IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 14, Issue 1 (Sep. - Oct. 2013), PP 46-53
www.iosrjournals.org

Survey on Unsupervised Learning in Datamining

S.Uma Parameswari, B.Jayanthi,
M.Phil Scholar, School Of Computer Studies, RVS College Of Arts and Science.
Asst Professor, School Of Computer Studies, RVS College Of Arts and Science.

Abstract: *Data mining is a collection of techniques which is used to extract knowledge from huge amount of data. The first section contains the introduction and the next part describes the main task of data mining. Third part explains the clustering methods like partitioning and hierarchical methods.*
Keywords*: Clustering, k-means, agglomerative and divisive.*

I. Introduction

Data mining, which is also referred to as Knowledge Discovery in Databases (KDD) is defined as a process of finding valid, novel, useful and understandable patterns in data [2, 3, 1]. Data mining is about finding insights which are statistically reliable, previously unknown and actionable from data (4). This data must be available, relevant, adequate, and clean. The data mining problem must be well-defined cannot be solved by query and reporting tools and guided by a data mining process model [5].

The two primary goals of data mining are prediction and description. Prediction is used to predict unknown or future values by using some variables or fields in the database. Description concentrates on finding human-interpretable patterns describing the data.

Data mining involves six common classes of tasks:[1]Anomaly detection: Identifying unusual data records that require further investigation. Association rule learning : finding relationship between variables. For example a supermarket might gather information on customer purchasing. Using association rule mining the supermarket can determine which product are frequently purchased together and utilizes this information for marketing purpose. It is also referred to as market basket analysis [8]. Clustering is the task of finding 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 generating known structure to new data. For example, an e-mail program might attempt to classify an e-mail as "spam". Regression tries to find a function which models the data with the least error. Summarization provides a more comfortable representation of the data set, which includes visualization and report generation [6]. Sequential pattern mining find set of data items that occur frequently together in several order. Sequential pattern mining extracts frequent subsequences from a sequence database.

Clustering and classification are both fundamental tasks in Data Mining. Classification is used mostly as a supervised learning method, clustering for unsupervised learning . The goal of clustering is descriptive, that of classification is predictive [7].

II. Partitioning Methods

Partitional methods obtain a single level partition of objects. These methods usually are based greedy heuristics that are used iteratively to obtain a local optimum solution. Given n objects these methods make $k < n$ clusters of data and use an iterative relocation method. It is assumed that cluster has at least one object and each object belongs to only one cluster. Objects may be relocated between clusters as the clusters are refined. The aim of partitional methods is to reduce the variance within each cluster as much as and have large variance between the clusters. Since the partitional methods do not normally explicitly control the inter-cluster variance, heuristics maybe, for ensuring large inter-cluster variance- One may therefore consider the aim to be minimum ratio like *a/b* where *a* is some measure of within cluster variance and *b* is some measure of be between cluster variation. We will discuss two methods,

1. K-means methods
2. Density based methods
3. Expectation Maximization methods

Both of these methods converge to a local minimum which minimum they converge to primarily on the starting points.

III. The K-Means Method

K-means is the simplest and most popular classical clustering method that is easy to implement. Classical method can only be used if all the objects is located in the main memory. This method is called K-means since each of the K clusters is represented by the mean of the objects within it. It is also called the

centroid method since at each step the point of each cluster is assumed to be known and each of the remaining points are allocated in cluster whose centroid is nearest to it. Once this allocation is completed, the centroids of the cluster are recomputed using simple means and the process of allocating points to each cluster is repeated until there is no change in the clusters (or some other stopping criterion).

The K-means method uses the Euclidean distance measure $D(x, y) = (\sum (xi - yi)^2)^{1/2}$

This appears to work well compact clusters. The K-means method may be described as follows:

1. Select the number of clusters be *k.*
2. Pick *k* seeds randomly unless the user has some insight into the data as centroids of the *k* clusters.
3. Compute the Euclidean distance of each object in the dataset from each of the centroids.
4. Allocate each object to the cluster it is nearest to based on the distances computed in the previous step.
5. Compute the centroids of the clusters by computing the means of die attribute values of the objects in each cluster".

