**CHAPTER 1**

# INTRODUCTION

## Overview

With the increasing progress of both computer software and hardware, a large amount of data is generated and collected from different sources daily. Data collected contains useful information which is hidden so one can extract useful information from this data by discovering patterns in the data. However, the major problem for obtaining useful knowledge from data lies in the data itself. Therefore, Data mining plays an important role in extracting knowledge and useful patterns from the data.

Recently, data management and processing for wireless sensor networks (WSNs) has become a topic for research in several fields of computer science, such as distributed system, the database system, and the data mining [1]. We have proposed Clustering a data mining approach for making efficient clusters in wireless Sensor Network. Different sensor receives data and form clusters using k-prototype clustering algorithm discussed in chapter 2 and transmit this data to the base stations.

The nature of sensor data, certain special characteristics and limitations of WSNs prevent them from being directly applied to the traditional data mining techniques. Advances in wireless communication led to the development of low-power sensors and the deployment of large-scale sensor networks. The areas such as business, society, engineering and health care are generating data at an exponential rate. This has become possible due to tremendously rapid development of information technology. Data mining satisfies the need of extracting the useful information and knowledge from thus large scale data. [1].

In data mining, clustering is important technique. The clustering algorithms are widely used in image processing, customer segmentation, gene expression analysis [4] and text documents analysis [4] and text documents analysis etc. The aim of clustering is to divide a set of data objects into clusters such that data objects in the same cluster are more similar to each other than those in other clusters.

In real world, data sets usually contain both numeric and categorical attributes. However, most existing clustering algorithms assume all attributes are either numeric or categorical, examples of which include the k-mean, k-modes, fuzzy k-modes. When mixed data are encountered, most of them usually exploit transformation approaches to convert one type of the attributes to the other and then apply traditional single-type clustering [3]. However, in most cases, transformation scheme may result in loss of information, leading to undesired clustering outcomes [11].

A Technique known as Clustering is well established in data mining. This is used for grouping the data based on its similarity. It is used in various applications. One such application is Wireless Sensor Networks (WSNs). The WSNs usually consist of energy processing and storage limited devices known as sensor nodes. Sensor nodes are usually distributed over a certain area covered by the WSN to observe phenomena (light, temperature, humidity) by collecting data. Physical clustering of sensor nodes aims at grouping together sensor nodes that are sensing correlated data and selecting one of them as a representative, while turning others off [2]. This minimizes the energy consumption and thus extends the lifetime of the nodes.

In sensor networks, data mining is the process of extracting patterns and application-oriented models with acceptable accuracy from a rapid, continuous and possibly non ended flow of data streams from sensor networks. In this case, whole data cannot be stored and must be processed immediately. Data mining algorithm has to be sufficiently fast to process high-speed arriving data. The conventional data mining algorithms are meant to use the multistep techniques and multi-scan mining algorithms and handle the static data for analyzing static data-sets. Therefore,

Conventional data mining techniques could not be used for handling the high dimensionality, massive quantity and distributed nature of the data generated by the WSNs.

The k-means algorithm is famous for its efficiency in clustering large data sets but usually when working only on numeric values it is not used to cluster real world data containing categorical values. In this project we have used k-prototype algorithm which extend the k-means algorithm to categorical domains and domains with mixed numeric and categorical values. The k-modes algorithm uses a simple matching dissimilarity measure to deal with categorical objects, replaces the means of clusters with modes, and uses a frequency-based method to update modes in the clustering process to minimize the clustering cost function [3]. With the help of these extensions the k–modes algorithm enables the clustering of categorical data same as done in k-means algorithm. Further, the k-prototypes algorithm integrates the k-means and k -modes algorithms to allow for clustering objects described by mixed numeric and categorical attributes.

**1.2 Wireless Sensor Network**

A wireless sensor network (WSN**)** (sometimes called a wireless sensor and actor network(WSAN)are spatially distributed autonomous sensors to monitor physical or environmental conditions, such as pressure, sound, temperature etc. and to cooperatively pass their data to a main location. The more advanced networks are bi-directional which enables the control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. Today such networks are used in many consumer and industrial applications, such as machine health monitoring, industrial process monitoring and control and so on.

**1.3 Clustering**

Clustering can be considered the most important unsupervised learning problem. So, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data.  
The definition of clustering can be “the process of organizing objects into groups whose members are similar in some way” [6]. A collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters is known as clusters.

Clustering is mainly used to determine the intrinsic grouping in a set of unlabeled data. But what constitutes a good clustering? How can we can we decide that? It can be shown that there is no absolute “best” criterion which would be independent of the final aim of the clustering. Consequently, the user must supply this criterion, in such a way that the result of the clustering will suit their needs. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding “natural clusters” and describe their unknown properties (“natural” data types), in finding useful and suitable groupings (“useful” data classes) or in finding unusual data objects (outliers detection).

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early grouping is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

**1.4 K-mode**

The cause that the k-means algorithm cannot cluster categorical objects is its dissimilarity measure. These barriers can be removed by making the following modifications to the k-means algorithm:

1. Using a simple matching dissimilarity measure for categorical objects,

2. Replacing means of clusters by modes, and

3. Using a frequency-based method to find the modes

Like the k-means algorithm the k-modes algorithm also produces locally optimal solutions that are dependent on the initial modes and the order of objects in the data set

The k-modes algorithm extends the k-means algorithm for clustering categorical data by using a simple matching dissimilarity measure for categorical objects, modes instead of means for clusters and a frequency based method to update the modes using the k-means method to minimize the cost of clustering cost function [1].

**1.5 K-prototype**

In our project we used a k-prototypes algorithm for clustering mixed data. In our method, we first named the distribution centroid to represent the prototype of categorical attributes in a cluster [2]. Then we integrate the mean with distribution centroid to represent the prototype of a cluster with mixed attributes, and propose a new dissimilarity measure, which takes account of the significance of each attribute, to evaluate the dissimilarity between data objects and prototypes.

The algorithm is used to present an effective representation for the categorical attribute part in a mixed prototype since the mean is good enough for the numeric attribute part, and on the other hand to consider the significance of different attributes towards the clustering process.

It is straightforward to integrate the k-means and k-modes algorithms into the k-prototypes algorithm that is used to cluster the mixed-type objects. The k-prototypes algorithm is practically more useful because frequently encountered objects in real world databases are mixed-type objects.

**1.4 Motivation**

As the data collected from different sources consists of both categorical and numerical values.

