Ca' Foscari University

Department of Computer Science

MASTER THESIS

On Determining the Number of Dominant-Set Clusters

Venice, October, 2014

By:

Tinsae Gebrechristos Dulecha

Supervisor:

Prof. Marcello Pelillo

Acknowledgments

My deepest gratitude and appreciation goes to my supervisor Professor Marcello Pelillo, who is too much rich of ideas and good personality, without his systematic supervision, positive approach and guidance, this work would not be possible and also thanks to the whole members of informatics department of Ca'Foscari, for providing me a simple and comfortable environment that helps me curve my future career path to the field of AI.

I also want to thank Farshad Nourbakhsh and my best friend Eyasu Zemene who made a great contribution for the successful completion of this paper. Likewise, thanks to my dearest friends Bertukan Yohannes, Surafel Melaku and Achamyeleh Dangnew who were there for me whenever I need their help. I would also like to extend my gratitude to Dekama Safaye, Tizazu Eticha and Aynalem Tsegaye for covering my living expense of the first seventh month of my study.

At last but not least, I would like to thank my family : my father, Gebrechristos Dulecha, my lovely mother Etagegn Bishaw, my brothers and sisters ,who are always happy for me for the good things happening to me more than i am happy for myself and always caring for me, supporting and having my back.

Thank you GOD for being with me in my life!!!

ABSTRACT

Cluster analysis (Clustering) is the process of finding group of objects where, objects in the same group will be similar (related) to one another and dissimilar from objects in other groups. The fundamental and major problem in cluster analysis is how many clusters are appropriate for the description of a given system, which is a basic input for many clustering algorithm. In this thesis we build a new method called "On Determining the Number of Dominant-Set Clusters" for automatically estimating the number of clusters in unlabeled data sets, based on the Motzkin-Straus theorem. Motzin-strauss were able to show a connection between clique number (ω(G)) and the global optimal value of a certain quadratic function over the standard simplex. Moreover, they have used the definition of stability number and have shown that this maximization is equal to stability number in unweighted scenario.

In our work, we have inspired by this theorem so we have extended to the weighted case to detect the number of maximal cliques (clusters). Finally we came to design a two step method to determine the number of clusters. In the first step, we use dissimilarity matrix as an input and by minimizing it with replicator, we are able to detect the minimum number of clusters based on our defined stability number. And then, we examine the existence of undetected cluster based on the idea of "Efficient-out-of-sample extension of dominant-set clusters" paper.

After determining the number of clusters(cluster representatives) in order to check whether our approach determine the right number of clusters or not we propagate the class labels using graph transduction ,a popular semi-supervised learning algorithm, to unlabeled instances and we evaluate the accuracy of clusters formed.

In order to test the effectiveness of our approach, we have conducted an experiment on different toy, generated using different *matlab* functions, real, download from UCI machine learning page, datasets. We also tested our approach using some social network data sets to further extend our work. The experiments which has been performed on these datasets shows a good and promising results.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cluster analysis (clustering) is the process of finding groups of objects where, objects in the same group will be similar to one another and dissimilar from objects in other groups. As the main goal of cluster analysis is to assign objects in a dataset into meaningful classes, it can be applied in different area of specialization holding different functionalities. For example in sociology, it can be used to recognize communities within large groups of people, in image segmentation, it can be used to divide a digital image in to distinct regions for border detection or object recognition and other in areas like, image processing, computer vision, bioinformatics, signal processing and medical imaging.

Clustering is an unsupervised learning which learns by observation rather than using labels. Since there is no prior knowledge about the classes at the beginning of an investigation, it signifies the fact that the classification of the observed data into classes is only determined by the information provided by the given data. There are different types of clustering algorithms that differs in their notion of what their constituent is and the type of technique they use to find the right cluster in an efficient way. Some of widely used algorithms are; Partitional clustering algorithm, Hierarchical clustering algorithm, Spectral clustering algorithm, Density Based algorithm and Grid algorithm. Each of them has their own advantages and disadvantages.

