Combine Two Clustering Technique to Improve Result

Mr. Rahul Malik
M.Tech. Scholar
Deptt. Of CSE, JIET, Jind

Mr. Raj Kumar
Assistant Professor
Department of CSE, JIET, Jind

ABSTRACT

Cluster analysis or clustering is the assignment of a set of observations into subsets (called *clusters*) so that observations in the same cluster are similar in some sense. This research work deals with two of the most delegated clustering algorithms In k-means clustering, we are given a set of n data points in d-dimensional space Rd and an integer k and the problem is to determine a set of k points in Rd, called centers, so as to minimize the mean squared distance from each data point to its nearest centers. A popular heuristic for k-means clustering is Lloyd's algorithm. In this paper, we present a simple and efficient implementation of Lloyd's k-means clustering algorithm and KLC clustering algorithm, which we call the filtering algorithm. Distribution of data and manipulation allows for solving larger problems and executing applications that are distributed in nature. In this paper we present a grid-based distributed Support Vector Machine (SVM) algorithm. The Grid is a distributed computing infrastructure that enables coordinated resource sharing within dynamic organizations consisting of individuals, in situations and resources.

Keywords

Data Mining, K-mean, Lloyd's Clustering, Distributed Computing, SVM-Algorithm.

1. INTRODUCTION

With the enormous amount of data stored in files, databases, and other repositories, it is increasingly important, if not necessary, to develop powerful means for analysis and perhaps interpretation of such data and for the extraction of interesting knowledge that could help in decision-making. Data Mining, also popularly known as Knowledge Discovery in Databases (KDD), refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases.

The Knowledge Discovery in Databases process comprises of a few steps leading from raw data collections to some form of new knowledge. Clustering problems arise in many different applications, such as data mining and knowledge discovery, data compression and vector quantization, and pattern recognition and pattern classification. The notion of what constitutes a good cluster depends on the application and there are many methods for finding clusters subject to various criteria, both ad hoc and systematic. Among clustering formulations that are based on minimizing a formal objective function, perhaps the most widely used and studied is k-means clustering. Given a set of n data points in real d-dimensional space, Rd, and an integer k, the problem is to determine a set of k points in Rd, called centers, so as to minimize the mean squared distance from each data point to its nearest center. This measure is often called the squared-error distortion and this type of clustering falls into the general category of variance based clustering. Clustering based on k-means is closely related to a number of other clustering and location problems. These include the Euclidean k-medians (or the multisource Weber problem) in which the objective is to minimize the sum of distances to the nearest center and the geometric k-center problem in which the objective is to minimize the maximum distance from every point to its closest center. One of the most popular heuristics for solving the k-means problem is based on a simple iterative scheme for finding a locally minimal solution. This algorithm is often called the k-means algorithm. There are a number of variants to this algorithm, so, to clarify which version we are using, we will refer to it as Lloyd's algorithm. Lloyd's algorithm is based on the simple observation that the optimal placement of a center is at the centroid of the associated cluster (see. Given any set of k centers Z, for each center z 2 Z, let V .z. denote its neighborhood, that is, the set of data points for which z is the nearest neighbor.

2. K- MEAN ALGORITHM

Clustering is the process of partitioning or grouping a given set of patterns into disjoint *clusters*. This is done such that patterns in the same cluster are alike and patterns belonging to two different clusters are different. Clustering has been a widely studied problem in a variety of application domains including neural networks, AI, and statistics. The k-means method has been shown to be effective in producing good clustering results for many practical applications. However, a direct algorithm of k-means method requires time proportional to the product of number of patterns and number of clusters per iteration. This is computationally very expensive especially for large datasets. We propose a novel algorithm for implementing the k means method. Our algorithm produces the same or com.

The simplest and most popular among iterative and hill climbing clustering algorithms is the K-means algorithm (KMA). As mentioned above, this algorithm may converge to a suboptimal partition. Since stochastic optimization approaches are good at avoiding convergence to a locally optimal solution, these approaches could be used to find a globally optimal solution. The stochastic approaches used in clustering include those based on simulated annealing, genetic algorithms, evolution strategies and evolutionary programming [2]–[6].

