**INTRODUCTION**

CHAPTER 1

**INTRODUCTION**

Changes are inevitable and are difficult to observe. In the same way, geographical changes are always challenging, as they occur over vast regions and thus making it tricky to identify, especially when they are minor ones. On geographical pretext, a satellite on the move takes an image of a region over a period of time, which can either be a small one or a large period of time.

On a comparative context, in order to find the changes, we adopt many strategies, of which most of them result in giving the basic clusters. We also do have some advanced algorithms that would be more significant, while displaying the results. One such is the DBSCAN algorithm.

**REQUIREMENTS**

CHAPTER 2

**REQUIREMENTS**

The software requirements for our project are as follows

1. Operating System : 7.0/X/2003/2008
2. Matlab software

The hardware requirements for our project are as follows:

1. 1 GB Processor
2. Pentium 4 processor
3. Minimum 1 GB RAM
4. 50 GB HDD

**DATA MINING**

CHAPTER 3

**DATA MINING**

Data Mining is the process of analyzing data from different perspectives and summarizing it into useful information. This relatively young and interdisciplinary field of computer science is the process of extracting patterns from large data sets by combining methods from statistics and artificial intelligence with database management.

The actual data mining task is the automatic or semi-automatic analysis of large quantities of data to extract previously unknown interesting patterns such as groups of data records (cluster analysis), unusual records (anomaly detection) and dependencies (association rule mining). This usually involves using database techniques such as spatial indexes. These patterns can then be seen as a kind of summary of the input data, and used in further analysis or for example in machine learning and predictive analytics.

Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

* 1. **WHAT CAN DATA MINING DO?**

Data mining is primarily used today by companies with a strong consumer focus - retail, financial, communication, and marketing organizations. It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. And, it enables them to determine the impact on sales, customer satisfaction, and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data.

With data mining, a retailer could use point-of-sale records of customer purchases to send targeted promotions based on an individual's purchase history. By mining demographic data from comment or warranty cards, the retailer could develop products and promotions to appeal to specific customer segments.

The National Basketball Association (NBA) is exploring a data mining application that can be used in conjunction with image recordings of basketball games. The Advanced Scout software analyzes the movements of players to help coaches orchestrate plays and strategies.

### HOW DOES DATA MINING WORK?

While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks. Generally, any of four types of relationships are sought:

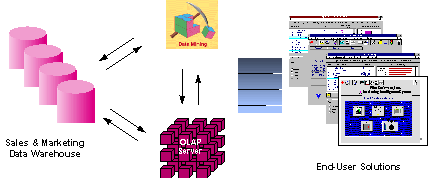
* **Classes**: Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
* **Clusters**: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
* **Associations**: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.
* **Sequential patterns**: Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

### ****ARCHITECTURE OF DATA MINING:****

To best apply these advanced techniques, they must be fully integrated with a data warehouse as well as flexible interactive business analysis tools. Many data mining tools currently operate outside of the warehouse, requiring extra steps for extracting, importing, and analyzing the data. Furthermore, when new insights require operational implementation, integration with the warehouse simplifies the application of results from data mining. The resulting analytic data warehouse can be applied to improve business processes throughout the organization, in areas such as promotional campaign management, fraud detection, new product rollout, and so on. Figure 1 illustrates architecture for advanced analysis in a large data warehouse.

An OLAP (On-Line Analytical Processing) server enables a more sophisticated end-user business model to be applied when navigating the data warehouse.

  The Data Mining Server must be integrated with the data warehouse and the OLAP server to embed ROI-focused business analysis directly into this infrastructure. An advanced, process-centric metadata template defines the data mining objectives for specific business issues like campaign management, prospecting, and promotion optimization.



*Figure 1 - Integrated Data Mining Architecture*

**PROCESS**

CHAPTER 4

**DATA MINING PROCESS**

How exactly is data mining able to tell you important things that you didn't know or what is going to happen next? The technique that is used to perform these feats in data mining is called modeling. Modeling is simply the act of building a model in one situation where you know the answer and then applying it to another situation that you don't. For instance, if you were looking for a sunken Spanish galleon on the high seas the first thing you might do is to research the times when Spanish treasure had been found by others in the past. You might note that these ships often tend to be found off the coast of Bermuda and that there are certain characteristics to the ocean currents, and certain routes that have likely been taken by the ship’s captains in that era. You note these similarities and build a model that includes the characteristics that are common to the locations of these sunken treasures. With these models in hand you sail off looking for treasure where your model indicates it most likely might be given a similar situation in the past. Hopefully, if you've got a good model, you find your treasure.

This act of model building is thus something that people have been doing for a long time, certainly before the advent of computers or data mining technology. What happens on computers, however, is not much different than the way people build models. Computers are loaded up with lots of information about a variety of situations where an answer is known and then the data mining software on the computer must run through that data and distill the characteristics of the data that should go into the model. Once the model is built it can then be used in similar situations where you don't know the answer. For example, say that you are the director of marketing for a telecommunications company and you'd like to acquire some new long distance phone customers. You could just randomly go out and mail coupons to the general population - just as you could randomly sail the seas looking for sunken treasure. In neither case would you achieve the results you desired and of course you have the opportunity to do much better than random - you could use your business experience stored in your database to build a model.

* 1. **PRE-PROCESSING:**

Before data mining algorithms can be used, a target data set must be assembled. As data mining can only uncover patterns actually present in the data, the target dataset must be large enough to contain these patterns while remaining concise enough to be mined in an acceptable timeframe. A common source for data is a data mart or data warehouse. Pre-processing is essential to analyze the multivariate datasets before data mining.

The target set is then cleaned. Data cleaning removes the observations containing noise and those with missing data.

Data mining involves six common classes of tasks:

* Anomaly detection (Outlier/change/deviation detection) – The identification of unusual data records, that might be interesting or data errors and require further investigation.
* Association rule learning (Dependency modeling) – Searches for relationships between variables. For example a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.
* Clustering – is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.
* Classification – is the task of generalizing known structure to apply to new data. For example, an email program might attempt to classify an email as legitimate or spam.
* Regression – Attempts to find a function which models the data with the least error.
* Summarization – providing a more compact representation of the data set, including visualization and report generation.

### SCIENCE & TECHNOLOGY:

In recent years, data mining has been used widely in the areas of science and engineering, such as bioinformatics, genetics, medicine, education and power engineering.

In the study of human genetics, sequence mining helps address the important goal of understanding the mapping relationship between the inter-individual variation in human DNA sequences and variability in disease susceptibility. In lay terms, it is to find out how the changes in an individual's DNA sequence affect the risk of developing common diseases such as cancer. This is very important to help improve the diagnosis, prevention and treatment of the diseases. The data mining method that is used to perform this task is known as multifactor dimensionality reduction.