A fundamental problem of applying most of these clustering approaches is that the number of clusters needs to be pre-specified before the clustering is conducted. However, the clustering results may heavily depend on the number of clusters specified. Thus, it is necessary to provide educated guidance for determining the number of clusters in order to achieve appropriate clustering results. At the current stage of research, there are different methods of determining the number of clusters even if, none of them is completely satisfactory. The gap method, which is proposed by Tibshirani, et al.[TWH00], is one of those methods used to compare the within-cluster dispersions in the observed data to the expected within-cluster dispersions assuming that the data came from an appropriate null reference distribution. Even though, the simulation results reported by Tibshirani, et al. indicated that the gap method is a potentially powerful approach in estimating the number of clusters for a dataset, recent studies have shown that there are situations where the gap method may perform poorly. For instance, when the data contain clusters which consist of objects from well separated exponential populations it doesn't work properly. Another well-known approach for determining number of cluster is the method proposed by Duda and Hart [DH73]. In their method [DH73], the null hypothesis that the m^{th} cluster is homogeneous is tested against the alternative that it should be subdivided into two clusters. Motivated by the above statement, we proposed a new approach which is based on the concept of Motzkin-Strauss theorem, Massimiliano Pavan and Marcello Pelillo [PP04].

Our proposed method is a two-step process where, in the first step we used dissimilarity matrix as an input and by minimizing it with replicator we were able to detect the minimum number of clusters based on our defined stability number. In the second step, we examined the existence of undetected cluster based on the idea of "efficient-out-of-sample extension of dominant-set clusters". Finally, to visualize whether our approach determines the number of clusters correctly or not, we used graph transduction which is one of the well known semi-supervised learning algorithms. We recovered the whole cluster structure and evaluate the accuracy of the recovered clusters.

The rest of the thesis is organized as follows: the next chapter explains basic concepts of graph theory, cluster analysis, graph transduction. Chapter three is about the very notion of a cluster based on dominant set, how we can identify dominant set using replicator dynamics and how we can predict cluster membership of out sample instances (unseen instances). Chapter four covers related works done by different scholars on determining number of clusters. In chapter five our proposed approach will be presented with experimental results using toy, real and social network datasets. Finally, we will discuss about some applications, future works and conclude our work.

Chapter 2

Background Concept

2.1 Basic Graph Theory Definition and Notations

A graph G is a set of pair (V, E); where V is a set of vertices(nodes) and E is a set of links called edges which connect the vertices. The vertices in the graph represent the objects(data points) where as, the edges represent the relationship (dis(similarity)) between the objects. The values assigned to edges is referred as edge weight. Based on the direction and weights of edges, there are different category of graphs. The graph is undirected if all the edges are bidirectional and directed if the edges points only in one direction. In other words; the graph is directed if the edges have a direction and undirected if the edges have no direction. The graph is called unweighted if the edge weight value is represented in terms of either 1 or 0 (there exist edge or not) and weighted otherwise. The number of vertices is the order of the graph, whereas the number of edges is its size. Graph $G = (V, E)$ is complete if all its vertices are pairwise adjacent, i.e. $\forall i, j \in V, (i, j) \in E$. A clique C is a subset of V such that $G(C)$ is complete. The maximum clique problem ask for a clique of maximum cardinality or maximum weight in case of weighted graph.

Complement Graph($\bar{G}$): The complement graph of $G = (V, E)$ is the graph $\bar{G} = (V, \bar{E})$, where $\bar{E}$= (i, j)|i, j ∈ V, i ≠j and (i, j)∉E.

Complete subgraph: a complete subgraph is a subgraph in which all pairs of nodes are connected by an edge. It will be maximal complete subgraph when it is not contained

in any other complete subgraph.

Clique: in a graph $G = (V, E)$ clique is a subset of the vertex set $C \subseteq V$, such that every two vertices in C is adjacent to each other ;i.e. there exists an edge connecting any two vertices of C. It also referred to as a maximal complete subgraph, where all vertices are connected.

Maximal clique: is a clique that can't be extended by including additional adjacent vertex. In other words, it's a clique which does not exist exclusively within the vertex set of a larger clique.

Maximum clique: is a clique of the largest possible size in a given graph.

Clique number: for a graph G clique number is the order of the largest clique in the graph and denoted by $\omega(G)$.

Independent set(Stable set): An independent set(stable set, vertex packing) is a subset of V, whose elements are pairwise non-adjacent. The size of a maximum independent set is called the stability number of graph and denoted by α(G). The maximum weight independent set problem asks for an independent set of maximum weight.