Typically, these stochastic approaches take a large amount of time to converge to a globally optimal partition. In this paper, we propose an algorithm based on GA, prove that it converges to the global optimum with probability one and compare its performance with that of some of these algorithms.

Genetic algorithms (GA's) work on a coding of the parameter set over which the search has to be performed, rather than the parameters themselves. These encoded parameters are called solutions or chromosomes and the objective function value at a solution is the objective function value at the corresponding parameters. GA's solve optimization problems using a population of a fixed number, called the population size, of solutions.During each generation, they produce a new population from the current population by applying genetic operators' viz., *natural selection, crossover, and mutation*. Each solution in The population is associated with a figure of merit (fitness value) depending on the value of the function to be optimized. The selection operator selects a solution from the current population for the next population with probability proportional to its fitness value. Crossover operates on two solution strings and results in another two stings. Typical crossover operator exchanges the segments of selected stings across a crossover point with a probability. The mutation operator toggles each position in a string with a probability, called the *mutation probability*. For a detail study on GA, readers are referred to [7]. Recently, it has been shown that the GA's that maintain the best discovered solution either before or after the selection operator asymptotically converge to the global optimum. There have been many attempts to use GA's for clustering [4], [5], [6]. Even though all these algorithms, because of mutation, may converge to the global optimum, they face the following problems in terms of computational efforts. In the algorithms where the representation of chromosome is such that it favours easy crossover, the fitness evaluation is very expensive as in [4]. In the algorithms where the fitness evaluation is simple, either the crossover operation is complicated or it needs to be repeatedly applied on chromosomes to get legal strings [5], [6]. In this sense, selection and crossover are complementary to each other in terms of computational complexity. GA's perform most efficiently when the representation of the search space under consideration has a natural structure that facilitates efficient coding of solutions. Also, genetic operators defined on these codes must produce valid solutions with respect to the problem. Thus, in order to efficiently use GA's in various applications, one has to specialize GA's to the problems under consideration by hybridizing them with the traditional gradient descent approaches. A hybrid G that retains, if possible, the best features of the existing algorithm, could be the best algorithm for the problem under consideration. Davis also made these observations in his handbook [8]. Since KMA is computationally attractive, apart from being simple, we chose this algorithm for hybridization. The resulting hybrid algorithm is called the *genetic K-means algorithm* (GKA). We use the K-means operator, one step of KMA, in GKA instead of the crossover operator used in conventional GA's. We also define a biased mutation operator specific to clustering; called distances based mutation, and use it in GKA. Thus, GKA combines the simplicity of the K-means algorithm and the robust nature of GA's. Using finite Markov chain theory, we derive conditions on the parameters of GKA for its convergence to a globally optimal partition. We conduct experiments to analyse the significance of the operators used in GKA and the performance of GKA on different data sets and varying sizes of search spaces. We show through simulations that even if many duplicates of KMA starting with different initial partitions are run, the best partition obtained is not necessarily a global optimum, whereas almost every run of GKA eventually converge to a globally optimal partition. We also compare the performance of GKA with that of some of the algorithms based on GA, evolution strategies and evolutionary programming, which possibly converge to a global optimum, and show that GKA is faster than them in the next section, the statement of the problem under consideration along with a brief description of KMA is given. The conditions on the parameters of GKA are derived which ensure its convergence to the global optimum The main disadvantage of the k-means algorithm is that the number of clusters, *K*, must be supplied as a parameter. In this paper we present a simple validity measure based on the intra-cluster and inter-cluster distance measures which allows the number of clusters to be determined automatically. The basic procedure involves producing all the segmented images for 2 clusters up to *Kmax* clusters, where *Kmax* represents an upper limit on the number of clusters. Then our validity measure is calculated to determine which is the best clustering by finding the minimum value for our measure. The validity measure is tested for synthetic images for which the number of clusters in known, and is also implemented for natural images.

