.032639	30669.00	64.00
550	87.991250	-21.920056	30369.00	90.00
550	87.995833	-21.289722	28151.00	64.00
550	88.006667	-21.391917	28750.00	64.00
550	88.012500	-20.864722	30369.00	64.00
550	88.044583	-21.124833	30159.00	64.00
550	88.048333	-21.643194	30849.00	64.00
550	88.053333	-21.493583	28540.00	64.00
550	88.062083	-21.632556	28270.00	64.00
550	88.062500	-20.847472	30489.00	64.00
550	88.072500	-21.456389	27791.00	64.00
550	88.093750	-21.210444	30009.00	64.00
550	88.094583	-21.360389	30699.00	64.00
550	88.100417	-20.859250	29709.00	64.00
550	88.103750	-20.952389	27761.00	90.00
550	88.117500	-21.171306	28540.00	64.00
550	88.130417	-21.227500	30639.00	64.00
550	88.136667	-20.957361	27731.00	64.00
550	88.144583	-21.180139	30309.00	64.00
550	88.154583	-20.458583	27911.00	64.00
550	88.163750	-21.076361	30639.00	64.00
550	88.163750	-21.155333	30189.00	64.00
550	88.168750	-21.017194	29650.00	64.00
550	88.180000	-21.044861	28870.00	64.00
550	88.185000	-20.976417	29709.00	64.00
550	88.191667	-21.085194	28990.00	64.00
550	88.198333	-20.764083	29320.00	64.00
550	88.206667	-21.026194	30909.00	64.00
550	88.210417	-21.397667	30579.00	64.00
550	88.211250	-21.084194	29829.00	64.00
550	88.211667	-20.955778	28750.00	64.00
550	88.212917	-21.051417	29913.00	80.00
550	88.223750	-21.092194	28300.00	64.00
550	88.237083	-21.462194	30999.00	64.00
550	88.238333	-21.332472	29560.00	64.00
550	88.239583	-20.960139	31898.00	64.00
550	88.241250	-21.055056	31328.00	64.00
550	88.243750	-21.081000	29588.00	50.00
550	88.246250	-20.919972	31808.00	64.00

Tabla 4.3: Datos Cúmulo A550A

ID	RA_J2000	DE_J2000	Velocidad Radial	Error
550	88.251667	-21.086361	29680.00	64.00
550	88.263750	-21.099444	29829.00	64.00
550	88.265417	-21.654778	30369.00	64.00
550	88.274167	-21.180333	27491.00	64.00
550	88.275000	-21.123694	30999.00	90.00
550	88.279583	-21.030167	27731.00	64.00
550	88.285833	-21.689472	30999.00	64.00
550	88.287917	-21.243917	28480.00	64.00
550	88.299167	-21.110194	27671.00	64.00
550	88.329167	-20.965250	29440.00	64.00
550	88.335833	-20.300167	28031.00	64.00
550	88.336667	-20.984472	32198.00	64.00
550	88.342083	-21.075111	31628.00	90.00
550	88.382917	-21.123111	28720.00	64.00
550	88.391250	-21.194028	30909.00	64.00
550	88.395000	-20.985889	29769.00	64.00
550	88.405833	-21.017556	30489.00	64.00
550	88.425833	-21.554222	29769.00	64.00
550	88.431250	-21.179139	30789.00	64.00
550	88.437500	-21.557167	30069.00	64.00
550	88.440417	-21.074806	31029.00	64.00
550	88.442917	-21.188806	30759.00	64.00
550	88.452083	-21.002194	30789.00	64.00
550	88.474167	-21.469722	30549.00	64.00
550	88.481250	-21.312806	29919.00	64.00
550	88.483333	-21.347056	29110.00	64.00
550	88.491250	-21.219833	29290.00	64.00
550	88.494583	-20.916944	28990.00	64.00
550	88.502917	-20.145694	29080.00	64.00
550	88.516250	-20.730139	30759.00	64.00
550	88.533750	-21.417167	29290.00	90.00
550	88.537500	-20.803306	31358.00	90.00
550	88.553333	-21.710722	31298.00	64.00
550	88.575833	-21.940306	29709.00	64.00
550	88.606667	-21.240972	29530.00	64.00
550	88.627917	-21.126472	28540.00	64.00
550	88.645417	-21.189528	28420.00	64.00
550	88.680833	-21.920333	27761.00	64.00
550	88.691250	-21.575556	27461.00	90.00