For example in figure 2.1, we have the following vertices(V):

V=A,B,C,D,E,F,G, Then:

- (A,B,D),(D,E,F),(E,F,G),(D,F,G),(D,E,F,G); are the possible cliques.
- Maximal cliques= (A,B,C),(D,E,F,G)
- Maximum cliques = (D,E,F,G); since it is the clique with the highest number of vertices.
- The set (D,E,F),(D,E,G), (E,F,G) and (D,F,G); are not the maximal clique because they are the subset of maximum clique (D,E,F,G), and
- the Cliqe number($\omega(G)$)=4

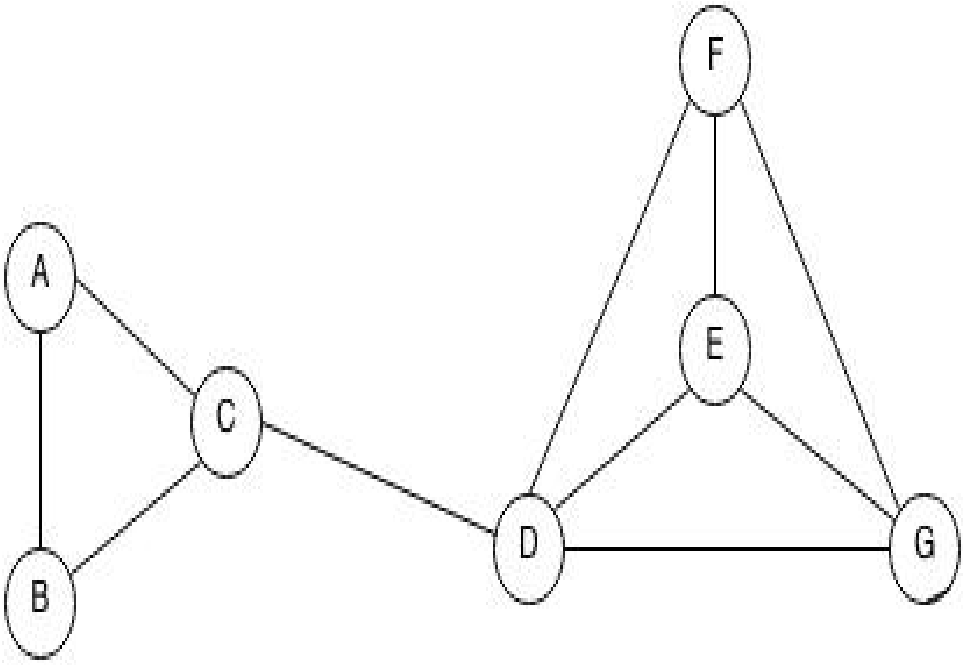

Fig. 2.1: *A graph with 7 1-vertex cliques (its vertices), 10 2-vertex cliques (its edges), 5 3-vertex cliques (ABC,DEF,EFG,DEG,DFG) and 1 4-vertex cliques (DEFG).*

2.2 Cluster Analysis

Cluster Analysis or Clustering is the process of finding a group of objects where; Ojects in the same group will be similar(related) to one another and dissimilar from objects in a different group. Cluster analysis has been used in different application areas like; image processing, computer vision, bioinformatics, signal processing, medical imaging and etc. The goal of cluster analysis is to partition a given input(a set of n objects organized as nxn matrix) into different similar groups based on a given condition. In general there are two types of clustering problems based on the input data type; referred to as Central(feature based) and Pairwise Clustering.

2.2.1 Central Clustering:

In this variation of clustering the objects to be clustered represented in terms of feature vectors. This means, the input set to the clustering algorithm is the n-dimensional feature vector. The K-Means is the well known feature based clustering algorithm.

2.2.1.1 K-Means Algorithm

Let X=$\{x_i\}$, i = 1,...,n be the set of n d-dimensional points to be clustered into a set of K clusters, C=$\{c_k, k = 1, ..., K\}$. The algorithm finds a partition where the distance between the empirical mean of a cluster and the points in the cluster is minimized. In other words the algorithm searchs for a compact cluster. Let us assume μ_k is the mean of the k^{th} cluster, Then the squared error(euclidean distance) between μ_k and the points in cluster c_k is defined as:

$$J(c_k) = \sum_{x_i \in c_k} \| x_i \text{ - } \mu_k \|^2$$

The goal of K-Means is to minimize the sum of the squared error over all the K clusters,

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \| x_i \text{ - } \mu_k \|^2$$

K-Means starts with an initial partition with K number of clusters and assigns patterns to clusters to reduce the squared error. The main steps of the algorithm is given below[JD88]:

1. Select k points as initial centroids

2. Repeat

3. Form k clusters by assigning all points to the closest centroid.

4. Recompute the centroid of each cluster

5. Until the centroids don't change

2.2.2 Pairwise clustering:

In many real-world applications there are situation where the feature vectors representation of a data is not easy to obtain. However, it is often possible to obtain a measure of the (dis)similarity between objects. The classical example is when the objects to be clustered are represented in terms of a graph. Pairwise clustering can be utilized in such type of situations.