In the area of electrical power engineering, data mining methods have been widely used for condition monitoring of high voltage electrical equipment. The purpose of condition monitoring is to obtain valuable information on the insulation's health status of the equipment. Data clustering such as self-organizing map (SOM) has been applied on the vibration monitoring and analysis of transformer on-load tap-changers (OLTCS). Using vibration monitoring, it can be observed that each tap change operation generates a signal that contains information about the condition of the tap changer contacts and the drive mechanisms. Obviously, different tap positions will generate different signals. However, there was considerable variability amongst normal condition signals for exactly the same tap position. SOM has been applied to detect abnormal conditions and to estimate the nature of the abnormalities.

Data mining methods have also been applied for dissolved gas analysis (DGA) on power transformers. DGA, as a diagnostics for power transformer, has been available for many years. Methods such as SOM has been applied to analyze data and to determine trends which are not obvious to the standard DGA ratio methods such as Duval Triangle.

A fourth area of application for data mining in science/engineering is within educational research, where data mining has been used to study the factors leading students to choose to engage in behaviors which reduce their learning and to understand the factors influencing university student retention. A similar example of the social application of data mining is its use in expertise finding systems, whereby descriptors of human expertise are extracted, normalized and classified so as to facilitate the finding of experts, particularly in scientific and technical fields. In this way, data mining can facilitate Institutional memory.

Other examples of applying data mining method applications are biomedical data facilitated by domain ontology, mining clinical trial data, traffic analysis using SOM, etc.

In adverse drug reaction surveillance, the Uppsala Monitoring Centre has, since 1998, used data mining methods to routinely screen for reporting patterns indicative of emerging drug safety issues in the WHO global database of 4.6 million suspected adverse drug reaction incidents.  Recently, similar methodology has been developed to mine large collections of electronic health records for temporal patterns associating drug prescriptions to medical diagnoses.

**SPATIAL DATA MINING**

CHAPTER 5

**SPATIAL DATA MINING**

The explosive growth of spatial data and widespread use of spatial databases

emphasize the need for the automated discovery of spatial knowledge. Spatial

data mining is the process of discovering interesting and previously unknown, but potentially useful patterns from spatial databases. The complexity of spatial data and intrinsic spatial relationships limits the usefulness of conventional data mining techniques for extracting spatial patterns. Efficient tools for extracting information from geo-spatial data are crucial to organizations which make decisions based on large spatial datasets, including NASA, the National Imagery and Mapping Agency (NIMA), the National Cancer Institute (NCI), and the United States Department of Transportation (USDOT). These organizations are spread across many application domains including ecology and environmental management, public safety, transportation, Earth science, epidemiology, and climatology.

Data mining, which is the partially automated search for hidden patterns in large databases, offers great potential benefits for applied GIS-based decision-making. Recently, the task of integrating these two technologies has become critical, especially as various public and private sector organizations possessing huge databases with thematic and geographically referenced data begin to realize the huge potential of the information hidden there. Among those organizations are:

* offices requiring analysis or dissemination of geo-referenced statistical data
* public health services searching for explanations of disease clusters
* environmental agencies assessing the impact of changing land-use patterns on climate change
* Geo-marketing companies doing customer segmentation based on spatial location.

#### CHALLENGES:

Geospatial data repositories tend to be very large. Moreover, existing GIS datasets are often splintered into feature and attribute components, that are conventionally archived in hybrid data management systems. Algorithmic requirements differ substantially for relational (attribute) data management and for topological (feature) data management. Related to this is the range and diversity of geographic data formats, that also presents unique challenges. The digital geographic data revolution is creating new types of data formats beyond the traditional "vector" and "raster" formats. Geographic data repositories increasingly include ill-structured data such as imagery and geo-referenced multi-media.

There are several critical research challenges in geographic knowledge discovery and data mining. Miller and Han offer the following list of emerging research topics in the field:

* **Developing and supporting geographic data warehouses** – Spatial properties are often reduced to simple aspatial attributes in mainstream data warehouses. Creating an integrated GDW requires solving issues in spatial and temporal data interoperability, including differences in semantics, referencing systems, geometry, accuracy and position.
* **Better spatio-temporal representations in geographic knowledge discovery** – Current geographic knowledge discovery (GKD) methods generally use very simple representations of geographic objects and spatial relationships. Geographic data mining methods should recognize more complex geographic objects (lines and polygons) and relationships (non-Euclidean distances, direction, connectivity and interaction through attributed geographic space such as terrain). Time needs to be more fully integrated into these geographic representations and relationships.
* **Geographic knowledge discovery using diverse data types** – GKD methods should be developed that can handle diverse data types beyond the traditional raster and vector models, including imagery and geo-referenced multimedia, as well as dynamic data types (video streams, animation).

In four annual surveys of data miners, data mining practitioners consistently identified that they faced three key challenges more than any others:

* Dirty Data
* Explaining Data Mining to Others
* Unavailability of Data / Difficult Access to Data

**SPATIAL DATA CLUSTERING**

CHAPTER6

**SPATIAL DATA CLUSTERING**

The spatial clustering methods, which are considered an important component of spatial data mining. The authors classify the methods into four categories: partitioning method, hierarchical method, density based method, and grid-based method.

Partitioning methods like the k-means, the k-medoids and EM clustering are methods which make uses of a technique called iterative reallocation to improve the clustering quality from an initial solution. These methods tend to find clusters that are of spherical shape and they are made for minimizing the distance from the data objects to their distance centers.

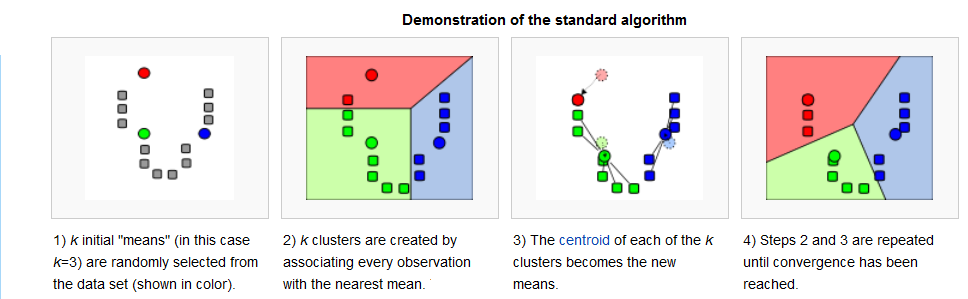
On the contrary of these, hierarchical clustering algorithms fixed the membership of a data object once it has been allocated to a cluster. BIRCH, CURE and CHAMELEON uses complex criteria for compressing and relocating data before merging clusters.

A third group of these methods is based on density of data points within a region to discover clusters. Belonging to this category are methods like DBSCAN, OPTICS and DENCLUE.