Tabla 4.4: Datos Cúmulo A550A

ID	RA_J2000	DE_J2000	Velocidad Radial	Error
550	88.703750	-21.681917	30039.00	64.00
550	88.704583	-21.655944	27731.00	64.00
550	88.717083	-21.913444	28570.00	64.00
550	88.723333	-20.942111	28480.00	64.00
550	88.745000	-20.978972	28480.00	90.00
550	88.805833	-20.739278	29380.00	64.00
550	88.840417	-21.711167	27311.00	90.00
550	88.840833	-20.920417	29200.00	64.00
550	88.842083	-20.742583	30309.00	64.00
550	88.928750	-20.869389	28450.00	90.00
550	89.190417	-20.749694	27761.00	90.00

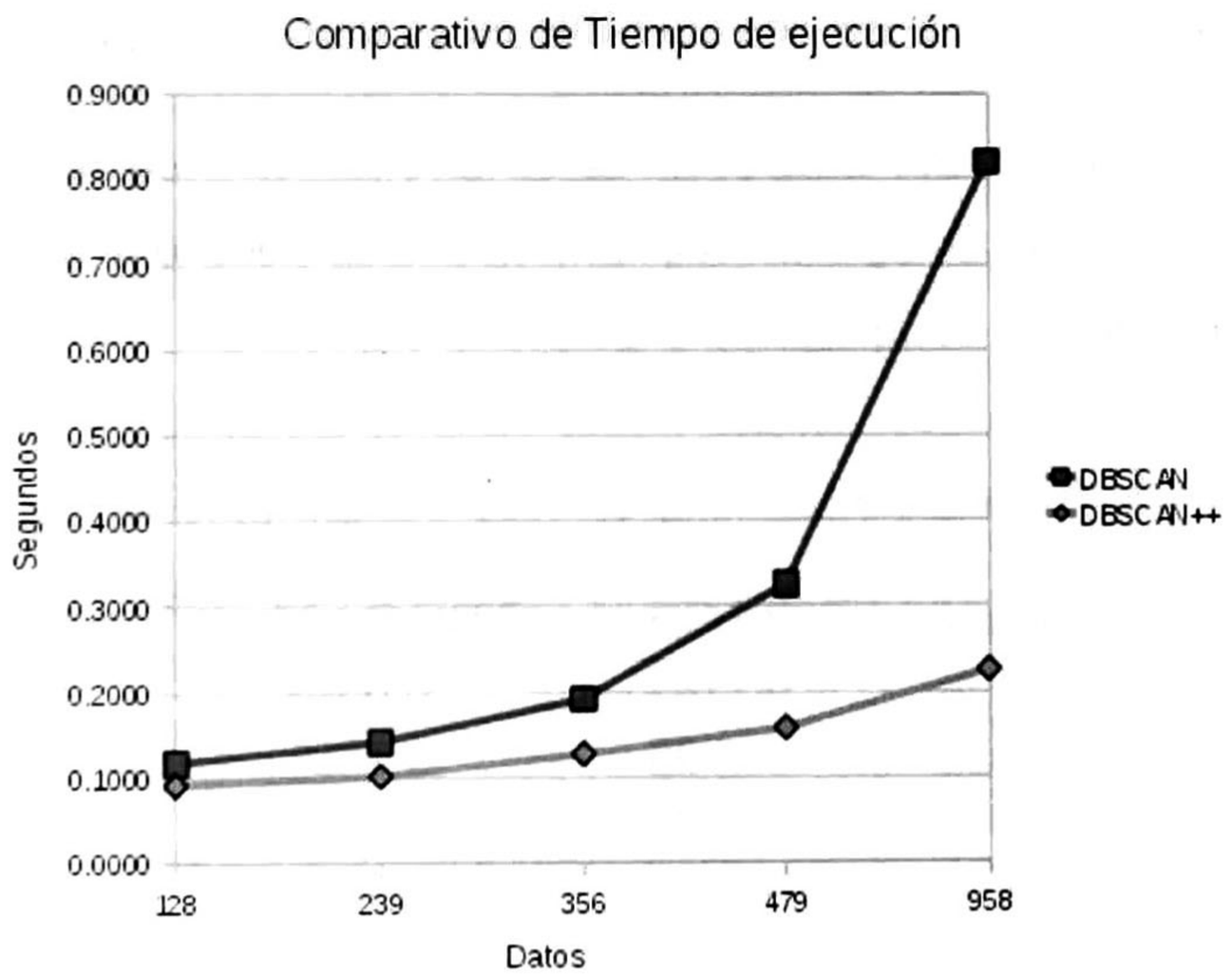

Figura 4.2: En este gráfico se muestran los resultados de la ejecusión de el DBSCAN contra el DBSCAN++