So, the conventional approaches like k-means clustering does not fit well for data containing both kind of attributes. So we have used new approaches which are extensions for existing clustering algorithms. K-prototype algorithm is used for data containing both kind of attributes categorical and numerical. These new approaches uses data mining for finding similarity measures between the attribute values for different data objects.

The main objective of using these algorithm is to minimize the cost function discussed in later section.

As the wireless sensor network collects incoming data like temperature, humidity and rainfall on the sensors and sends the collected data to the base stations but the collected data varies a lot.

So efficient clusters are formed so that a cluster contains homogeneous data. Therefore data objects with attribute values which have the most similarity measure are put in the same cluster.

## 

## 1.6 Organization of the Report

The report is divided into 5 chapters. Explanation about each chapter is given below.

**Introduction chapter** gives an overview about the algorithms used for clustering in wireless sensor network such as k-means, k-modes and k-prototype clustering.

**Literature Survey** contains introduction about the basic idea about the data mining techniques like Data preprocessing and its associated techniques , Classification, association , clustering and different types of clustering like k-means , k-mode, k-prototype, LEACH , WLEACH.

**Proposed Method** chaptercontains the methodology used for the application of k-prototype algorithm for clustering data set based on similarity measure of data objects.

**Experimental Result and Analysis** chaptercontains the sample data set used for the application of k-prototype clustering algorithm.

**Conclusion and Future Work** chapter gives overall concluded results which shows that k-prototype being a good clustering algorithm for numerical and categorical values suffers from drawbacks like mapping of categorical values to binary values.

# CHAPTER 2

# LITERATURE SURVEY

## 

## 2.1 Data Mining

Data mining, or knowledge discovery, is the computer-assisted process of digging through and analyzing enormous sets of data and then extracting the meaning of the data. Data mining tools predict behaviors and future trends, allowing businesses to make proactive, knowledge-driven decisions. Data mining tools can answer business questions that traditionally were too time consuming to resolve.

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both [14]. 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. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

Data mining derives its name from the similarities between searching for valuable information in a large database and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material, or intelligently probing it to find where the value resides.

**2.2 Data Mining Techniques**

There are several major data mining technique have been developing and using in data mining projects recently including association, classification, clustering, prediction, sequential pattern and decision tree.

**2.2.1 Data Preprocessing**

Raw data is highly susceptible to noise, missing values, and inconsistency. The quality of

Data affects the data mining results. In order to help improve the quality of the data and

Consequently, of the mining results raw data is preprocessed so as to improve the efficiency and ease of the mining process. Data preprocessing is one of the most critical steps in a data mining process which deals with the preparation and transformation of the initial dataset.

Data preprocessing methods are divided into following categories:

1. Data cleaning

2. Data integration

3. Data transformation

4. Data reduction

**2.2.1.1 Data Cleaning**

Data cleaning routines work to “clean” the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies. If users believe the data are dirty, they are unlikely to trust the results of any data mining that has been applied to it. Furthermore, dirty data can cause confusion for the mining procedure, resulting in unreliable output. Although most mining routines have some procedures for dealing with incomplete or noisy data, they are not always robust. Instead, they may concentrate on avoiding over fitting the data to the function being modeled.

**2.2.1.2 Data Integration**

It is likely that your data analysis task will involve data integration, which combines data from multiple sources into a coherent data store, as in data warehousing. These sources may include multiple databases, data cubes, or flat files. There are a number of issues to consider during data integration. Schema integration can be tricky. How can real world entities from multiple data sources be 'matched up'? This is referred to as the entity identification problem. For example, how can the data analyst or the computer be sure that customer id in one database, and customer-number in another refer to the same entity? Databases and data warehouses typically have metadata that is, data about the data. Such metadata can be used to help avoid errors in schema integration. Redundancy is another important issue. An attribute may be redundant if it can be derived from another table, such as annual revenue. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

**2.2.1.3 Data Transformation**

In data transformation, the data are transformed or consolidated into forms appropriate for mining. Data transformation can involve the following:

1. Normalization, where the attribute data are scaled so as to fall within a small specified range, such as – 1.0 to 1.0, or 0 to 1.0.

2. Smoothing works to remove the noise from data. Such techniques include binning, clustering, and regression.

3. Aggregation, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.

4. Generalization of the data, where low level or 'primitive' (raw) data are replaced by higher level concepts through the use of concept hierarchies. For example, categorical attributes, like street, can be generalized to higher level concepts, like city or county. Similarly, values for numeric attributes, like age, may be mapped to higher level concepts, like young, middle-aged, and senior.

**2.2.1.4 Data Reduction**

Complex data analysis and mining on huge amounts of data may take a very long time, making such analysis impractical or infeasible. Data reduction techniques have been helpful in analyzing reduced representation of the dataset without compromising the integrity of the original data and yet producing the quality knowledge. The concept of data reduction is commonly understood as either reducing the volume or reducing the dimensions (number of attributes). There are a number of methods that have facilitated in analyzing a reduced volume or dimension of data and yet yield useful knowledge. Certain partition based methods work on partition of data tuples. That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results.

Strategies for data reduction include the following:

1. Data cube aggregation, where aggregation operations are applied to the data in the construction of a data cube.

2. Dimension reduction, where irrelevant, weakly relevant, or redundant attributes or dimensions may be detected and removed.

3. Data compression, where encoding mechanisms are used to reduce the data set size. The methods used for data compression are wavelet transform and Principal Component Analysis.

4. Numerosity reduction, where the data are replaced or estimated by alternative, smaller data representations such as parametric models (which need store only the model parameters instead of the actual data e.g. regression and log-linear models), or nonparametric methods such as clustering, sampling, and the use of histograms.

5. Discretization and concept hierarchy generation, where raw data values for attributes are replaced by ranges or higher conceptual levels. Concept hierarchies allow the mining of data at multiple levels of abstraction, and are a powerful tool for data mining.

**2.2.2 Association**

Association is one of the best known data mining technique. In association, a pattern is discovered based on a relationship between items in the same transaction. That’s is the reason why association technique is also known as relation technique*.* The association technique is used in market basket analysisto identify a set of products that customers frequently purchase together.

Retailers are using association technique to research customer’s buying habits. Based on historical sale data, retailers might find out that customers always buy crisps when they buy beers, and therefore they can put beers and crisps next to each other to save time for customer and increase sales.