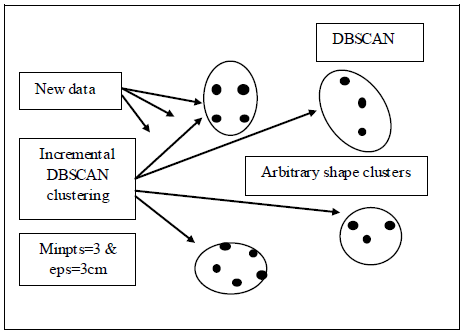
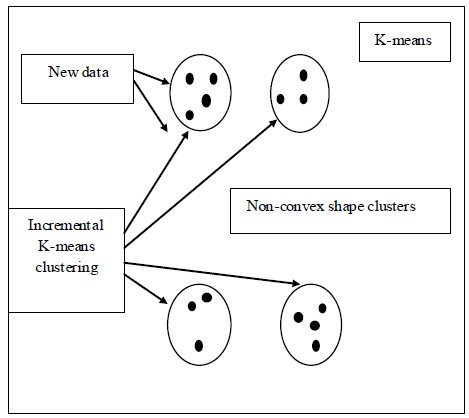
Finally, to increase the efficiency of clustering, grid-based clustering methods approximate the dense regions of the clustering space by quantizing it into a finite number of cells that contain more than a number of points as dense. Clusters are then formed by connecting the dense cells. Falling under this category are methods like STING, and CLIQUE.

* 1. **COMPARISON OF THE CLUSTERING METHODS:**
     1. **DBSCAN Vs k-MEANS:**

k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. Currently, the time complexity for implementing k - means is O (I \* k **\* d \***n), where I is the number of iterations. If we could use the KD-Tree data structure in the implementation, it can further reduce the complexity to O (I **\* k \***d \* log (n)).



* + 1. **ADVANTAGES OF DBSCAN OVER k-MEANS:**
* **CLUSTER SHAPES:**
  + When new data are coming into the old database, then sometimes new clusters are formed. In case of K-means clustering the cluster shapes must be fixed.
  + But in case of DBSCAN clustering, it discovers new clusters of arbitrary shape depends on its radius eps and Minpts. It does not follow any fixed shape like K-means clustering.
  + The following figure shows this difference clearly

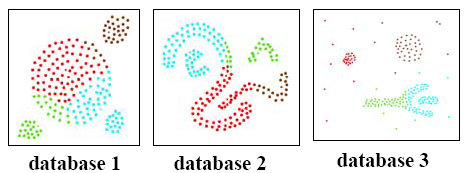


.

* **PREDEFINED CLUSTERS NUMBERS:**
  + In case of K-means clustering the total number of clusters must be predefined but in case of DBSCAN clustering the clusters are formed based on the new coming data, there is no need to predefine the number of clusters.
* **OUTLIERS HANDLING**
  + The K-means clustering algorithm is sensitive to noise and outlier data points because a small number of such data can substantially influence the mean value. But the DBSCAN algorithm has the ability to efficiently handle the noisy data even in the dynamic environment where the data are changed randomly.
    1. **DBSCAN Vs CLARANS:**

A *k*-medoids type algorithm called CLARANS (Clustering Large Applications based upon RANdomized Search) was proposed, which combines the sampling technique with PAM. However, unlike CLARA, CLARANS does not confine itself to any sample at any given time. While CLARA has a fixed sample at each stage of the search, CLARANS draws a sample with some randomness in each step of the search. Conceptually, the clustering process can be viewed as a search through a graph, where each node is a potential solution (a set of *k* medoids). Two nodes are *neighbors* (that is, connected by an arc in the graph) if their sets differ by only one object. Each node can be assigned a cost that is defined by the total dissimilarity between every object and the medoid of its cluster. At each step, PAM examines all of the neighbors of the current node in its search for a minimum cost solution. The current node is then replaced by the neighbor with the largest descent in costs.

**Clusters discovered by CLARANS**



**Clusters discovered by DBSCAN**

****

* **PERFORMNCE EVALUATION:**
* DBSCAN is more effective in discovering clusters of arbitrary shape than CLARANS.
* DBSCAN can identify noise whereas CLARANS cannot.
* Runtime of CLARANS is comparatively very large.
* CLARANS cannot be applied for large databases.
* Results show that DBSCAN outperforms CLARANS by a factor of at least 100 in terms of efficiency.

**MATLAB**

CHAPTER 7

**MATLAB**

**MATLAB** (**mat**rix **lab**oratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++,Java, and Fortran.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics. MATLAB is widely used in academic and research institutions as well as industrial enterprises.

* 1. **SYNTAX:**

The MATLAB application is built around the MATLAB language, and most use of MATLAB involves typing MATLAB code into the Command Window (as an interactive mathematical shell), or executing text files containing MATLAB code and functions.

### VARIABLES:

Variables are defined using the assignment operator, =. MATLAB is a weekly dynamically typed programming language. It is a weekly typed language because types are implicitly converted. It is a dynamically typed language because variables can be assigned without declaring their type, except if they are to be treated as symbolic objects, and that their type can change. Values can come from constants, from computation involving values of other variables, or from the output of a function.

### VECTORS/MATRICES:

As suggested by its name (a contraction of "Matrix Laboratory"), MATLAB can create and manipulate arrays of 1 (vectors), 2 (matrices), or more dimensions. In the MATLAB vernacular, a *vector* refers to a one dimensional (1×*N* or *N*×1) matrix, commonly referred to as an array in other programming languages. A *matrix* generally refers to a 2-dimensional array, i.e. an *m*×*n* array where *m* and *n* are greater than 1. Arrays with more than two dimensions are referred to as multidimensional arrays. Arrays are a fundamental type and many standard functions natively support array operations allowing work on arrays without explicit loops. Therefore the MATLAB language is also an example of array programming language.

* **GRAPHICS AND GRAPHICAL USER INTERFACE PROGRAMMING**:

MATLAB supports developing applications with graphical user interface features. It also has tightly-integrated graph-plotting features. For example the function *plot* can be used to produce a graph from two vectors *x* and *y*.

## INTERFACING WITH OTHER LANGUAGES:

## MATLAB can call functions and subroutines written in the C programming language or Fortran. A wrapper function is created allowing MATLAB data types to be passed and returned. The dynamically loadable object files created by compiling such functions are termed "MEX-files" (for MATLAB executable).

## Libraries written in Java, ActiveX or .NET can be directly called from MATLAB and many MATLAB libraries (for example XML or SQL support) are implemented as wrappers around Java or ActiveX libraries. Calling MATLAB from Java is more complicated, but can be done with MATLAB extension, which is sold separately by MathWorks, or using an undocumented mechanism called JMI (Java-to-Matlab Interface), which should not be confused with the unrelated Java Metadata Interface that is also called JMI.

As alternatives to the MUPAD based Symbolic Math Toolbox available from MathWorks, MATLAB can be connected to Maple or Mathematica.Libraries also exist to import and export MathML.

**DBSCAN ALGORITHM**

CHAPTER 8

**DBSCAN**

**DBSCAN** (for density-based spatial clustering of applications with noise) is a data clustering algorithm proposed by [Martin Ester](http://en.wikipedia.org/w/index.php?title=Martin_Ester&action=edit&redlink=1), Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996. It is a density-based clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding nodes. DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature. OPTICS can be seen as a generalization of DBSCAN to multiple ranges, effectively replacing the \varepsilon parameter with a maximum search radius.