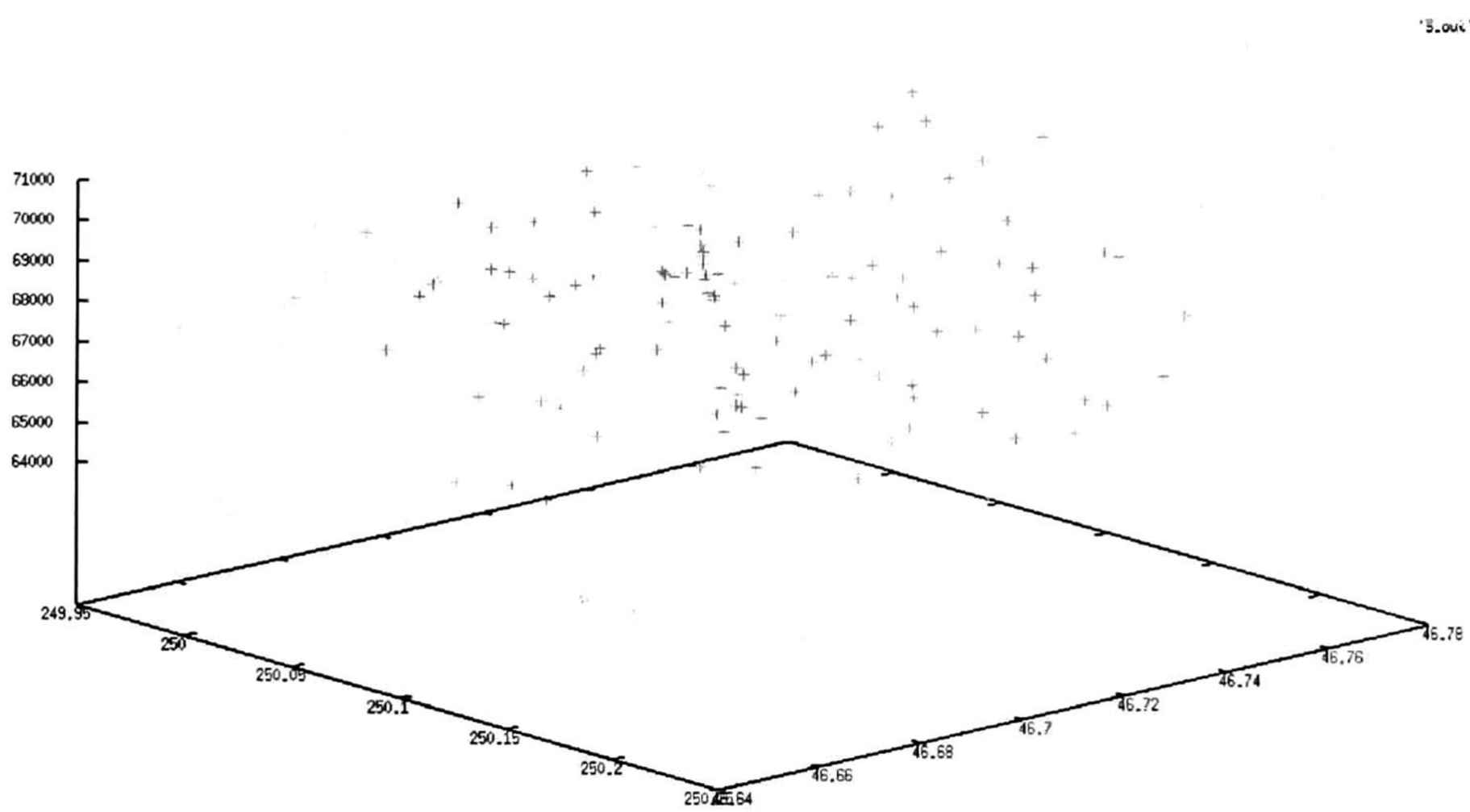

Figura 4.3: En este Gráfico se muestra la distribución de salida generados por el algoritmo DBSCAN++

Esto muestra que al momento de utilizar este algoritmo con grandes bases de datos se obtiene una reducción que puede llegar a ser considerable en el tiempo de procesamiento.

La Complejidad del algoritmo DBSCAN es de $O(C_1 * \ n * log\ n)$ y la complejidad del algoritmo DBSCAN++ es de $O(C_2 * \ n * log\ n)$ y podemos decir que siempre se cumple que: $C_2 \ < \ C_1$, lo cual nos da una complejidad menor.

Tabla 4.5: Resultados al comparar DBSCAN VS DBSCAN++

Datos	**DBSCAN**	**DBSCAN++**
128	0.1182	0.0918
239	0.1423	0.1021
356	0.1921	0.1276
479	0.3254	0.1575
958	0.8212	0.2253

En la figura 4.1, se muestra la gráfica generada por los datos de entrada de los 128 datos, y enseguida en la figura 4.3, se muestra los datos generados por el DBSCAN++, en el que

se puede observar que el rango de las coordenadas de la velocidad radial aumenta a 64,000 cuando en el cúmulo anterior el rango alcanzaba los 62,000 esto significa que los datos que estaban cercanos a este valor no fueron incluidos en el cluster.

Capítulo 5

Conclusión y trabajo futuro

5.1. Introducción

En este capítulo se presentan las conclusiones finales de la tesis y se expone un análisis de las limitantes de la tesis que pueden ser atacadas como trabajo a futuro.

5.2. Conclusión