**2.2.3 Classification**

Classification is a classic data mining technique based on machine learning. Basically classification is used to classify each item in a set of data into one of predefined set of classes or groups. Classification method makes use of mathematical techniques such as decision trees, linear programming, neural network and statistics. In classification, we develop the software that can learn how to classify the data items into groups.

Classification is a task in Data mining. Data mining, as indicated before, is a machine learning discipline, and is inspired by pattern recognitions, which is a branch of science, of which one of its goals is to classify objects into a number of categories referred to as classes. Objects refer to compact data units specific to a particular problem, which is in general, known as patterns.

Classification prediction encompasses two levels: classifier construction and the usage of the classifier constructed. The former is concerned with the building of a classification model by describing a set of predetermined classes from a training set as a result of learning from that dataset. Each sample in the training set is assumed to belong to a predefined class, as determined by the class attribute label. The model is represented as classification rules, decision trees, or mathematical formula. The later involves the use of a classifier built to predict or classify unknown objects based on the patterns observed in the training set.

The entire process begins with collection of evidence acquired from various data sources or warehouses. In the ideal situation, the data should be of low-dimensionality, independent and

discriminative so that its values are very similar to characteristics in the same class but very different in features from different classes. Raw data hardly satisfies these conditions and therefore a set of procedures called feature generation, extraction and selection is required to provide a relevant input for classification system.

For example, we can apply classification in application that “given all records of employees who left the company, predict who will probably leave the company in a future period.” In this case, we divide the records of employees into two groups that named “leave” and “stay”. And then we can ask our data mining software to classify the employees into separate groups.

**2.2.4 Clustering**

Clustering is a data mining technique that makes meaningful or useful cluster of objects which have similar characteristics using automatic technique. The clustering technique defines the classes and puts objects in each class, while in the classification techniques, objects are assigned into predefined classes [3].

Clustering is a division of data into groups of similar objects. Representing the data by fewer clusters necessarily loses certain fine details, but achieves simplification. It models data by its clusters. Data modeling puts clustering in a historical perspective rooted in mathematics, statics, and numerical analysis. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data mining applications such as scientific data exploration, information retrieval and text mining, spatial database applications, web analysis, CRM, marketing, medical diagnostics, computational biology, and many others.

To make the concept clearer, we can take book management in library as an example. In a library, there is a wide range of books in various topics available. The challenge is how to keep those books in a way that readers can take several books in a particular topic without hassle. By using clustering technique, we can keep books that have some kinds of similarities in one cluster or one shelf and label it with a meaningful name. If readers want to grab books in that topic, they would only have to go to that shelf instead of looking for entire library.

**2.2.5 Prediction**

The prediction, as it name implied, is one of a data mining techniques that discovers relationship between independent variables and relationship between dependent and independent variables. For instance, the prediction analysis technique can be used in sale to predict profit for the future if we consider sale is an independent variable, profit could be a dependent variable. Then based on the historical sale and profit data, we can draw a fitted regression curve that is used for profit prediction.

**2.2.6 Sequential Pattern**

Sequential patterns analysis is one of data mining technique that seeks to discover or identify similar patterns, regular events or trends in transaction data over a business period.

In sales, with historical transaction data, businesses can identify a set of items that customers buy together a different times in a year. Then businesses can use this information to recommend customers buy it with better deals based on their purchasing frequency in the past.

**2.2.7 Decision Trees**

Decision tree is one of the most used data mining techniques because its model is easy to understand for users. In decision tree technique, the root of the decision tree is a simple question or condition that has multiple answers. Each answer then leads to a set of questions or conditions that help us determine the data so that we can make the final decision based on it.

**2.2.8 Clustering Techniques**

Clustering methods can be divided into two basic types: hierarchical and partitional clustering. Within each of the types there exists a wealth of subtypes and different algorithms for finding the clusters.

Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones, or by splitting larger clusters. The clustering methods differ in the rule by which it is decided which two small clusters are merged or which large cluster is split. The end result of the algorithm is a tree of clusters called a dendrogram, which shows how the clusters are related. By cutting the dendrogram at a desired level a clustering of the data items into disjoint groups is obtained.

Partitional clustering, on the other hand, attempts to directly decompose the data set into a set of disjoint clusters. The criterion function that the clustering algorithm tries to minimize may emphasize the local structure of the data, as by assigning clusters to peaks in the probability density function, or the global structure. Typically the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters.

**2.2.8.1 k-means Clustering**

The k-means algorithm one of the mostly used clustering algorithms, is classified as a partitioned or nonhierarchical clustering method. Given a set of numeric objects X and an integer number k (≤n), the k-means algorithm searches for a partition of X into k clusters that minimizes the within groups sum of squared errors (WGSS). This process is often formulated as the following mathematical program problem P [1].

The k-means algorithm has the following important properties:

1. It is efficient in processing large data sets.

2. It often terminates at a local optimum.

3. It works only on numeric values.

4. The clusters have convex shapes.

**2.2.8.2 k-mode Clustering**

In principle the formulation of problem P in Section 3 is also valid for categorical and mixed type objects. The cause that the k-means algorithm cannot cluster categorical objects is its dissimilarity measure and the method used to solve problem P2. These barriers can be removed by making the following modifications to the k-means algorithm:

1. Using a simple matching dissimilarity measure for categorical objects,

2. Replacing means of clusters by modes, and

3. Using a frequency-based method to find the modes to solve problem P2

**2.2.8.2.1 Dissimilarity Measure**

Let X, Y be two categorical objects described by m categorical attributes. The dissimilarity measure between X and Y can be defined by the total mismatches of the corresponding attribute categories of the two objects. The smaller the number of mismatches is, the more similar the two objects. This measure is often referred to as simple matching. Formally,

d2(X,Y) = (2.3)

Where

δ(xj,yj) = (2.4)

**2.2.8.2.2 Mode of a set**

Let X be a set of categorical objects described by categorical attributes, A1, A2,..., Am. Definition 1. A mode of X = {X1, X2,..., Xn} is a vector Q = [q1, q2,..., qm] that minimizes

D(X, Q) = (2.5)

When (5) is used as the dissimilarity measure for categorical objects, the cost function (1) becomes

P(W, Q) = (2.6)

where wi ∈ W and Ql = [q1,q2,..., qn] ∈ Q. To minimize the cost function the basic k-means algorithm can be modified by using the simple matching dissimilarity measure to solve P1, using modes for clusters instead of means and selecting modes according to Theorem 1 to solve P2. In the basic algorithm we need to calculate the total cost P against the whole data set each time when a new Q or W is obtained. To make the computation more efficient we use the following algorithm:

1. Select k initial modes, one for each cluster.

2. Allocate an object to the cluster whose mode is the nearest to it. Update the mode of the cluster after each allocation.