* 1. **TERMINOLOGY USED:**

Density = number of points within a specified radius (Eps)

A point is said to be:

* + **CORE POINT:**

It is a point inside the cluster i.e., a point having an Eps neighborhood not less than MinPts.

* + **BORDER POINT:**

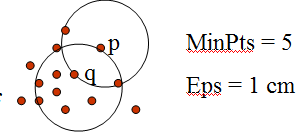
It is a point which lies on the border of clusters. It has its Eps neighborhood less than MinPts.

* + **NOISE POINT:**

It is any point that is not a core point or a border point.

* + **CONCEPTS: ε-NEIGHBOURHOD:**
* **ε- Neighborhood**: - Objects within a radius of ε from an object. (epsilon-neighborhood)
* **Core objects** :- ε-Neighborhood of an object contains at least MinPts of objects
* **DIRECTLY DENSITY- REACHABLE:**
* A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if p belongs to NEps(q)

core point condition: |NEps (q)| >= MinPts



* **DENSITY- REACHABLE:**
* An object *p* is density-reachable from *q* w.r.t ε and *MinPts* if there is a chain of objects *p1*,…,*pn*, with *p1*=*q*, *pn*=*p* such that *pi+1*is directly density-reachable from *pi* w.r.t ε and *MinPts* for all 1 <= *i* <= *n*
* Two border points of same cluster are not density reachable from each other because the core point condition might not hold for both of them.
* *q* is density-reachable from *p*
* **DENSITY – CONNECTIVITY:**
  + Object *p* is density-connected to object *q* w.r.t ε and *MinPts* if there is an object *o* such that both *p* and *q* are density-reachable from *o* w.r.t ε and *MinPts*

*P* and *q* are density-connected to each other by *r*

Density-connectivity is symmetric

* 1. **BASIC IDEA:**

[](http://en.wikipedia.org/wiki/File:DBSCAN-Illustration.svg)

DBSCAN's definition of a cluster is based on the notion of *density* reachability. Basically, a point qis directly density-reachable from a point p if it is not farther away than a given distance \varepsilon (i.e., is part of its \varepsilon-neighborhood) and if p is surrounded by sufficiently many points such that one may consider p and q be part of a cluster. q is called density-reachable (note: this is different from "directly density-reachable") from p if there is a sequence p_1,\ldots,p_n of points with p_1 = pand p_n = q where each p_{i+1} is directly density-reachable from p_i. Note that the relation of density-reachable is not symmetric (since q might lie on the edge of a cluster, having insufficiently many neighbors to count as a genuine cluster element), so the notion of *density-connected* is introduced: two points p and q are density-connected if there is a point o such that both p and qare density reachable fromo.

A cluster, which is a subset of the points of the database, satisfies two properties:

1. All points within the cluster are mutually density-connected.
2. If a point is density-connected to any point of the cluster, it is part of the cluster as well.
   1. **ALGORITHM:**

DBSCAN requires two parameters: \varepsilon (eps) and the minimum number of points required to form a cluster (minPts). It starts with an arbitrary starting point that has not been visited. This point's \varepsilon-neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized \varepsilon-environment of a different point and hence be made part of a cluster.

If a point is found to be a dense part of a cluster, its \varepsilon-neighborhood is also part of that cluster. Hence, all points that are found within the \varepsilon-neighborhood are added, as is their own \varepsilon-neighborhood when they are also dense. This process continues until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.

* 1. **PSEUDOCODE:**

DBSCAN(D, eps, MinPts)

C = 0

for each unvisited point P in dataset D

mark P as visited

N = regionQuery(P, eps)

if sizeof(N) < MinPts

mark P as NOISE

else

C = next cluster

expandCluster(P, N, C, eps, MinPts)

expandCluster(P, N, C, eps, MinPts)

add P to cluster C

for each point P' in N

if P' is not visited

mark P' as visited

N' = regionQuery(P', eps)

if sizeof(N') >= MinPts

N = N joined with N'

if P' is not yet member of any cluster

add P' to cluster C

## COMPLEXITY:

DBSCAN visits each point of the database, possibly multiple times (e.g., as candidates to different clusters). For practical considerations, however, the time complexity is mostly governed by the number of region Query invocations. DBSCAN executes exactly one such query for each point, and if an indexing structure is used that executes such a neighborhood query in , O(log n),an overall runtime complexity of O(n.log n) is obtained. Without the use of an accelerating index structure, the run time complexity is O(n^2). Often the distance matrix of size (n^2-n)/2  is materialized to avoid distance recomputations. This however also needs O(n^2) memory.

## ADVANTAGES:

1. DBSCAN does not require you to know the number of clusters in the data a priori, as opposed to k-means.
2. DBSCAN can find arbitrarily shaped clusters. It can even find clusters completely surrounded by (but not connected to) a different cluster. Due to the MinPts parameter, the so-called single-link effect (different clusters being connected by a thin line of points) is reduced.
3. DBSCAN has a notion of noise.
4. DBSCAN requires just two parameters and is mostly insensitive to the ordering of the points in the database. (Only points sitting on the edge of two different clusters might swap cluster membership if the ordering of the points is changed, and the cluster assignment is unique only up to isomorphism.)

## DISADVANTAGES:

1. DBSCAN can only result in a good clustering as good as its distance measure is in the function region Query(P,\varepsilon). The most common distance metric used is the Euclidean distance measure. Especially for high-dimensional data, this distance metric can be rendered almost useless due to the so called "Curse of dimensionality", rendering it hard to find an appropriate value for \varepsilon. This effect however is present also in any other algorithm based on the Euclidean distance.
2. DBSCAN cannot cluster data sets well with large differences in densities, since the minPts-\varepsilon combination cannot be chosen appropriately for all clusters then.

## PARAMETRIC EQUATIONS:

Every data mining task has the problem of parameters. Every parameter influences the algorithm in specific ways. For DBSCAN the parameters \varepsilon and MinPts are needed. The parameters must be specified by the user of the algorithms since other data sets and other questions require different parameters. An initial value for \varepsilon can be determined by a k-distance graph. As a rule of thumb, k can be derived from the number of dimensions in the data set D as k\geq D+1. However, larger values are usually better for data sets with noise.

Although this parameter estimation gives a sufficient initial parameter set the resulting clustering can turn out to be not the expected partitioning. Therefore research has been performed on incrementally optimizing the parameters against a specific target value.

OPTICS can be seen as a generalization of DBSCAN that replaces the \varepsilon parameter with a maximum value that mostly effects performance. MinPts then essentially becomes the minimum cluster size to find. While the algorithm is a lot easier to parameterize than DBSCAN, the results are a bit more difficult to use, as it will usually produce a hierarchical clustering instead of the simple data partitioning that DBSCAN produces.

**DESIGN**

CHAPTER 9

**INTRODUCTION**

System design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of system theory to product development.