STUDENT	AGE	MARK1	MARK2	MARK3
S1	18	73	75	57
S2	18	79	85	75
S3	23	70	70	52
S4	20	55	55	55
S5	22	85	86	87
S6	19	91	90	89
S7	20	70	65	60
S8	21	53	56	59
S9	19	82	82	60
S10	47	75	76	77

6. Check if the stopping criterion has been met, then go to Step 7. If not, go to Step 3.
7. One may decide to stop at this stage or to split a cluster or combine two clusters or some stopping criterion is met.

The time complexity is *O (n)* and is guaranteed to find a local minimum**.**

Example

Consider the data about students given in Table .The only attributes are the age and the three Marks.

Step 1and 2: Let the three seeds be the first three students as shown in Table

STUDENT	AGE	MARK1	MARK2	MARK3
S1	18	73	75	57
S2	18	79	85	75
S3	23	70	70	52

Step 3 and 4: Now compute the distances using the four attributes and using the sum of absolute differences for simplicity (i.e. using the K-median method). The distance values for all the objects are given in Table wherein columns 6, 7 and 8 give the three distances from the three seeds respectively. Based on these distances, each student is allocated to the nearest cluster.

First iteration- allocate each object to the nearest cluster

					Distances from Clusters			
C1	18.0	73.0	75.0	57.0	From C1	From C2	From C3	Allocation to the nearest cluster
C2	18.0	79.0	85.0	75.0				
C3	23.0	70.0	70.0	52.0				
S1	18.0	73.0	75.0	57.0	0.0	34.0	18.0	C1
S2	18.0	79.0	85.0	75.0	34.0	0.0	52.0	C2
S3	23.0	70.0	70.0	52.0	18.0	52.0	0.0	C3
S4	20.0	55.0	55.0	55.0	42.0	76.0	36.0	C3
S5	22.0	85.0	86.0	87.0	57.0	23.0	67.0	C2
S6	19.0	91.0	90.0	89.0	66.0	32.0	82.0	C2
S7	20.0	70.0	65.0	60.0	18.0	46.0	16.0	C3
S8	21.0	53.0	56.0	59.0	44.0	74.0	40.0	C3
S9	19.0	82.0	82.0	60.0	20.0	22.0	36.0	C1
S10	47.0	75.0	76.0	77.0	52.0	44.0	60.0	C2

The first iteration leads to two students in the first cluster and four each in the second and third clusters.
Steps 3 and 4: Use the new cluster means to recompute the distance of each object to each of to means, again allocating each object to the nearest cluster. The following table shows the second iteration result.

Step 5: The following table compares the cluster means of clusters found in the above table with the original seeds.

Table Comparing new centroids and the seeds

	AGE	MARK1	MARK2	MARK3
C1	18.5	77.5	78.5	58.5
C2	26.5	82.5	84.3	82.0
C3	21	61.5	61.5	56.5
Seed1	18	73	75	57
Seed2	18	79	85	75
Seed3	23	70	70	52

Second iteration- allocate each object to the nearest cluster

					Distances from Clusters			
C1	18.5	77.5	78.5	58.5	From C1	From C2	From C3	Allocation to the nearest cluster
C2	26.5	82.5	84.3	82.0				
C3	21	61.5	61.5	56.5				
S1	18.0	73.0	75.0	57.0	10.0	52.3	28.0	C1
S2	18.0	79.0	85.0	75.0	25.0	19.8	62.0	C2
S3	23.0	70.0	70.0	52.0	27.0	60.3	23.0	C3
S4	20.0	55.0	55.0	55.0	51.0	90.3	16.0	C3
S5	22.0	85.0	86.0	87.0	47.0	13.8	79.0	C2
S6	19.0	91.0	90.0	89.0	56.0	28.8	92.0	C2
S7	20.0	70.0	65.0	60.0	24.0	60.3	16.0	C3
S8	21.0	53.0	56.0	59.0	50.0	86.3	17.0	C3
S9	19.0	82.0	82.0	60.0	10.0	32.3	46.0	C1
S10	47.0	75.0	76.0	77.0	52.0	41.3	74.0	C2

The number of students in cluster 1 and in again 2 and the other two clusters still has four students each. A more careful look shows that the clusters have not changed at all. Therefore the method has converged rather quickly for this very simple dataset.