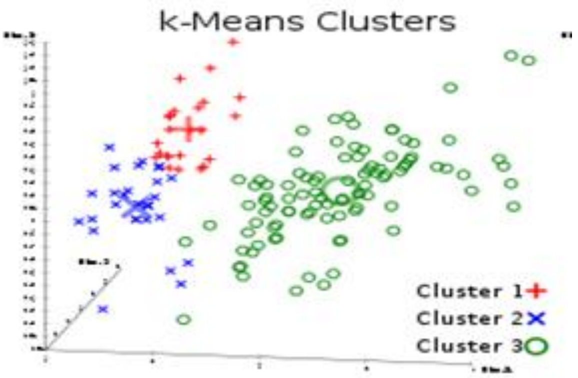


Figure 1: K-Mean clustering

.

3. SVM

Identification of distinct clusters of documents in text collections has traditionally been addressed by making the assumption that the data instances can only be represented by homogeneous and uniform features. Many real-world data, on the other hand, comprise of multiple types of heterogeneous interrelated components, such as web pages and hyperlinks, online scientific publications and authors and publication venues to name a few. In this paper, we present K-SVMeans, a clustering algorithm for multi-type interrelated datasets that integrates the well-known K-Means clustering with the highly popular Support Vector Machines. The experimental results on authorship analysis of two real world web-based datasets show that K-SVMeans can successfully discover topical clusters of documents and achieve better clustering solutions than homogeneous data clustering.

We present K-SVMeans that clusters datasets with heterogeneous similarity characteristics. K-SVMeans simultaneously clusters along one dimension of the data while learning a classifier in another dimension, which, in turn effects the intermediate cluster assignment decisions in the original dimension. K-SVMeans clustering is a hybrid clustering solution that merges the well-known K-Means clustering algorithm with Support Vector Machines (SVM), a highly popular supervised learning algorithm that has been shown to be highly effective, especially for text and graph area clustering.Although clustering is a decades old problem, research in multivariate data clustering where the data can be represented by multiple interrelated components has gained momentum only in the past couple of years due to its applicability in many domains. The initial directions towards multivariate clustering started with the simultaneous clustering of both rows and columns of contingency tables, also known as co-clustering, bi-clustering, or block clustering. [2] Proposed a spectral graph partitioning algorithm that clusters documents based on words, and words based on documents by finding the normalized cut of the bipartite graph. The same problem has been addressed in [3] by taking an information-theoretic approach. The proposed solution attempts to minimize the loss in mutual information between the original and the clustered contingency tables. In [7], a partial singular value decomposition of the edge-weight matrix of the bipartite graph is computed to co-cluster words and documents. Although these works have laid the foundations of multivariate data clustering, these algorithms cannot handle multi-type interrelated data objects. A multi-type extension of the bipartite spectral graph partitioning has been proposed in [5] for textual datasets and then for images and surrounding texts in web environment [4]. The data objects form a tripartite graph, and the tripartite graph is treated as two separate bipartite graphs. The spectral partitioning of the bipartite graphs is obtained by minimizing the cuts of both bipartite graphs using semi definite programming in$m$+n+t dimensional space where each dimension represents the dimension of a separate data type. The high-dimensionality of the problem space is prohibitive and prevents its applicability to real-world datasets of big sizes. A multi-way clustering framework is proposed in [1] that maximizes the mutual information between the clusters of multiple data types based on representation of the interaction between each pair of data types as a contingency table of co-occurrence counts. The generation of clusters is performed by a combination of agglomerative (bottom-up) and partitional (top-down) clustering of different data types. The decision as to which types will be clustered agglomeratively or partitional, and the order of their clustering is determined by a clustering schedule determined beforehand of the clustering process, and an optimal clustering schedule needs to be provided by the user.

4. PROPOSED WORK

A key feature of K-SVMeans is the integration of Support Vector Machines with K-Means clustering. In this section, we provide background on SVMs and Online Support Vector Learning, which is a key component in clustering decisions made by K-SVMeans. K-Means with Online SVM and provide details of K-SVMeans algorithm.