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

* 1. **E-R DIAGRAMS:**
* The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
* The Entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the data modelling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions. The set of primary components that are identified by the ERD are:
* Data object
* Relationship
* Attributes
* Various types of indicators

A conceptual ER model may be used as the foundation for one or more [logical data models](http://en.wikipedia.org/w/index.php?title=Logical_data_models&action=edit&redlink=1). The purpose of the conceptual ER model is then to establish structural [metadata](http://en.wikipedia.org/wiki/Metadata) commonality for the [master data](http://en.wikipedia.org/wiki/Master_data) entities between the set of logical ER models. The conceptual data model may be used to form commonality relationships between ER models as a basis for data model integration. A logical ER model does not require a conceptual ER model especially if the scope of the logical ER model is to develop a single disparate information system. The logical ER model contains more detail than the conceptual ER model. In addition to master data entities, operational and transactional data entities are now defined. The details of each data entity are developed and the entity relationships between these data entities are established. The logical ER model is however developed independent of technology into which it will be implemented.

* 1. **UML DIAGRAMS:**

In the field of Software Engineering, the **Unified Modeling Language(UML)** is a standard visual specification language for object modeling. UML is a general purpose modeling language that includes a graphical notation used to create an abstract model of a system referred to as UML model. The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

* User Model View

This view represents the system from the user’s perspective. The analysis representation describes a usage scenario from the end-users perspective.

* + Structural model view

In this model the data and functionality are arrived from inside the system. This model view models the static structures.

* Behavioural Model View

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

* Implementation Model View

In this the structural and behavioural as parts of the system are represented as they are to be built.

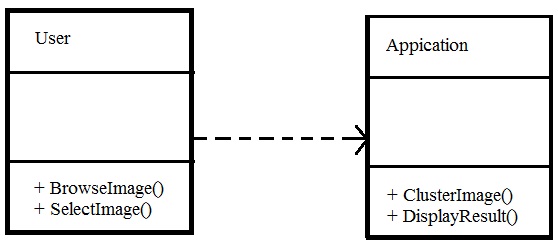
* Environmental Model View

In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

* + 1. **STRUCTURAL DIAGRAMS**

1. **CLASS DIAGRAM:**

Class diagrams identify the class structure of a system, including the properties and methods of each class. Also depicted are the various relationships that can exist between classes, such as an inheritance relationship. The Class diagram is one of the most widely used diagrams from the UML specification



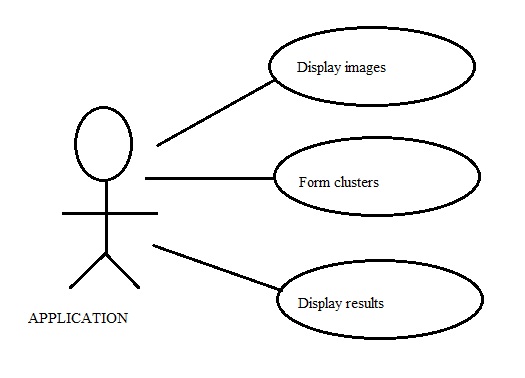
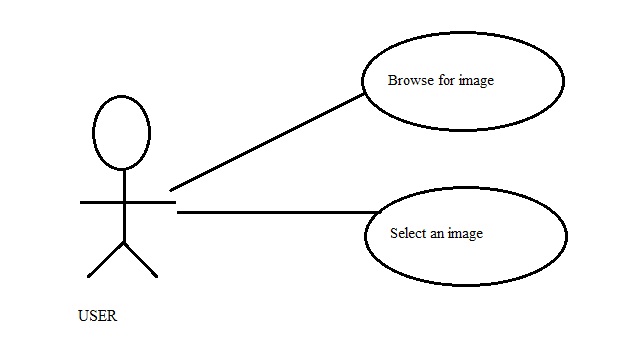
* + 1. **BEHAVIOURAL DIAGRAMS**

1. **USE CASE DIAGRAM:**

Use Case diagrams identify the functionality provided by the system (use cases), the users who interact with the system (actors), and the association between the users and the functionality. Use Cases are used in the Analysis phase of software development to articulate the high-level requirements of the system. The primary goals of Use Case diagrams include:

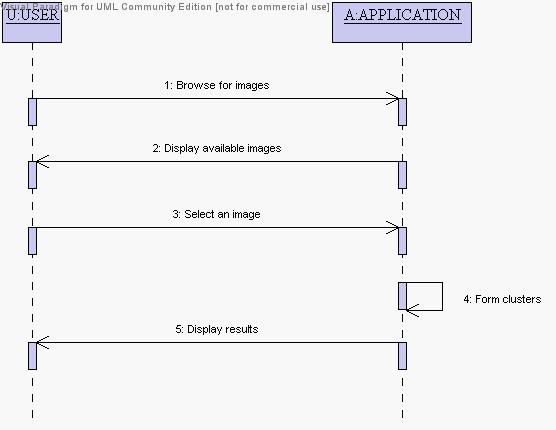
* Providing a high-level view of what the system does
* Identifying the users ("actors") of the system
* Determining areas needing human-computer interfaces

Use Cases extend beyond pictorial diagrams. In fact, text-based use case descriptions are often used to supplement diagrams, and explore use case functionality in more detail.



* + 1. **SEQUENCE DIAGRAM:**

Sequence diagrams document the interactions between classes to achieve a result, such as a use case. The Sequence diagram lists objects horizontally, and time vertically, and models these messages over time.

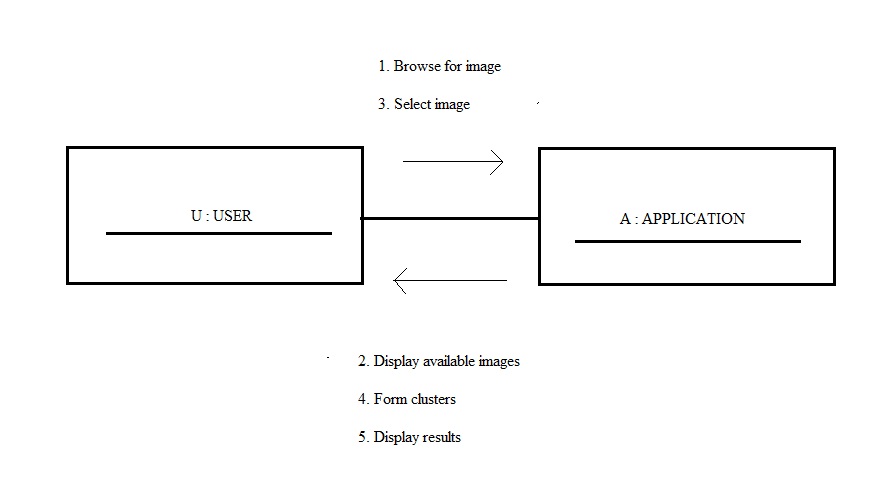


* + 1. **COLLABORATION DIAGRAM:**

Collaboration diagrams model the interactions between objects. This type of diagram is a cross between an object diagram and a sequence diagram. It uses free-form arrangement of objects which makes it easier to see all iterations involving a particular object.