Se realizó la presentación de un algoritmo de clutering de minería de datos espacial basado en densidad DBSCAN++. Este algoritmo representa un cambio con respecto al DBSCAN original en la forma de realizar las consultas de expansión. Este cambio implica que el impacto que el algoritmo tiene durante el tiempo de ejecución en dichas consultas sea reducido, lo que brinda mayores ventajas sin perjudicar los resultados obtenidos.

El uso de los puntos bordes no implica que aumente el número de puntos ruido que sean considerados como puntos pertenecientes a un cluster ya que de todas maneras este punto iba a ser considerado para las tareas de clustering. En vez de eso en algunas ocasiones lo que se obtuvo fue que datos que debían ser considerados como pertenecientes a un cluster no lo fueron, por lo que fue necesario la creación de una nueva función para verificar que cada

punto ruido realmente lo sea.

La motivación de esta tesis fue encontrar primeramente un método de minería de datos espacial útil en la manipulación de datos de cúmulos de galaxias.
El método elegido fue el DBSCAN, al que posteriormente se prosiguió a que pudiera ser utilizado con este tipo de datos y luego se le añadieron 2 mejoras a este algoritmo que fueron el uso de Árboles R* y el uso de puntos borde al momento de realizar la expansión propia del algoritmo. El resultado obtenido fue una reducción en el tiempo de procesamiento.

5.3. Trabajo futuro

Como trabajo futuro tenemos:

Realizar un análisis más detallado sobre el desempeño del algoritmo para poder aumentar su desempeño.

Ejecutar el algoritmo DBSCAN++ con una base de datos completa de cúmulos de galaxias.