3. After all objects have been allocated to clusters, retest the dissimilarity of objects against the current modes. If an object is found such that its nearest mode belongs to another cluster rather than its current one, reallocate the object to that cluster and update the modes of both clusters.

4. Repeat 3 until no object has changed clusters after a full cycle test of the whole data set.

Like the k-means algorithm the k-modes algorithm also produces locally optimal solutions that are dependent on the initial modes and the order of objects in the data set. In our current implementation of the k-modes algorithm we include two initial mode selection methods. The first method selects the first k distinct records from the data set as the initial k modes.

**2.2.8.3 k-prototype Clustering**

The algorithm is based on the k-means paradigm but removes the numeric data only limitation whilst preserving its efficiency. The algorithm clusters objects with numeric and categorical attributes in a way similar to k-means. Because objects are clustered against k prototypes instead of k means of clusters, called as k-prototype algorithm.

K-prototype is used to dynamically update the k prototypes in order to maximize the intra cluster similarity of objects. The object similarity measure is derived from both numeric and categorical attributes. When applied to numeric data the algorithm is identical to k-means.

**2.2.8.3.1 Mathematical Preliminaries**

Let X = {X1 ,X2 ,...Xn } denote a set of n objects and X i = [xi1, xi2 ,...,x im] be an object represented by m attribute values. Let k be a positive integer. The objective of clustering X is to find a partition which divides objects in X into k disjoint clusters. For a given n, the number of possible partitions is definite but extremely large. It is impractical to investigate every partition in order to find a better one for a classification problem. A common solution is to choose a clustering criterion to guide the search for a partition. A clustering criterion is called a cost function.

**2.2.8.3.1.1 Cost Function**

The widely used cost function is the trace of the within cluster dispersion matrix.

One way to define this cost function is:

E = (2.7)

Here, Q l = [ q i1 , qi2 , ... , qim ] is the representative vector or prototype for cluster l, and yil is an element of a partition matrix . D is a similarity measure often defined as the square Euclidean distance.

Y has the following two properties, (1) 0 ≤ yil ≤ 1 and (2) ∑ yil = 1 . Y is called a hard partition if yil ∊{0,1} . Otherwise, it is a fuzzy partition 2 . In a hard partition, ytl = 1 indicates that object Xi is assigned to cluster l by Y.

The inner term E l = ∑ yil d ( X i , Q l ) in Eq. (2.7) is the total cost of assigning X to cluster l, i.e., the total dispersion of objects in cluster l from its prototype Q l . E l is minimized if

qij = for j=1,….,m (2.8)

where n l = ∑ yil is the number of objects in cluster l.

When X has categorical attributes, we can introduce a similarity measure as

d(X1,Q1) = + (2.9)

where δ (p,q)=0 for p=q and δ (p,q)=1 for p≠q. and are values of numeric attributes, whereas and are values of categorical attributes for object i and the prototype of cluster l. m r and m c are the numbers of numeric and categorical attributes. γ 1 is a weight for categorical attributes for cluster l.

We can rewrite E l as

E1 = +γ1  (2.10)

where is the total cost on all numeric attributes of objects in cluster l. is minimized if is calculated by Eq. (2.2). Let Cj be the set containing all unique values in the categorical attribute j and p( cj ∊ Cj | l ) the probability of value c j occurring in cluster l. in Eq. (2.10) can be rewritten as

= (2.11)

where n1 is the number of objects in cluster 1.

**2.2.8.3.2 Similarity Measure**

The cost function Eq. (2.11) is defined on Eq. (2.9), which is a combined similarity measure on both numeric and categorical attributes between objects and cluster prototypes. The similarity measure on numeric attributes is the square Euclidean distance whereas the similarity measure on categorical attributes is the number of mismatches between objects and cluster prototypes. Weight γ l is introduced to avoid favoring either type of attribute.

The influence of weight γ l in clustering can be illustrated by Figure 1. Assume the triangles and diamonds represent a set of objects described by a categorical and two numeric attributes. Triangle and diamond represent two values of the categorical attribute whereas numeric attribute values are reflected by locations of the objects. These objects are partitioned into two clusters.

### **Fig 2.1 Example of similarity measure**

If γ l =0, clustering only depends on numeric attributes, i.e., locations of the objects. The result will be the two clusters separated by the vertical dashed line. If γ l > 0, then object c may change to the right cluster because it is close to that cluster and its categorical value is the same as that of the majority of objects in that cluster. Similarly, object d may change to the left cluster. However, object a may still stay in the left cluster because it is too far to the right, even though it has a categorical value as same as that of the majority objects in that cluster. Similarly, object e may still be in the right cluster. Object b becomes uncertain, depending on whether γ l is biased towards numeric or categorical attributes. If γ l is biased to the categorical attribute, object b may change to the right cluster. Otherwise, it can stay in the left one. The choice of γ l is dependent on distributions of numeric attributes. Generally speaking, γ l is related to σ l, the average standard deviation of numeric attributes in cluster l. In practice, σ l can be used as a guidance to determine γ l. However, since σ l is unknown before clustering, the overall average standard deviation σ of numeric attributes can be used for all σ l. In an iterative algorithm, σ l can be calculated from a preceding clustering result.

# CHAPTER 3

# PROPOSED METHOD

## 3.1 Application of k-means Clustering on a Data Set

The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step.

After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more.

We have taken following eight points (with (x, y) representing locations) group them into three clusters A1(2, 10) A2(2, 5) A3(8, 4) A4(5, 8) A5(7, 5) A6(6, 4) A7(1, 2) A8(4, 9). Initial cluster centers are: A1(2, 10), A4(5, 8) and A7(1, 2). The distance function between two points *a=(x1, y1)* and *b=(x2, y2)* is defined as: *ρ(a, b) = |x2 – x1| + |y2 – y1|* .