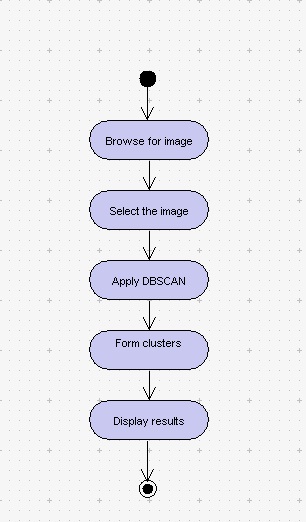
1. **COLLABORATION DIAGRAM:**

This diagram is an interaction diagram that stresses or emphasizes the structural organization of the objects that send and receive messages. It shows a set of objects, links between objects and messages send and received by those objects. There are used to illustrate the dynamic views of the system.

****

* + 1. **ACTIVITY DIAGRAMS:**

Activity diagram shows the flow from one activity to another within a system. The activities may be sequential or branching objects that act and are acted upon. These also show the dynamic view of the system.



**IMPLEMENTATION**

CHAPTER 10

**IMPLEMENTATION**

* 1. **INPUTS :**

The clustering using DBSCAN algorithm requires some initial parameters which are necessary for the whole process. Following are the major inputs to run the algorithm:

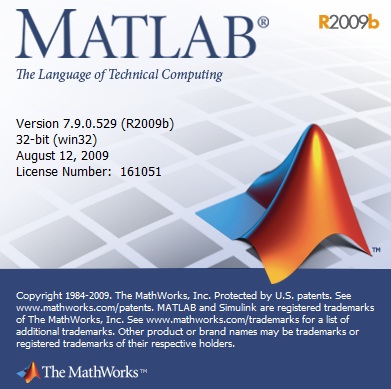
* The spatial image
* Threshold distance for number of clusters to be formed.
* Min points to from a cluster
  1. **OUTPUT:**

The output is the specified number of clusters where each cluster is viewed on GUI.

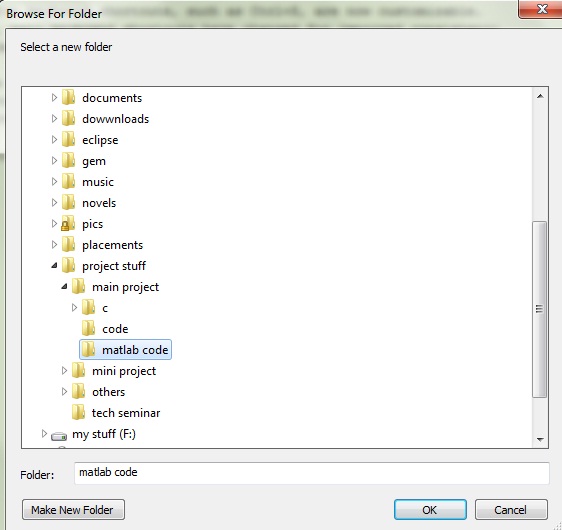
**10.3 STEPS OF EXECUTION:**

Our project has been completely implemented by using Matlab software. The steps of execution of our project are as follows.

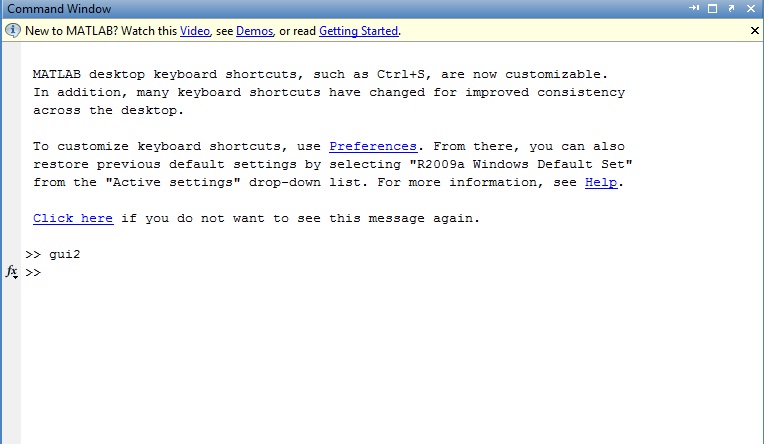
1. Start the Matlab software.



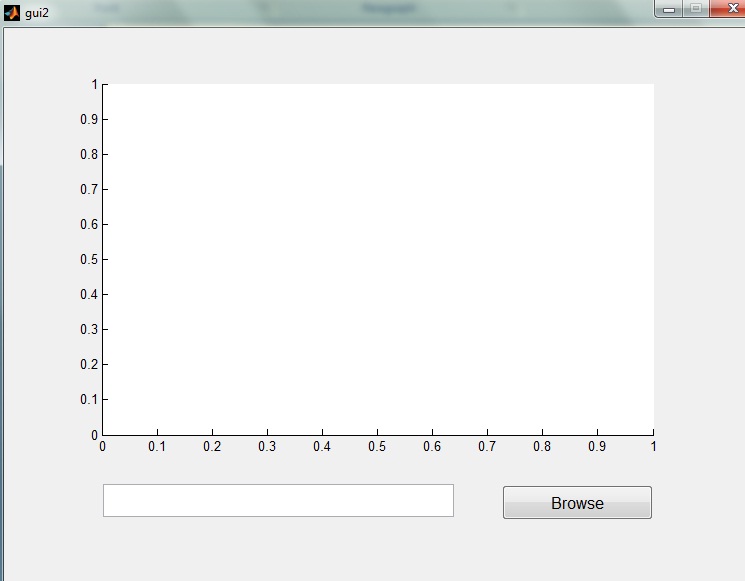
1. Choose the appropriate folder in which the code is saved.



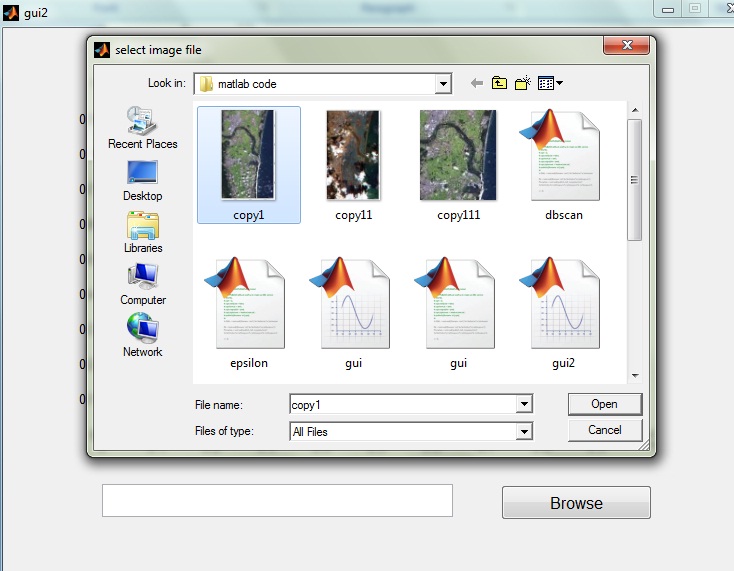
1. In the command prompt, type the name of the project i.e., “gui2” and press enter.