Cluster 1 – S1, S9
Cluster 2 – S2, S5, S6, S10
Cluster 3 – S3, S4, S7, S8

IV. Density Based Method

Density-based methods are based on the assumption that clusters are high density collections of arbitrary shape that are separated by a large space of low density data (which is assumed to noise). Therefore the basis for density-based methods is that for each data point in a cluster, at least a minimum number of points must exist within a given distance. Data that is not within such high density cluster is regarded as outliers or noise. Another way of looking at density-based clustering is that the clusters are dense regions of probability density in the data sets.

Objects are declared to be outliers if there are few other objects in their neighbourhood. The size parameter *R* determines the size of the clusters found. If *R* is big enough, there would be one big cluster and no outliers. If *R* is small, there will file small dense clusters and there might be many outliers. The following concepts are required in the DBSCAN method:

a.Neighbourhood: The neighbourhood of an object y is defined as all the objects that are within the radius *R* from y,

b.Core object: An object y is called a core object if there are *N* objects within its neighbourhood.

c.Proximity: Two objects are defined to be in proximity to each other if they belong to the same cluster. Object *z1* is in proximity to object z_2 if two conditions are satisfied:

(1) The objects are close enough to each other, i.e. within a distance of *R*.

(2) *Z2* is a core object as defined above.

d.*Connectivity:* Two objects z1 and z2 are connected if there is a path or chain of z*1,z2,…zn.from z1 to zn su*ch that each zi+1 is in proximity to object z*i.*

We now outline the basic algorithm for density-based clustering:

1. Select values of *R* and *N.*
2. Arbitrarily select an object *p.*

3. Retrieve all objects that are connected to given *R* and *N.*
4. If *p* is a core object, a cluster is formed.
5. If *p* is a border object, no objects are in its proximity. Choose another object. Go to step3.
6. Continue the process until all of the objects have been processed.

V. Expectation Maximization (EM)

Expectation Maximization method is based on the assumption that the objects *in the dataset have attributes* whose values are distributed according to some (unknown) linear combination (or mixture) of simple probability distributions. While the K- MEANS method involves assigning objects to clusters to minimize within-group variation, the EM method assigns objects to different clusters with certain probabilities in an attempt to maximize expectation (or likelihood) of assignment.

The EM method consists of a two-step iterative algorithm. The first step, called the estimation step or E-step, involves estimating the probability distributions of the clusters given in the data. The second step, called the maximization step or the M-step, involves finding the model parameters that maximize the likelihood of the solution.

The EM method assumes that all attributes are independent random variables. In a simple case of just two clusters with objects having only a single attribute, we may assume that the values vary according to a normal distribution. The EM method requires that we now estimating the following parameters:

1. The mean and standard deviation of the normal distribution for cluster 1.
2. The mean and standard deviation of the normal distribution for cluster 2.
3. The probability *p* of a sample belonging to duster 1 and therefore probability belonging to cluster 2.

The EM method then works as follows:

1. Guess the initial values of the five parameters (the two means, the two standard deviation and the probability *p)* given above.
2. Use the two normal distributions (given the two guesses of means and two guess! standard deviations) and compute the probability of each object belonging to each (J two clusters.
3. Compute the likelihood of data coming from these two clusters by multiplying the sum of the probabilities of each object.
4. Re-estimate the five parameters and go to Step 2 until a stopping criterion has been met.

The EM method is based on a statistical model that, when appropriate, shows optimal results. The method assumes that all attributes are independent and normally distributed.

Hierarchical Methods

Hierarchical methods produce a nested series of clusters. The hierarchical methods attempt to capture structure of the data by constructing a tree of clusters. Two types of hierarchical approaches are possible.

(1)*Agglomerative* approach for merging groups (or bottom-up approach), each object at the start of cluster by itself and the nearby clusters are repeatedly merged resulting in larger and larger cluster until some stopping criterion is met.

(2)Divisive approach (or the top-down approach), all the objects are put in a single cluster to start until repeatedly performs splitting of clusters resulting in smaller and smaller clusters.