Solution:

Iteration 1

First we list all points in the first column of the table above. The initial cluster centers – means, are (2, 10), (5, 8) and (1, 2) - chosen randomly. Next, we will calculate the distance from the first point (2, 10) to each of the three means, by using the distance function:

**Table 3.1: Distance of points from the initial centers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | (2, 10) | (5, 8) | (1, 2) |  |
|  | **Point** | **Dist Mean 1** | **Dist Mean 2** | **Dist Mean 3** | **Cluster** |
| A1 | (2, 10) | 0 | 5 | 9 | 1 |
| A2 | (2, 5) | 5 | 6 | 4 | 3 |
| A3 | (8, 4) | 12 | 7 | 9 | 2 |
| A4 | (5, 8) | 5 | 0 | 10 | 2 |
| A5 | (7, 5) | 10 | 5 | 9 | 2 |
| A6 | (6, 4) | 10 | 5 | 7 | 2 |
| A7 | (1, 2) | 9 | 10 | 0 | 3 |
| A8 | (4, 9) | 3 | 2 | 10 | 2 |

After first iteration

Cluster 1 contains the points (2, 10)

Cluster 2 contains the points (8, 4) (5, 8) (7, 5) (6, 4) (4, 9)

Cluster 3 contains the points (2, 5) (1, 2)

Next, we need to re-compute the new cluster centers (means). We do so, by taking the mean of all points in each cluster.

For Cluster 1, we only have one point A1(2, 10), which was the old mean, so the cluster center remains the same.

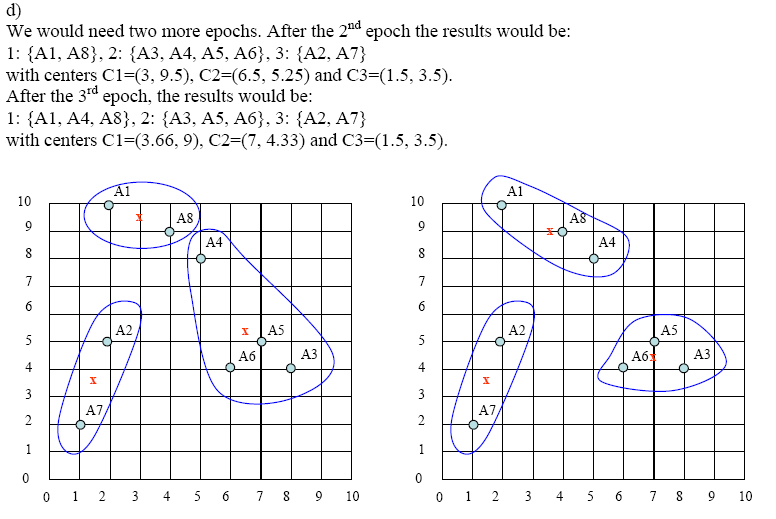
For Cluster 2, we have ( (8+5+7+6+4)/5, (4+8+5+4+9)/5 ) = (6, 6)

For Cluster 3, we have ( (2+1)/2, (5+2)/2 ) = (1.5, 3.5)



**Fig 3.1 Clusters of points using k-mean**

The initial cluster centers are shown in red dot. The new cluster centers are shown in red x. That was Iteration1 (epoch1). Next, we go to Iteration2 (epoch2), Iteration3, and so on until the means do not change anymore. In Iteration2, we basically repeat the process from Iteration1 this time using the new means we computed.



**Fig 3.2 Final cluster of points taken**

But due to drawbacks of k-means algorithm that it works with only numerical value we need to have a technique capable of working with both numerical and categorical values. K-prototype solves this purpose by applying data mining and forming clusters based on the both numerical and categorical values .

## 3.2 Clustering mixed attributes using k-prototype algorithm

We are basically using k-prototype algorithm to cluster dataset containing both numeric and categorical values.

K-prototype is an extension of k-mean algorithm. So the basic ideology of k-prototype algorithm is same as k-mean algorithm. The k-means based methods are efficient for processing large data sets, thus very attractive for data mining. The major handicap for them is that they are often limited to numeric data. The reason is that these algorithms optimize a cost function defined on the Euclidean distance measure between data points and means of clusters. Minimizing the cost function by calculating means limits their use to numeric data.

The algorithm is based on the k-means paradigm but removes the numeric data only limitation whilst preserving its efficiency. The algorithm clusters objects with numeric and categorical attributes in a way similar to k-means. Because objects are clustered against k prototypes instead of k means of clusters, we call it the k-prototypes algorithm. We have developed a method to dynamically update the k prototypes in order to maximize the intra cluster similarity of objects. The object similarity measure is derived from both numeric and categorical attributes. When applied to numeric data the algorithm is identical to k-means.

Basic k-prototype algorithm is as follows:

1. Given a dataset X select k initial prototypes one for each cluster.

2. Allocate each object in ***X*** to a cluster whose prototype is the nearest to it.

3. Update the prototype of the cluster after each allocation. After all objects have been allocated to a cluster, retest the similarity of objects against the current prototypes.

If an object is found such that its nearest prototype belongs to another cluster rather than its current one, reallocate the object to that cluster and update the prototypes of both clusters.

4 Repeat (3) until no object has changed clusters after a full cycle test of ***X***.

The algorithm is built upon three processes, initial prototypes selection, initial allocation*,* and re-allocation. The first process simply randomly selects k objects as the initial prototypes for clusters. The second process initializes different points to cluster. Starting from a set of initial cluster prototypes, this process assigns each object to a cluster and updates the cluster prototype accordingly after each assignment. In Figure 2, X[i] represents object i and X[i,j] the value of attribute j for object i. Oprototypes[] and Cprototypes[] store the numeric and categorical attribute partsof cluster prototypes respectively. Oprototypes[i,j] and Cprototypes[i,j] are two

corresponding numeric and categorical elements of the prototype for cluster *i*. **Distance**() is a square Euclidean distance function. Clustership[] and ClusterCount[] record cluster membership of objects and numbers of objects in clusters. SumInCluster[] sums up numeric values of objects in clusters and is used to update numeric attributes of cluster prototypes. FrequencyInCluster[] records frequencies of different values of categorical attributes in clusters. The function **HighestFreq**() returns the value with the highest frequency.

The reallocation process is similar to the initial allocation process except that after reallocation of an object, prototypes for both the previous and current clusters of the object are updated. Variable moves records the number of objects which have changed clusters in the process.

**3.3 Procedure**

To explain how k-prototype work we have taken a small database of some companies located at different location. The attributes contain the company name, there revenue, Type of company and Location. There are basically 3 categorical attribute and 1 numeric attribute of this dataset. We have taken 7 objects which need to be clustered in 3 of these cluster.