Apéndice A

Codigos en C++

```
#include "dbscan.h"
#include "Point.h"
#include "rtree.h"
#include <math.h>
#include <iostream>
using namespace std;
float distance(Point *p1, Point *p2)
{
        float* point1coords = new float[4];
        point1coords = p1 -> GetPointCoords();
        float* point2coords = new float[4];
        point2coords = p2 -> GetPointCoords();
        float xd = point2coords[0] - point1coords[0];
        float yd = point2coords[1] - point1coords[1];
        float zd = point2coords[2] - point1coords[2];
        float ed = point2coords[3] - point1coords[3];
        return sqrt(xd * xd + yd * yd + zd * zd);
        //return sqrt(xd * xd + yd * yd + zd * zd + ed * ed);
}
/** vector<Point*> density_reachable
```

(Point point, vector<Point*> main_p, float epsilon, int min_points)*
Recibe un punto, el Vector completo de puntos, eps y min points
Regresa el vector de cluster de puntos vecinos

**/

```
vector<Point*> density_reachable
(Point *point, vector<Point*> main_p, float epsilon, int min_points)
{
        int tam=0;
        vector<Point*> cluster;
        vector<Point*>::iterator it;
        if ((point -> GetClusterID() == 0) || (point -> GetClusterID() == 1))
        {
                for (it = main_p.begin(); it != main_p.end(); it++)
                {
                Point *p = *it;
                if (p -> GetClusterID() == 0)
//||(p->GetClusterID()==1))//si aun no tiene un cluster asignado
                {
                        float dist = distance(point, *it);
                //para cada punto calcula la dist con el point actual
                        cout << dist<< "\n";
                        if (dist <= epsilon)
                        // Si la distancia es menor o igual a eps
                        {
                                cluster.push_back(*it);
                                // anade este punto al cluster
                                tam=cluster.size();
                        }
                }
        }
}
```

```
        }
        return cluster;
        //regresa el cluster de los vecinos del point actual
}

/**
vector<Point*> density_connected
(vector<Point*> cl, vector<Point*> main_p,
int start_place_in_cluster, float epsilon, int min_points, int cl_ID)
*/

vector<Point*> density_connected(vector<Point*> cl, vector<Point*> main_p,
int start_place_in_cluster, float epsilon, int min_points, int cl_ID)
{
        vector<Point*> cluster;
        int clusterSizeBefore = cl.size();
        int clusterSizeAfter;
        //if(cl.size() >= min_points)
        //{
        cout << "found cluster = \n";
        for (unsigned int i=start_place_in_cluster; i<cl.size(); i++)
        //for que empieza desde el punto 1 clusterSizeBefore
        {
                cluster=density_reachable
                (cl[i],main_p, epsilon,min_points);
                //recibe los vecinos de cl[i]
                //print_data(cluster); //126.31
                vector<Point*>::iterator it1;
                for (it1=cluster.begin(); it1!=cluster.end(); it1++)
                //for para todos los vecinos de cl[i]
                {
```

```
vector<Point*>::iterator it2;
Point* point_tmp = *it1;
int quantity = 0;
//declaracion de quantity = 0
for (it2 = cl.begin(); it2 != cl.end(); it2++)
//for
{
        if (*it1 == *it2)
        //si cluster[i] == cl[j]
        {
                quantity++;
                //se incrementa contador
        }
}
if (quantity == 0)
                //si no hubo ninguno igual
                {
                        cl.push_back(point_tmp);
                        //introduce Point* point_tmp al cl
                }
        }
}

clusterSizeAfter = cl.size();
// calcula el tamanio del nuevo cluster
cout<<"clusterSizeAfter: "<<clusterSizeAfter<<endl;
if (clusterSizeBefore != clusterSizeAfter)
//si hubo un incremento
{
        density_connected(cl, main_p, clusterSizeBefore,
        epsilon, min_points, cl_ID);
```

```
}
else
//si no hubo un incremento
{
        vector<Point*>::iterator it1;
        vector<Point*>::iterator it2;
        if (clusterSizeAfter >= min_points)                    120
        // Se verifica si clusterSizeAfter >= min_points
        {
                //cout << "found cluster = \n";
                //update information in clusterID
                for (it1 = cl.begin(); it1 != cl.end(); it1++)
                //itera en todos los vecinos de cl
                {
Point* p = *it1;
                //se crea un punto para iterar
for (it2 = main_p.begin(); it2 != main_p.end(); it2++)         130
//vecinos de todos los puntos
{