Support Vector Machines are well known for their generalization performance and ability to handle high dimensional data which is a common case in document classification problems. Considering the binary classification case, let ((x1, y1) · · · (xn, yn)) be the training dataset where xi are the feature vectors that represent the observations and yi ∈ (−1, +1) be the two labels that each observation can be assigned to. From these observations, SVM builds an optimum hyper plane – a linear discriminant in the kernel transformed higher dimensional feature space – that maximally separates the two classes by the widest margin by minimizing the following objective function

min (**w**,**b**,ξ**i**) **w** =∑ **w**T + C
N
i=1
ξi (1)

Where **w** is the norm of the hyper plane, b is the offset, y(xi) are the labels and ξi are the slack variables that permit the non-separable case by allowing misclassification of training instances.

K-SVMeans is a K-Means based clustering algorithm for heterogeneous datasets where clustering along one data type learns a classifier in another, and the classifiers effect the clustering decisions made by the clusterer. The original formulation of K-Means algorithm first initializes k clusters with data objects and then assigns each object di, 1 ≤ i ≤ N to a cluster ci, 1 ≤ i ≤ k where di's distance to the representative of its assigned cluster ci is minimum. Variants of K-Means algorithm differ in the initialization of clusters (e.g. random or maximum cluster distance initialization), the definition of similarity (e.g. Euclidean or Kullback-Leibler Divergence), or the definition of cluster representativeness (e.g. mean, median or weighted centroid vector). K-SVMeans is independent of any of those variations, but for brevity, we describe the algorithm for Spherical K-Means with random initialization that represents each cluster by its centroid vector We start with a brief overview of traditional K-Means. Given n data objects x1, x2 · · · xn , $\forall$**xi** ∈ **Rw** where **w** is the size of the feature space and each xi is normalized such that ||xi|| = 1, K-Means partitions the xi into k disjoint clusters π1, π2, · · · , πk, so that

k
i=1
πi = {x1, x2, · · · xn} where $\pi i \cap \pi j = \emptyset$, i _= j

where the centroid ci of each cluster πi is defined as ci = _xk∈πiThe goal of the clusterer is to maximize the similarity between the data objects and their assigned clusters, hence, the objective function becomes
max$Q$ = k= j=1
xTi
cj $\forall \pi i$ 1 ≤ i ≤ k (5)
K-Means optimizes the objective function iteratively by following two steps: A cluster assignment step, where each data object is assigned to a cluster with the closest centroid,followed by a cluster centroid update step. The algorithm terminates when the change in the objective function value between two successive iterations is below a given threshold. Upon the termination of the algorithm, each data object belongs to one of the k clusters. This partitioning, however, is done on a single dimension.
Consider that the instances in the set X = (x1, x2, · · · , xn), which we want to obtain a clustering solution, are related to another set U = (u1, u2, · · · , um) in some way. Each xi can be related to one or multiple uj 's in a $X \longrightarrow U$ mapping where objects in U denote a unique property of xi. The reverse map

$U \rightarrow X$ lets us represent each u as a mixture of the xi's that are connected to it. Let T denote the relationship matrix where Tij = 1if xi is related to uj , and zero otherwise.

During the clustering process, the intermediate cluster assignments in K-SVMeans are determined by two conditions. In the first condition, a data object xi is reassigned from a cluster πi to πj if xi is closer to πj 's centroid than πi's centroid and the u's of xi are classified into the positive class by πj 's SVM and into the negative class by πi's SVM.

200

K-SVMeans Cluster Assignment

Definitions:

xi:*Objects to be clustered*

dij : *distance of object xi to cluster* πj

m(*i*): *assigned cluster of xi*

l(πi): *SVM learner of cluster* πi

ˆ*y*(*u*, π) = _*n*

z=1 $\alpha\pi$

*z***K**π(*u*, *u*πz

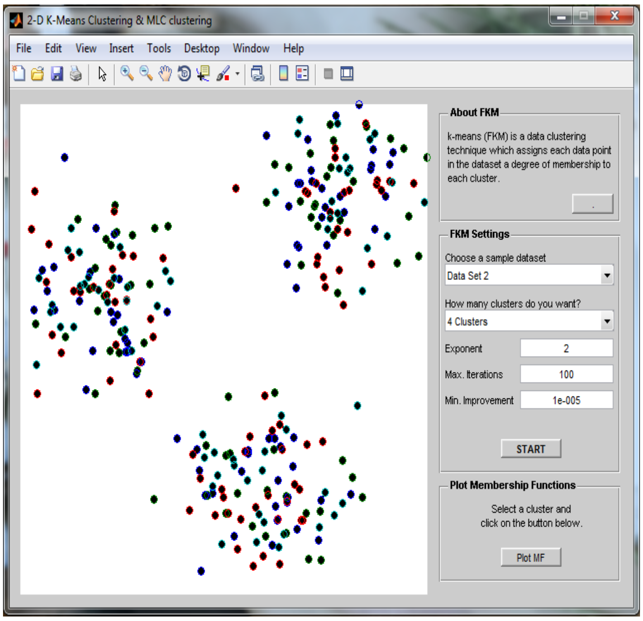
) + *b*π *SVM decision*

value **u** *for cluster* π

λ: Penalty term

5. RESULT AND ANALYSISE

Select the dataset on which we want to implement algorithm and how much cluster it can make as shown in figure 2


2-D K-Means Clustering & MLC clustering
File Edit View Insert Tools Desktop Window Help

About FKM

k-means (FKM) is a data clustering technique which assigns each data point in the dataset a degree of membership to each cluster.

FKM Settings

Choose a sample dataset: Data Set 2

How many clusters do you want? 4 Clusters

Exponent: 2

Max. Iterations: 100

Min. Improvement: 1e-005

START

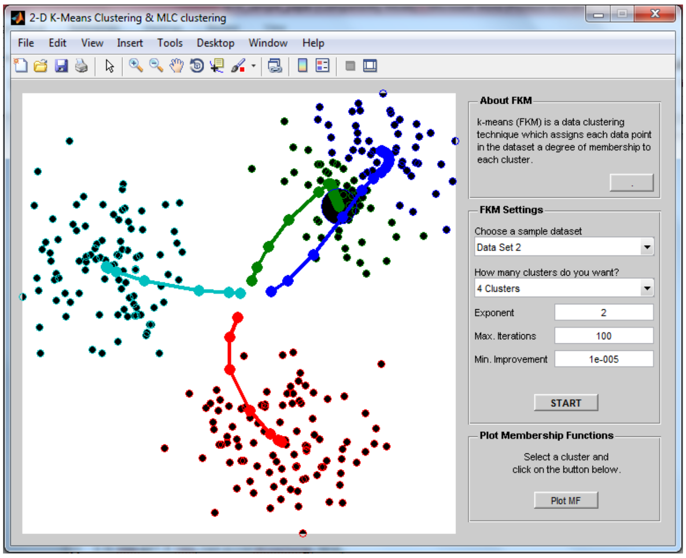
Plot Membership Functions

Select a cluster and click on the button below.

Plot MF

We get the 4 clusters which represent the similar data in different -2 groups as shown in figure 3. There are different colors for different columns in dataset. We use the LIC dataset to get result. The coloring is shown below:

- This colour shows the Policy Number in Dataset.
- This colour shows the Initial Name in Dataset.
- This colour shows the Premium Amount in Dataset.
- This colour shows the Age in Dataset.
- This colour shows the Agent Code in Dataset


2-D K-Means Clustering & MLC clustering
File Edit View Insert Tools Desktop Window Help

About FKM

k-means (FKM) is a data clustering technique which assigns each data point in the dataset a degree of membership to each cluster.