VI. Agglomerative Method

The basic idea of the agglomerative method is to start out with *n* clusters for *n* data points, which each cluster consisting of a single data point. Using a measure of distance, at each step of the method, the method merges two nearest clusters, thus reducing the number *of* clusters and building successfully larger clusters. The process continues until the required number of clusters has been obtained or all the data points are in one cluster. The agglomerative method leads to hierarchical clusters in which at each step we build larger and larger clusters that include increasingly dissimilar objects.

The agglomerative method is basically a bottom-up approach which involves the following steps.

1 Allocate each point to a cluster of its own. Thus *n* clusters for *n* objects.
2. Create a distance matrix by computing distances between all pairs of clusters either by using single-link metric or the complete-link metric. Some other metric may also be used to sort these distances in ascending order.
3. Find the two clusters that have the smallest distance between them.
4. Remove the pair of objects and merge them.
5. If there is only one cluster left then stop.
6. Compute all distances from the new cluster and update the distance matrix after the merger and go to Step 3.

Example
Use agglomerative clustering for clustering the data by using the centroid method for computing the distances between clusters.

Student	Age	***Mark!***	***Mark2***	***Mark3***
S1	18	73	75	57
S2	18	79	85	75
S3	23	70	70	52
S4	20	55	55	55
S5	22	85	86	87
S6	19	91	90	89
S7	20	70	65	60
S8	21	53	56	59
S9	19	82	82	60
S10	47	75	76	77

Steps 1 and 2: Allocate each point to a cluster and compute the distance matrix using the centroid method. The distance matrix is symmetric. The last column would be for S10 but the diagonal element is zero.

	S1	S2	*S3*	*S4*	S5	S6	*S7*	*S8*	S9
S1	0								
S2	34	0							
S3	18	52	0						
C1	41	75	38	0					
S5	57	23	67	95	0				
S6	66	32	82	106	15	0			
S7	18	46	16	30	65	76	0		
S8	44	74	40	8	91	104	28	0	
S9	20	22	36	60	37	46	30	115	0
S10	52	44	60	90	55	70	60	98	58

The above matrix gives the distance of each object with every other object.
Step 3 & 4: The smallest distance is 8 between object S4 and object S8. They are combined and removed and combined into cluster C1 in the place of S4.then the new
distance matrix is calculated. Except cluster C1, all the data remain unchanged.

Step 5 & 6:

	S1	S2	*S3*	*S4*	S5	S6	*S7*	*S8*	S9
S1	0								
S2	34	0							
S3	18	52	0						
S4	42	76	36	0					
S5	57	23	67	95	0				
S6	66	32	82	106	15	0			
S7	18	46	16	30	65	76	0		
S8	44	74	40	8	91	104	28	0	
S9	20	22	36	60	37	46	30	115	0
S10	52	44	60	90	55	70	60	98	58

The smallest distance now is 15 between objects S5 & S6. They are combined in a cluster and S5 and S6 are removed.

We find the shortest distance again. S3 and S7 are at a distance 16. We merge them and put C3.

	S1	S2	*S3*	*C1*	C2	*S7*	S9
S1	0						
S2	34	0					
S3	18	52	0				
C1	41	75	38	0			
C2	61.5	27.5	74.5	97.5	0		
S7	18	46	16	29	69.5	0	
S9	20	22	36	59	41.5	30	0
S10	52	44	60	88	62.5	60	58

In the same way all are merged together.

VII. Divisive Method

The divisive method is the opposite of the agglomerative method in that the method starts with Whole dataset as one cluster and then proceeds to recursively divide the cluster into two sub-clusters and continues until each cluster has only one object or some other stopping criterion has reached. There are two types of divisive methods:

1. *Monothetic:* It splits a cluster using only one attribute at a time. An attribute that has, most variation could be selected.
2. *Polythetic:* It splits a cluster using all of the attributes together.

A typical polythetic divisive method works like the following:

1. Decide on a method of measuring the distance between two objects. Also decide a threshold distance.
2. Create a distance matrix by computing distances between all pairs of objects within the cluster. Sort these distances in ascending order.
3. Find the two objects that have the largest distance between them. They are the dissimilar objects.
4. If the distance between the two objects is smaller than the pre-specified threshold and there is no other cluster that needs to be divided then stop, otherwise continue.
5. Use the pair of objects as seeds of a K-means method to create two new clusters.
6. If there is only one object in each cluster then stop otherwise continue with -Step 2.