As per the algorithm we randomly choose 3 clusters to be cluster or the prototype.

After this the first thing to do is to allocate the objects from the dataset to one of these cluster based on their similarity.

For this we find the distance of object to all these 3 clusters and allocate the object to the cluster having the least distance. Here the distance is calculated using two measures:

For calculating distance of numeric attribute following formulae is used:

D(Xi , Qi)= ∑ (xij-qij)2

Here D(Xi , Qi) is the function having arguments

Xi =ith object of the dataset

Qi= ith cluster of the 3 cluster taken.

For calculating distance of categorical attribute following formulae is used:

If the value of the attribute of the two object are same the function will return 0 otherwise 1.

Function is ∂ (xij,qij) where xij is the jth attribute of the ith object and qij is the jth attribute of the ith cluster.

Using above formulae we have calculated the distance for each of the object like:

Object 1 to cluster 1: 0+0+0+0=0

Object 1 to cluster 2: 1+1+1+0=3

Object 1 to cluster 3: 1+36+1+0=38

So the object 1 will be assigned to the cluster 1.

Similarly

Object 2 to cluster 1: 1+1+1+0=3

Object 2 to cluster 2: 1+1+1+0=3

Object 2 to cluster 3: 1+100+1+1=103

So the object 2 will be assigned to the cluster 1.

After following these step we get object 3 in cluster 2, object 4 in cluster 2, object 5 in cluster 3, object 6 in cluster 3 and object 7 in cluster 3.

This is the initial allocation step. Now we need to calculate the all the 4 attributes of the cluster.

For numeric attribute we sum the total value of all the object attribute value and divide the sum from number of objects in the cluster.

For categorical attribute we check the frequency of all the attribute value and assign the most frequent attribute to the value of cluster. In this way clusters now have new attributes value.

We again assign different objects to different cluster just like the above method. This process works as long as position of cluster do not change.

**3.2 Algorithm**

For Initial allocation:

FOR i = 1 TO NObjects

Mdistance=Distance(X[i],oprototypes[1]) + gamma\*Sigma(X[i],cprototypes[1])

FOR j = 1 TO NClusters

distance=Distance(X[i],oprototypes[j])+gamma\*

Sigma(X[i],cprototypes[j])

IF (distance < Mdistance)

Mdistance=distance

cluster=j

ENDIF

ENDFOR

Clustership[i]=cluster

ClusterCount[cluster] + 1

FOR j=1 TO NumberOfNumericAttributes

SumCluster[cluster,j]+X[i,j] oprototypes[cluster,j]=SumCluster[cluster,j]/ClusterCount[cluster]

ENDFOR

FOR j=1 TO NCategoric

FrequencyInCluster[cluster,j,X[i,j]]+1 cprototypes[cluster,j]=HighestFreq(FrequencyInCluster,cluster,j)

ENDFOR

ENDFOR

**For re-allocation:**

moves=0

FOR i = 1 TO NObjects …

(To find the cluster whose prototype is the nearest to object i. ) …

IF (Clustership[i]<>cluster)

moves+1

oldcluster=Clustership[i]

ClusterCount[cluster] + 1

ClusterCount[oldcluster] - 1

FOR j=1 TO NumberOfNumericAttributes

SumInCluster[cluster,j] + X[i,j] SumInCluster[oldcluster,j] - X[i,j] oprototypes[cluster,j]=SumInCluster[cluster,j]/ClusterCount[cluster] oprototypes[oldcluster,j]= SumInCluster[oldcluster,j]/ClusterCount[oldcluster]

ENDFOR

FOR j=1 TO NumberOfCategoricAttributes

FrequencyInCluster[cluster,j,X[i,j]] + 1

FrequencyInCluster[oldcluster,j,X[i,j]] - 1

cprototypes[cluster,j]=HighestFreq(cluster,j) cprototypes[oldcluster,j]=HighestFreq(oldcluster,j)

ENDFOR

ENDIF

ENDFOR

# CHAPTER 4

# EXPERIMENTAL RESULT AND ANALYSIS

## 4.1 Result Obtained after applying k-prototype to a numeric and categorical data set

We have taken a small database of some companies located at different location. The attributes contain the company name, there revenue, Type of company and Location. There are basically 3 categorical attribute and 1 numeric attribute of this dataset. We have taken 100 objects which need to be clustered in 4 of these cluster.

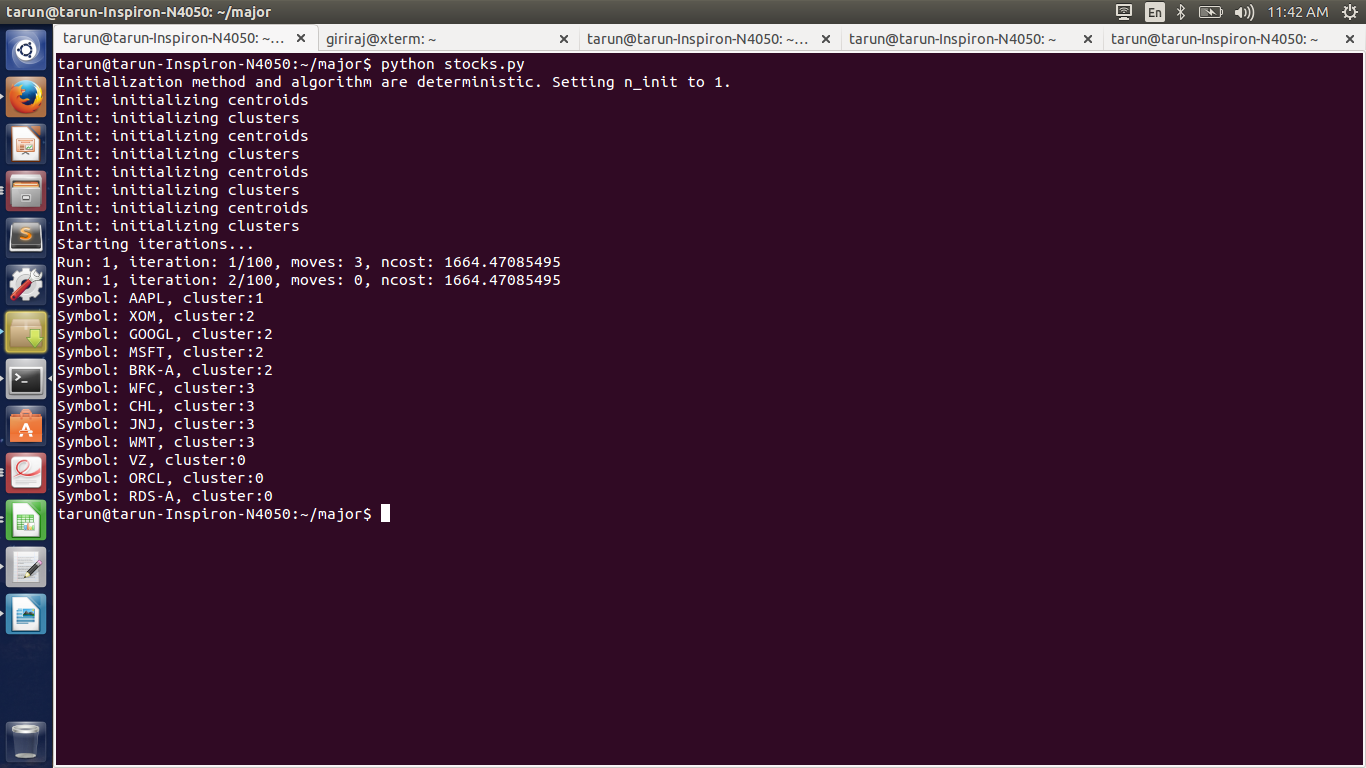
Here is the small portion of the database we have taken:

**Table 4.1: Dataset taken for analyzing k-prototype algorithm**

|  |  |  |  |
| --- | --- | --- | --- |
| Company | Market Price | Type of Company | Location |

|  |  |  |  |
| --- | --- | --- | --- |
| AAPL | 738.5 | Tech | USA |
| XOM | 369.5 | Nrg | USA |
| GOOGL | 368.2 | tech | USA |
| MSFT | 346.7 | Tech | USA |
| BRK-A | 343.5 | Fin | USA |
| WFC | 282.4 | Fin | USA |
| CHL | 282.2 | Tel | CN |
| JNJ | 279.7 | Cons | USA |
| WMT | 257.2 | Cons | USA |
| VZ | 205.2 | Tel | USA |
| ORCL | 192.1 | Tech | USA |
| RDS-A | 195.7 | Nrg | NL |

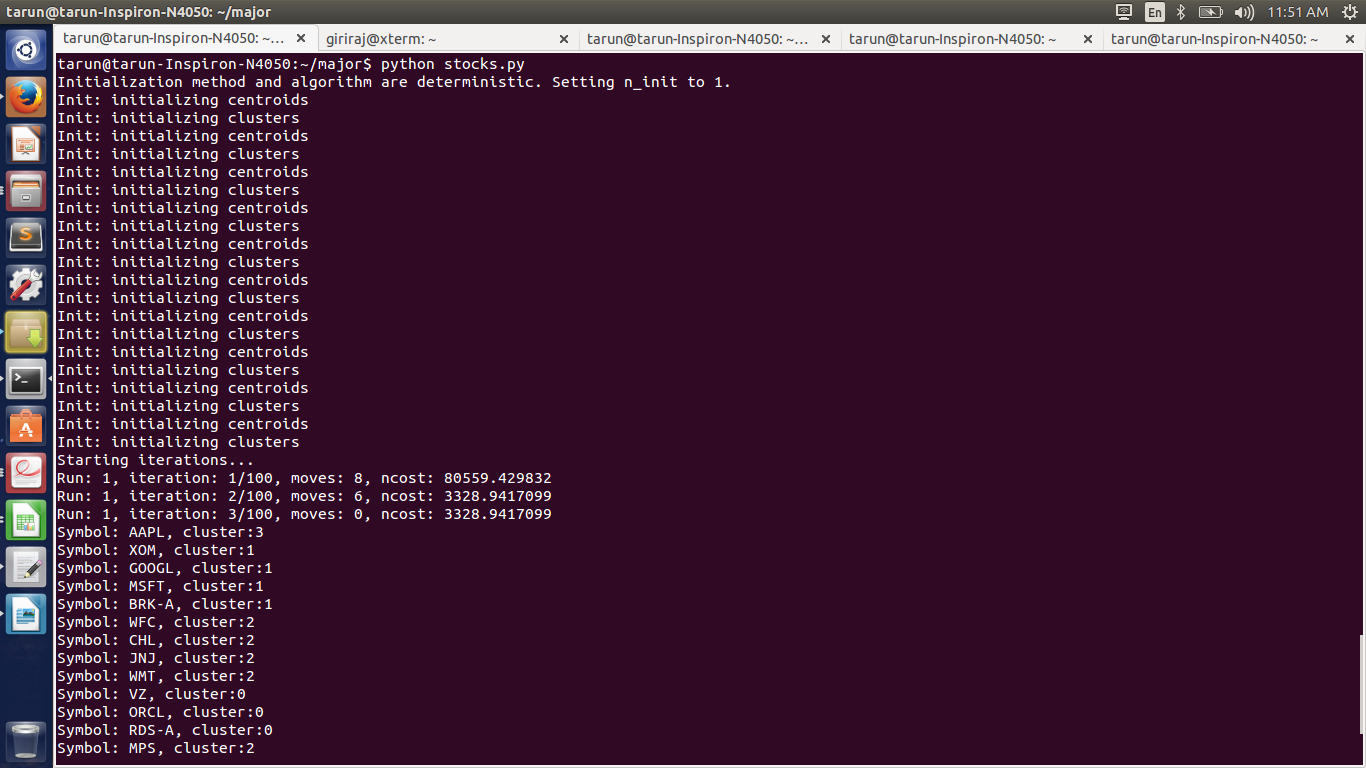
**4.2 Output on running k-prototype on the given dataset**