FKM Settings

Choose a sample dataset: Data Set 2

How many clusters do you want? 4 Clusters

Exponent: 2

Max. Iterations: 100

Min. Improvement: 1e-005

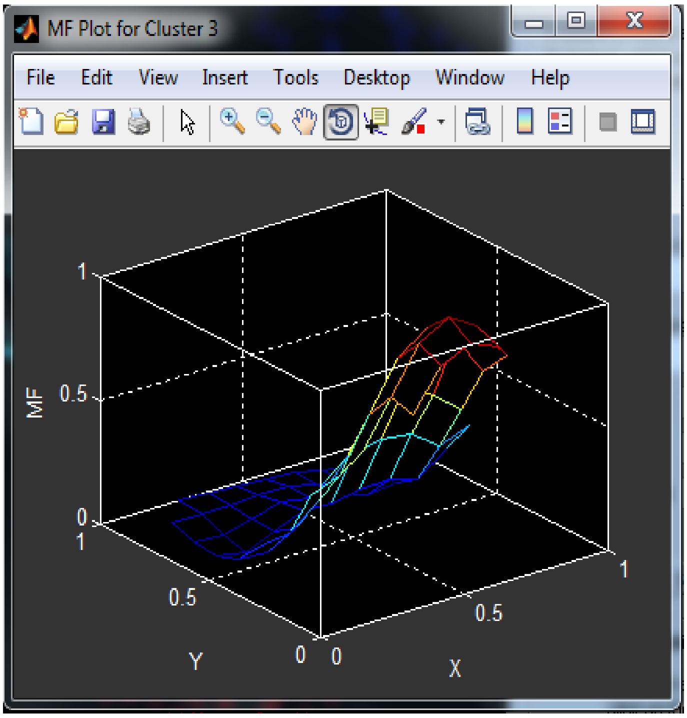
START

Plot Membership Functions

Select a cluster and click on the button below.

Plot MF

MF Plot for dataset 1 before implement algorithm as shown in figure 3



MF Plot for dataset 1 after implement algorithm as shown in figure 4 which define the value of MF as comparing with X and Y axis. The plot is 3- D. We can convert it in 2_D by which rotating tool. Which is easy to understand.

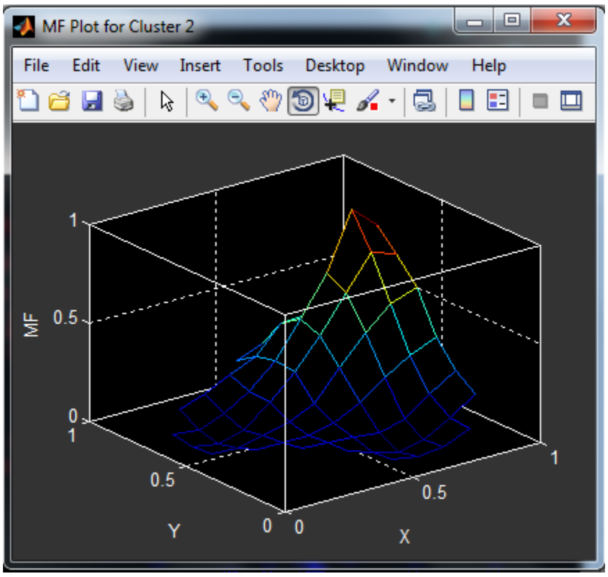


Figure 4: Membership Function for Cluster 2

6. REFERENCES

[1] R. Bekkerman, R. El-Yaniv, and A. McCallum.“Multi-way distributiona clustering via pairwise interactions.” In Proceedings of ICML’05, pages 41–48, 2010.

[2] Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. “A support vector clustering method”. In International Conference on Pattern Recognition, 2000.

[3] Y.W. Lim and S.U. Lee, “On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques”.Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. .

[4] Ahmed S, Coenen F, Leng PH (2006) “Tree-based partitioning of date for association rule mining”. Knowl Inf Syst 10(3):315–331

[5] Banerjee A, Merugu S, Dhillon I, Ghosh J (2005) “Clustering with Bregman divergences”. J Mach Learn Res 6:1705–1749.

[6] Bonchi F, Lucchese C (2006) “On condensed representations of constrained frequent patterns”. Knowl Inf Syst 9(2):180–201.

[7] Breiman L (1968) Probability theory. “Addison-Wesley, Reading. Republished (1991) in Classics of mathematics”. SIAM, Philadelphia.