        Point* p2 = *it2;
        if (p -> GetPointID() == p2 -> GetPointID())
        //pasa los puntos de
        {
                int id = p2 -> GetClusterID();
                //se asigna el id
                p2 -> SetClusterID(id + cl_ID);
                //se actualiza el id                             140
                *it2 = p2;
                //se actualiza
        }
}
```

```
                }
            }
            print_cluster(cl);
            // se imprime
        }
//}
        cluster.empty();
        //se limpia el cluster
        //returning updated list of points in main_p
        return main_p;
}
/**
int print_data(vector<Point*> cl)
Funcion que imprime un
*/
int print_data(vector<Point*> cl)
{
        vector<Point*>::iterator it;
        //iterador para recorrer el cluster
        int i=0;
        float * Miscoords= new float[4];
        //vector float
        for (it = cl.begin(); it != cl.end(); it++)
        //iteracion atravez del vector del cluster
        {
        Point* p = *it;
        cout <<"***** dato i= "<< ++i <<" *****"<<endl;
        //impresion de datos;

        Miscoords = p-> GetPointCoords();
        //OBTIENE LAS coordenadas del punto.
```

```
cout <<"PointID()= "<< p->GetPointID();
//impresion de datos;
cout <<"\tClusterID()= "<< p->GetClusterID()<<endl;
//impresion de datos;
cout <<"Coords[0]= "<< Miscoords[0];
//impresion de datos;
cout <<"\tcoords[1]= "<< Miscoords[1];
//impresion de datos;
cout <<"\tcoords[2]= "<< Miscoords[2];
//impresion de datos;
cout <<"\tcoords[3]= "<< Miscoords[3]<<"\n"<<endl;
//impresion de datos;
}
return i;
}
///** print_cluster(vector<Point*> cl)
void print_cluster(vector<Point*> cl)
{
vector<Point*>::iterator it;
for (it = cl.begin(); it < cl.end(); it++)
{
Point* point = *it;
float* pointcoord = new float[4];
pointcoord = point -> GetPointCoords();
int clID = point -> GetClusterID();
if (clID != 0)
{
cout << point->GetPointID() << "\t";
cout << pointcoord[0]<< "\t"<< pointcoord[1]<< "\t";
cout << pointcoord[2]<< "\t"<<endl;
cout << point->GetPointID() << "\t";
```

```
                cout << clID <<"\t"<< pointcoord[0]<<"\t"<< pointcoord[1] <<"\t"
                cout << pointcoord[2]<<"\t"<<pointcoord[3]<<"\t"<<endl;
                }
        }
        cout <<" "<< endl;
}
/**
void dbscan(vector<Point*> cl, float epsilon, int min_points)
*/

void dbscan(vector<Point*> cl, float epsilon, int min_points){
        int clust_ID = 2;
        vector<Point*>::iterator it;
        vector<Point*> cluster;
        int tam=0;
        for (it = cl.begin(); it != cl.end(); it++){
        //iterador por el vector de datos
                cout << "";
                cluster = density_reachable(*it, cl, epsilon, min_points);
                //VECTOR ALCANSABLES
                Point* point = *it;
                float* pointcoord = new float[4];
                pointcoord = point -> GetPointCoords();
                tam=cluster.size();
//              print_data(cluster);                    //impresion de  DATOS
                //cout << "it=\t"<< *it <<"\t";
                //cout <<pointcoord[0]<<"\t"<<pointcoord[1]<<"\t\t";
                //cout<< pointcoord[2]<<"\t\t"<< pointcoord[3]<< "\n";
//              print_data(cluster);                    //impresion de  DATOS
                cout <<"cluster.size() = "<< cluster.size();
                cout <<"min_points= "<< min_points; //
```

```
if (cluster.size() < min_points){       //cluster.size() < min_points
        vector<Point*>::iterator it1;
        vector<Point*>::iterator it2;                                    240
        for (it1 = cluster.begin(); it1 != cluster.end(); it1++){
                Point* p = *it1;
                for (it2 = cl.begin(); it2 != cl.end(); it2++){
                        Point* p2 = *it2;
                        if (p -> GetPointID() == p2 -> GetPointID()){
                                int id = p2 -> GetClusterID();
                                if (id == 0)
                                        p2 -> SetClusterID(id + 1);
                                *it2 = p2;
                        }                                                250
                }
        }
}
                cluster.empty();
        }
        else{
        cluster = density_connected(cluster, cl, 1, epsilon, min_points, clust_ID);
//                print_data(cluster);
        clust_ID = clust_ID + 1;
        }
    }                                                                    260
    cluster.empty();
}
//lgrind -i -lc texdbscan.cpp > dbscan.tex
```