Unlike central clustering, the pairwise clustering approach accepts the similarity matrix as input. Based on this similarity matrix it will try to partition the data points according to a set of coherence criteria. Dominant-set and pairwise clustering, and Normalized-Cut can be mentioned as Pairwise clustering types.

2.2.2.1 Normalized Cut

Normalized Cut is a method in which we cut a graph into two components to estimate the cost of the cut as a small fraction of the total affinity within a group.

$$\text{NCut}(A, B) = \frac{Cut(A,B)}{assoc(A,V)} + \frac{Cut(A,B)}{assoc(B,V)}$$

The score of the cut is denoted by the above equation, where V is a weighted graph and decomposed into two components A and B. $Cut(A, B)$ is sum of weights of all edges in V that has one end in A and other in B. The $assoc(A, V)$ and $assoc(B, V)$ are sum of weights of all edges with one end at A and B respectively.

The NCut essentially searches for the minimum(min) value of the criterion in $NCut(A, B)$. The *min* value signifies cutting the graph between two components or regions which has less edge weights between them and high internal edge weights[SM97].

2.3 Graph transduction

Among the machine learning community, Graph Transduction is the main topic while speaking of semi-supervised learning. Graph Transduction is a method which operates by propagating class membership information from labeled nodes to unlabeled nodes. The propagation works using the similarity between the nodes on the environment where only labeled nodes and unlabeled nodes exist. Usually the output of a graph transduction algorithm is the class assignment computed for the unlabeled nodes. when we see it from information theoretic point of view, the labeled nodes are the ones with zero entropy. This means; when initially their class is known the information they hold is with out any uncertainty, on the other hand the unlabeled nodes are the ones with maximum entropy because there is high rate of uncertainty to determine their class memebership.

Classical graph transduction algorithms initially assumes an unlabeled node's class might be any of the classes which exist in the current frame work with a uniform probability distribution. For example, if we are dealing with classification where there exists three classes, the initial prior probability for the unlabeled node to belong to one of the classes will be 1/3.

In a more formal way, assume there is a graph denoted by $G = (V, E)$, where V represent the total number of nodes(i.e the labeled and the unlabeled nodes together) and E represent the pairwise edges between nodes weighted by the similarity between the corresponding pairs of points. Then the data points are grouped as:

Labeled data : $\{(X_1, X1), ..., (X_L, Y_L)\}$ and

Unlabeled data : $\{X_{L+1,...,}X_n\}$

In many cases the number of labeled nodes(L) are less than the total existing nodes(n); i.e $L < n$ and if the edge between two nodes has high magnitude it means they have high degree of similarity and as a consequence they tend to be in the same class(or they belong to the same cluster). This concept is similar to the homophily analogy in social network analysis.

Finally the goal will be to propagate the information available at the few labeled nodes to the greater number of the unlabeled nodes in a consistent fashion.[EP12]

Chapter 3

Dominant sets

3.1 Dominant Set Clustering

Usually when we deal with the pairwise clustering processes, we represent the objects to be clustered as an undirected edge weighted graph where the n given points are represented by the vertices, and our similarities are the weights of the neighbour similarities (edges). This graph is then represented as an n by n similarity matrix where the value of the matrix are the weights that determines the corresponding similarity of the points of the corresponding column and row. That is if our similarity matrix is W, then the value of $w_{i,j}$ represent the similarity between the vertex i and the vertex j(which is the edge weight). Since there is no edge that connect a vertex to it self the main diagonal of the matrix is set to zero.

If we start from a very simple case, the binary case, our matrix becomes a (0,1) combination matrix that means an intermediate value is not allowed for the similarity(either they are similar or dissimilar). Here the graph is an undirected unweighed graph.The sort of structure in this graph that satisfy both the internal and external criteria is from a very classic notion of graph theory which is the notion of a `Maximal Clique` 2.1.

Before looking in detail the notion of dominant set let's see some definitions and ideas that leads them to the main definition of the notion of the dominant set.

Let G=(V,E,w) be an undirected weighted graph where V is the set of vertices, E is the set of edges and w is the edge weights which represent the similarity between pairs of linked points.