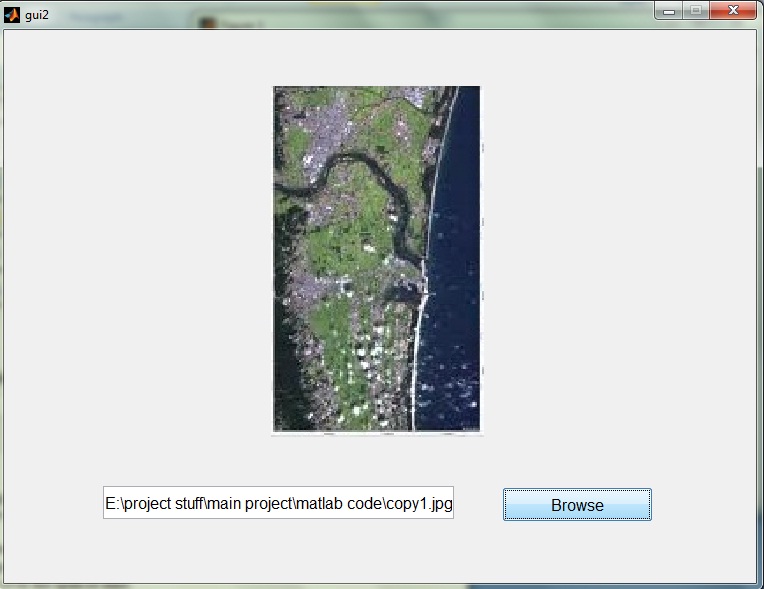
1. The code for the interface is thus executed and the interface of the project appears on the screen.

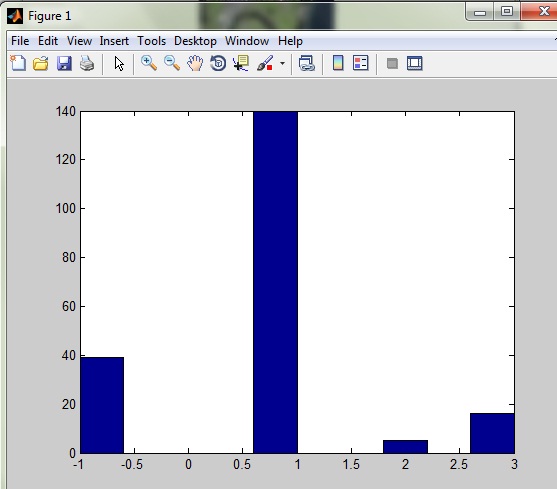


1. Choose the image by browsing the file system and click on “open”.

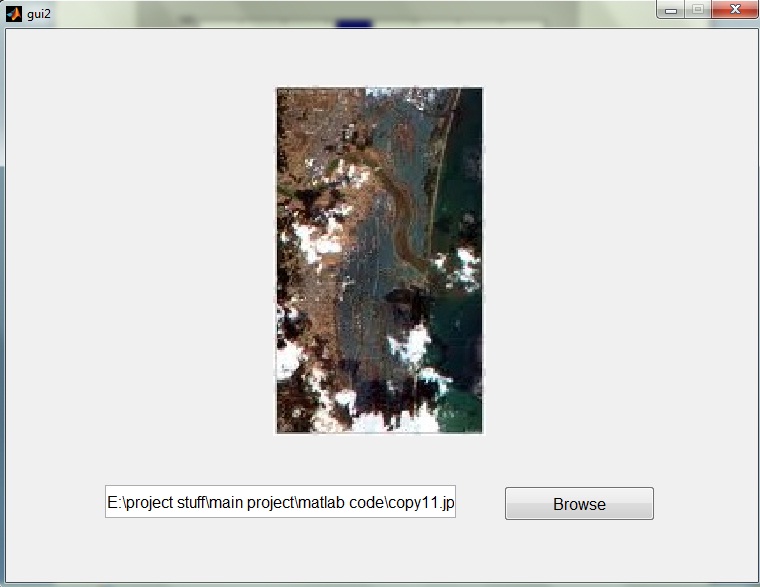


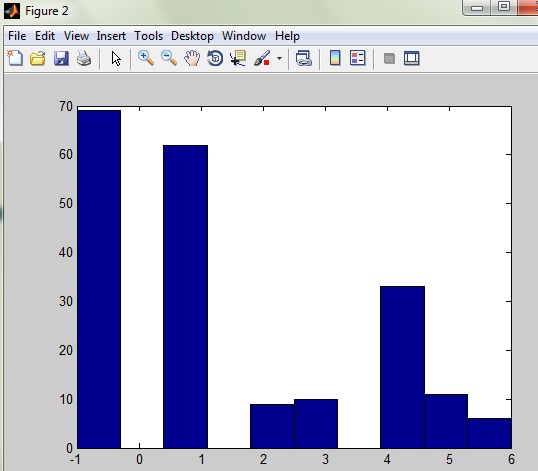
1. On selecting the image, the image is fed as input to the DBSCAN algorithm and thus the report of clusters is given out as output.



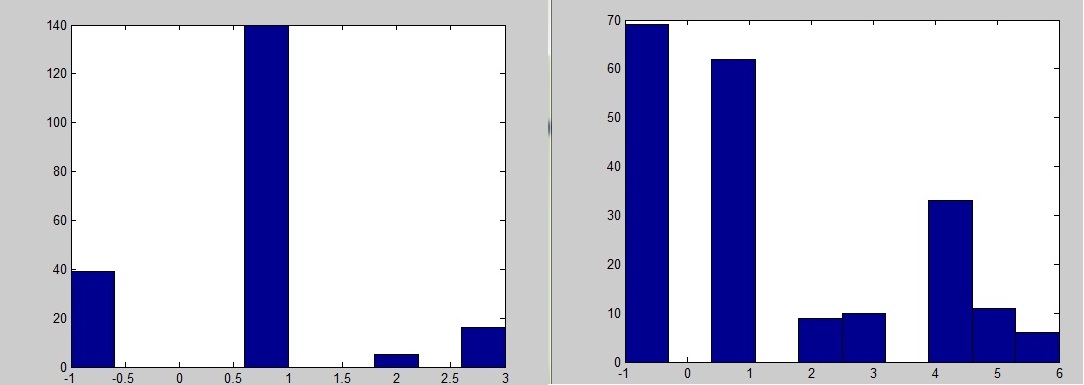


1. Similarly, on selecting the second image, another report shall be generated.





1. Thus the reports can be used to compare and identify the changes that have taken place over the span of time.



* The output is clustering of the image represented in the form of histograms.
* X-axis gives the cluster number.
* Y-axis gives the number of objects in each cluster.
* -1 cluster in the x-axis represents the outlier points which is defined as a set of points in the database not belonging to any cluster

**TESTING**

CHAPTER 11

**TESTING**

To test the software that we have developed, following are some of the available approaches.