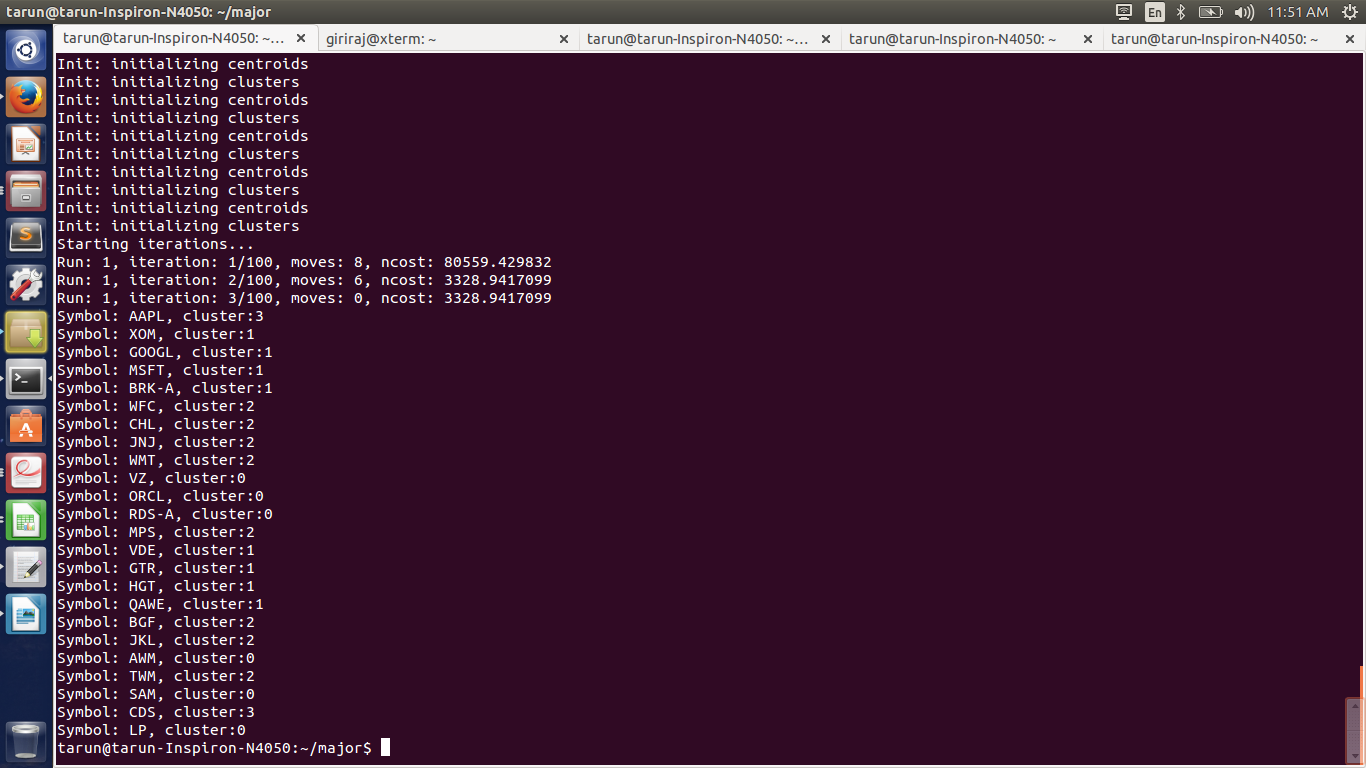
**Figure 4.2: Snapshot of the output of k-prototype for a dataset of 11 objects having 3 categorical and 1 numerical attribute**

Here we have clustered the objects into 4 cluster. The result is shown in these three snapshot.

In the figure 4.1 11 objects are clustered into 3 cluster whereas in figure 4.2 and figure 4.3 22 objects are clustered into 4 cluster.

****

**Figure 4.2: Snapshot of the output of k-prototype for a dataset of 22 objects having 3 categorical and 1 numerical attribute**

****

**Figure 4.2: Snapshot of the output of k-prototype for a dataset of 22 objects having 3 categorical and 1 numerical attribute**

**4.3 Analysis**

From the above result we have analyzed that clustering of points is dependent on both numeric and categorical values. The points, which are spatially close and have the same categorical values, are clustered together. If a point has a categorical value A but is close the majority of points having categorical value B and far away from the majority points having categorical value A, the point is clustered together with the points, which have categorical value B in majority. In the example we have taken, GOOGL and AAPL have two categorical values same so they should belong to same group but the distance between these two points is large so here XOM and GOOGL which has only one categorical attribute matching is taken into one cluster.

Other case in which categorical attribute is taken into higher consideration is object HGT and GTR. The distance between QAE and HGT is smaller as compared to HGT and GTR but none of their categorical attribute matches so HGT and GTR are placed in one cluster as per our example.

# CHAPTER 5

# CONCLUSION AND FUTURE WORK

## 5.1 Conclusion

Clustering has been widely applied to various domains to explore the hidden and useful patterns inside data. Because the most collected data in real world con tain both categorical and numeric attributes, the traditional clustering algorithm cannot handle this kind of data effectively.

The most attractive property of the k-means algorithm in data mining is its efficiency in clustering large data sets. However, that it only works on numeric data limits its use in many data mining applications because of the involvement of categorical data. The k-modes and k-prototypes algorithms have removed this limitation and extended the k-means paradigm into a generic data partitioning operation for data mining.

## 5.2 Future Work

We have presented the k-prototypes algorithm to cluster large real world data sets which uses data mining by finding the similarity measures between data points consisting of both numeric and categorical attribute. This algorithm preserves the efficiency of the k-means algorithm but removes its numeric data only limitation. We have demonstrated that it is efficient for clustering large data sets with mixed numeric and categorical values. Such data sets often occur in data mining applications.

K-prototype inherits the ideas of k-means, it applies Euclidean distance to numeric attributes and a distance function is defined to be added into the measure of the closeness between two objects. Object pairs with different categorical values will enlarge the distance between them. The main shortcomings of k-prototype may fall into followings:

1. Binary distance is employed for categorical value. If object pairs with the same categorical value, the distance between them is zero; otherwise it will be one. However, it will not properly show the real situation, since categorical values may have some degree of difference. For example, the difference between “high” and “low” shall not equal to the one between “high” and “medium”.

2. Only one attribute value is chosen to represent whole attribute in cluster center. Therefore, the categorical value with less appearance seldom gets the chance to be shown in cluster center, though these items may play an important role during clustering process. Additionally, since k-prototype inherits the ideas of k-means, it will retain the same weakness of k-means.

The major problem of existing clustering algorithms is that most of them treat every attribute as a single entity, and ignore the relationships among them. However, there may be some relationships among attributes. For example, the person with high incomes may always live in a costly residence, drive luxurious cars.

So we need to have a technique to convert categorical values into numerical values by not assigning them binary values but by finding an approach to convert categorical values into numerical values based on co-occurrence theory which says that if two items always show up in one object together, there will be a strong similarity between them. When a pair of categorical items has a higher similarity, they shall be assigned closer numeric values.