Bibliografía

[1] Tom M. Mitchell, *Machine Learning*, McGraw-Hill international editions Computer Science Series, 1997.

[2] 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. http://arxiv.org/abs/astro-ph/0608638, 2006.

[3] Marco Frailis, Alessandro De Angelis, Vito Roberto, *Data Management and Mining in Astrophysical Databases*, http://arxiv.org/abs/cs/0307032v2.

[4] E. D. Feigelson, G. J. Babu, *Statistical Challenges in Modern Astronomy*, PHYSTAT2003.

[5] Farhan Feroz and M.P. Hobson, *Multimodal nested sampling: an efficient and robust alternative to MCMC methods for astronomical data analysis*, Astronomical Data Analysis 2007.

[6] Ray Norris, Heinz Andernach, *Astronomical Data Management*, XXVIth IAU General Assembly, August 2006.

[7] Kirk D. Borne, *Data Mining in Astronomical Databases*, ESO Symposia: Mining the Sky, pp. 671-673, 2001.

[8] J. Yoo, *Analysis of Digital POSS-II Catalogs Using Hierarchical Unsupervised Learning Algorithms*, in Astronomical Data Analysis Software and Systems V, eds. G. Jacoby, J. Barnes, A.S.P Conf. Ser., 101, 41.

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

[10] J. A. López Aguerri *Evolución de Galaxias en Cúmulos* Plan Nacional de Astronomía y Astrofísica. IAC

[11] Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, 2nd International Conference on Knowledge Discovery and Data Mining, (1996).

[12] Shashi Shekhar, Pusheng Zhang, Yan Huang, and Ranga Raju Vatsavai, *Trends in Spatial Data Mining*, in Data Mining: Next Generation Challenges and Future Directions, Hillol Kargupta and Anupam Joshi(eds), AAAI/MIT Press (2003).

[13] Beckmann N., Kriegel H.-P., Schneider R, and Seeger B. *The R*-tree: An Efficient and Robust Access Method for Points and Rectangles.* ACM SIGMOD International Conference on Management of Data. Atlantic City, NJ, 322-331. (1990).

[14] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jörg Sander. *OPTICS: Ordering Points To Identify the Clustering Structure.* ACM SIGMOD International conference on Management of data: 49-60 (1999).

[15] Raymond T. Ng and Jiawei Han *Efficient and Effective Clustering Methods for Spatial Data Mining*

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

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

[18] MICHALSKI R. S., BRATKO I. and KUBAT M. *Machine Learning and Data Mining.* John Wiley & Sons (1998).

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

[20] Jorg Sander, Martin Ester, Hans-Peter Kriegel, Xiaowei Xu, *Density-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, `http://citeseerx.ist.psu.edu/viewdoc/summary10.1.1.63.1629`, (1998).

[21] Antomn Guttman *R-trees a dynamic mdex structure for spatial searching*, Proc ACM SIGMOD Int Conf on Management of Data, 47-57, 1984.

[22] Sellis, T., N. Roussopoulos, C. Faloutsos. *The R*-tree: A dynamic index for multidimensional objects.* In Proc. 13th Int. Conference on Very Large Data Bases, pp. 507-518, (1987).

[23] Hartigan, J. A., Wong, M. A. *Algorithm AS 136: A K-Means Clustering Algorithm.* Journal of the Royal Statistical Society, Series C (Applied Statistics)100-108 (1979).

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

Algoritmo de agrupamiento basado en Densidad Aplicado a la Búsqueda de Subestructuras en Cúmulos de Galaxias

del (la) C.

Arturo RAYMUNDO AVILÉS

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