* White box testing
* Black box testing
* Unit testing
* Integration testing
* Acceptance testing
* System testing
  1. **WHITE BOX TESTING:**

It involves the software testing for the following cases:

1. All independent paths with in a module have been exercised at least once

2. Exercise all logical decisions on their true and false sides

3. Execute all loops at their boundaries and within their operational bounds

4. Exercise internal data structures to ensure their validity.

To perform white box testing no. of techniques are available. Here we are giving some of the important techniques:

* Basis path testing
* Control structure testing
* **BASIS PATH TESTING:**

In this technique we will find out a basis set which ensures the execution of all the statements in the procedural design at least once.

The no. of basis sets that the given program will have can be found out using cycloramic complexity.

* **CYCLORAMIC COMPLEXITY:**

It measures the logical complexity of the program .It is the no. of independent paths in the basis set of a program. It is denoted by V (g).

V (g) =E-N+2

=P+1

=R

Where,

E is the no. of the edges

N is the no. of nodes

P is the no. of predicate nodes

R is the no. of regions in the given graph

* **CONTROL STRUCTURE TESTING:**

Note that basis path testing is one of the techniques used for testing how the control flows in a program. And some other are given below.

* Condition testing
* Data flow testing
* Loop testing
* **CONDITION TESTING:**

Here all the conditions in the program are tested for the no. of inputs and we will examine the condition.

* **DATAFLOW TESTING:**

In this technique we will examine how the declared data terms are being used and where they are used.

* **LOOP TESTING:**

Under this testing there are several types. Those are given below.

* Simple loops
* Concatenated loops
* Nested loops
* Unstructured loops
  1. **BLACK BOX TESTING:**

It mainly focuses on the functional requirements of the software. It enables the developer to derive sets of input conditions that will fully exercise all functional requirements for a program. It is a complementary approach that is likely to uncover different class of errors than white box testing methods.

It attempts to find the following types of errors:

1. Incorrect or missing functions

2. Interface errors

3. Errors in data structures

4. Behaviour or performance errors

5. Initialization and termination errors

To implement black box testing we will use the following techniques:

* Graph based testing
* Equivalence partitioning
* Boundary value analysis
* Comparison testing
* **GRAPH BASED TESTING:**

For the given program we will take the key actions as nodes and the events lead to the transition between those nodes are taken as edges we will analyze the functional behaviour of the program.

* **EQUIVALENCE PARTITIONING**:

1. It is the process of division of the input domain of a program into classes of data from which test cases can be derived

2. Equivalence partitioning strives to define a test case that uncovers classes of errors there by reducing the total no. of test cases that must be developed.

3. Test case design for equivalence partitioning is based on an evaluation of equivalence classes for an input condition (Input conditions range, specific value, member of a set, Boolean value).

* **BOUNDARY VALUE ANALYSIS:**

In this technique we will check all the loops at their boundary value for their correctness.

* **COMPARISON TESTING:**

Here we will run the given piece of code for multiple times and check for the consistency.

* 1. **UNIT TESTING:**

It focuses verification effort on the smallest unit of software design. In this testing, control paths are tested to uncover errors within the boundary of the module. The tests that occur as part of unit testing are given below:

* Module interface
* Local data structures
* Boundary conditions
* Independent paths
* Error handling paths
  1. **INTEGRATION TESTING:**

While combining different modules some errors may occur at the interfaces. To detect these errors, we will use the following tests:

* Regression testing
* Validation and verification testing
* Acceptance testing
* Alpha and beta testing
* **REGRESSION TESTING:**

When a new module is added to the existing software, the functional behaviour of it may change. To check whether it is properly functioning or not after the addition of the module, we will do this test.

* **VALIDATION AND VERIFICATION TESTING:**

Validation succeeds when software functions in a manner that can be reasonably expected by the customer.

Verification means are we building the product right i.e. all functions are being performed correctly.

* 1. **ACCEPTANCE TESTING:**

It is virtually impossible for the developer to foresee how the customer will really use a program. Instructions for use may be misinterpreted. Strange combinations of data may be regularly used. When the software is built for one customer, a series of tests of acceptance tests are conducted to enable the customer to validate all requirements. In fact acceptance testing can be conducted over of weeks or months.

* **ALPHA AND BETA TESTING:**

If software is developed as a product to be used by many customers, then we will use a process called alpha and beta testing. Alpha test is conducted at the developer site by the customer. Alpha tests are conducted in a controlled environment.

The beta test is conducted at one or more customer sites by the end user of the software. Unlike alpha testing, the developer is not present .Therefore the beta test is a live application of the software in an environment that can not be controlled by the developer.

* 1. **SYSTEM TESTING:**

Here we are testing the software at the system level. The following are the approaches for implementing at the system level:

* Recovery testing
* Security testing
* **RECOVERY TESTING:**

When an error occurs the system should be in a position to come into its normal state. Whether the software is performing that functionality or not we will test using this technique.

* **SECURITY TESTING:**

In this test we are checking the secrecy of the message while transmitting the message from one host to another is tested.

**CONCLUSION**

CHAPTER 12

**CONCLUSION**

Spatial Clustering using DBSCAN forms clusters which are more related and thus helps in producing useful (efficient) clusters compared to clusters produced by other clustering algorithms.

The clusters are formed for the two input representations (threshold distance and minimum points). Varied results were obtained for different similarity measure with a different representation.

This software has been developed using MATLAB and utmost care has been taken to satisfy user requirements while developing it.

The goals that are achieved by the software are:

* Less processing time and required information
* User Friendly

Intensive care has been taken by the development team to meet all the requirements of the client.

Finally we conclude that the project stands up in maintaining its sole motive of betterment to the society and work for a social cause.

**FUTURE ENHANCEMENTS**

CHAPTER 13

**FUTURE ENHANCEMENTS**

It is not possible to develop a system that meets all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

* + The System can be enhanced to increase look and feel of the Application.
  + Will try to scale the application by efficiently forming the clusters.

**BIBLIOGRAPHY**

**BIBLIOGRAPHY**

1. “Trends in Spatial Data Mining” by Shashi Shekhar, Pusheng Zhang, Yan Huang, Ranga Raju Vatsavai.
2. “Knowledge Discovery in Spatial Databases” by Martin Ester, Hans-Peter Kriegel, Jörg Sander.
3. “Visually Mining Through Cluster Hierarchies” by Stefan Brecheisen, Hans-Peter Kriegel, Peer Kroger, Martin Pfeie.
4. “Algorithms and Applications for Spatial Data Mining” by Martin Ester, Hans-Peter Kriegel, Jörg Sander.
5. “Spatial Data Mining: Accomplishments and Research Needs” by Shashi Shekhar and Pusheng Zhang.
6. “What's Special about Spatial Data Mining?” by Shashi Shekhar

Pusheng Zhang, Yan Huang, Ranga Raju Vatsavai.