Example

Solve the below data by using divisive method.

Student	Age	***Mark!***	***Mark2***	***Mark 3***
S1	18	73	75	57
S2	18	79	85	75
S3	23	70	70	52
S4	20	55	55	55
S5	22	85	86	87
S6	19	91	90	89
S7	20	70	65	60
S8	21	53	56	59
S9	19	82	82	60
S10	47	75	76	77

Distance matrix for the above example

	S1	S2	*S3*	*S4*	S5	S6	*S7*	*S8*	S9
S1	0								
S2	34	0							
S3	18	52	0						
S4	42	76	36	0					
S5	57	23	67	95	0				
S6	66	32	82	106	15	0			
S7	18	46	16	30	65	76	0		
S8	44	74	40	8	91	104	28	0	
S9	20	22	36	60	37	46	30	115	0
S10	52	44	60	90	55	70	60	98	58

The largest distance is 115 between the objects .S_8 and *S9.* They become the seeds of two new clusters. K-means is used to split the group into two clusters. The two clusters may be derived based on distances in the distance matrix.

Distances from the seeds of the two clusters

	S1	*S2*	*S3*	*S4*	S5	S6	S7	*S8*	S9
S8	44	74	40	8	91	104	28	0	115
S9	20	22	36	60	37	46	30	115	0

Cluster C1 includes S4, S7, S8 and S10.
Cluster *C2* includes *S1, S2, S3, S5, S6 and S9.*

Since none of the stopping criteria have been met, we decide to split the larger cluster next and then repeat the process. We find the largest distance in C2 first. The distance matrix given in Table 4.16 may be built using distances given in the distance matrix.

Distance matrix for objects in cluster C2

	S1	*S2*	S3	S5	*S6*
S1	0				
S2	34	0			
S3	18	52	0		
S5	57	23	67	0	
S6	66	32	82	15	0
S9	20	22	36	37	46

The largest distance in C2 is 82 between S3 and S6. C2 can therefore be split with S3 and S6 seeds. The distance matrix of cluster 1 is given in Table 4.17. The largest distance is 98 between S8 and S10. C1 can be split with S_8 and S10 as seeds. The method continues like this until one of the stopping criteria is met.
Distance matrix for objects in cluster C1

	S4	S7	S8
S4	0		
S7	30	0	
S8	8	28	0
S10	90	60	98

It should be noted that the splitting criterion does not take into account the shape of the cluster and essentially assumes that each cluster is spherical. It has been suggested that the next cluster to be split should be selected based on scatter and centroid distance.

VIII. Conclusion

Compared to Agglomerative and Divisive method, K-Means is the best method to compute numerical and categorical data.

References

[1] Robert J. Hilderman and Howard J. Hamilton. *Knowledge Discovery and Measures of Interest*. Kluwer Academic Publishers, 1 edition, 2001.
[2] Gregory Piatetsky-Shapiro, Usama Fayyad, and Padhraic Symth. *From Data Mining to Knowledge Discovery in Databases*. AAAI/MIT Press, 1996.
[3] Philip S. Yu, Ming-Syan Chen, and Jiawei Han. *Data Mining: An Overview from Database Perspective*. Ieee Trans. on Knowledge and Data Engineering, 1994.
[4] Elkan, C. (2001). Magical Thinking in Data Mining: Lessons from CoIL Challenge 2000. *Proc. of SIGKDD01*, 426-431.
[5] Lavrac, N., Motoda, H., Fawcett, T., Holte, R., Langley, P. & Adriaans, P. (2004). Introduction: Lessons Learned from Data Mining Applications and Collaborative Problem Solving. *Machine Learning* **57**(1-2): 13-34.
[6] http://en.wikipedia.org/wiki/Data_mining
[7] Veyssieres, M.P. and Plant, R.E. Identification of vegetation state and transition domains in California's hardwood rangelands. University of California, 1998
[8] Introduction to data mining by G.K.Gupta.