Definition: *let $S \subseteq V$ be a nonempty subset of vertices and ip$\in$S is a point in S,average weighted degree of a point ip with respect to S is defined as the sum of edge weights which connects ip to all points in S divided by the cardinality of S and denoted by $AWD_S(ip)$. Mathematically, the average weighted degree of a point ip with respect to a set of vertices S is expressed as*

$$AWD_S(ip) = \frac{1}{|S|} \sum_{p \in S} w_{ip,p} \tag{2.1.1}$$

Where $w_{ip,p}$ is the weight(the similarity) between the two points

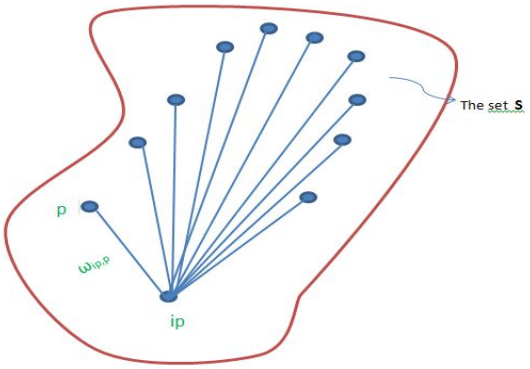

Fig. 3.1: *Average weighted degree of point ip*

This is the average weighted similarity between the point p and the other points in the set S.

The relative similarity,$\phi_S(ip, op)$, between two objects,ip and op(i and o to indicate the points inside and outside), with respect to the average similarity between node ip and its neighbours is described as the difference between the absolute similarity between ip and op ($w_{ip,op}$) and the average weighted similarity $AWD_S(ip)$

$$\phi_S(ip, op) = w_{ip,op} - AWD_S(ip) \tag{2.1.2}$$

This $\phi_S(ip, op)$ can be positive or negative based on the value of the absolute similarity and the average weighted similarity. If the absolute similarity is greater than the average weighted similarity, it is positive, otherwise it becomes negative.

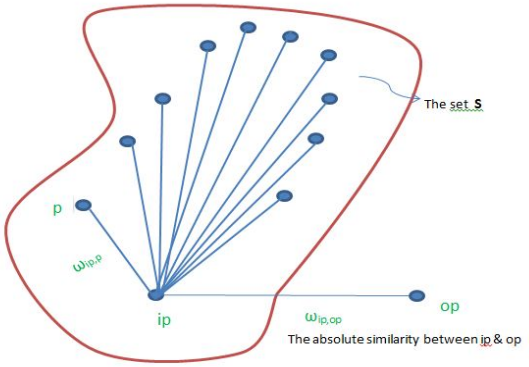

Fig. 3.2: *Relative similarity between two objects*

Using this result of $\phi_S(ip, op)$ it is possible to have the following recursive definition that allows us to assign a weight to a node. This is the main definition which allows us to give the main definition of dominant set. If the cardinality of the set S is 1 then by definition $W_S(ip) = 1$. Otherwise we have to sum up all the relative similarities between i and all other points in the set S, and this tells us how similar is point ip on average with respect to all other points in the set S except ip.

$$W_S(ip) = \sum_{p \in S \setminus \{ip\}} \phi_{S\setminus\{ip\}}(ip, p) W_{S\setminus\{ip\}}(p) \tag{2.1.3}$$

Then the weight of the set S is the sum of each weights $W_S(ip)$. We know $W_S(ip)$ the measure of how much tightly a vertex is coupled with other set of vertices in S. In other word it tells us whether we have to add or not a point ip to the set S.

Definition, Pavan and Pelillo [PP07]: *A non-empty subset of vertices $S \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq S$, is said to be dominant if:*

1. $W_S(i) > 0$ *for all* $i \in S$ *(2.1.4)*
2. $W_{S\cup\{i\}}(i) < 0$ *for all* $i \notin S$ (2.1.5)

These two conditions are exactly the same condition of the clustering criteria, so both criteria for a cluster are satisfied. Here we can say the notion of a cluster coincides with the notion of dominant set. If we know they coincide, how can we calculate dominant set? or how can we partition a set of data in to dominant set? Pelillo and Pavan, instead of using a standard algorithm to find dominant set,they transform the purely combinatorial problem of finding a dominant set in a graph in to a pure quadratic optimization problem and to solve the problem they used evolutionary game theory dynamical system.Using this algorithm it is possible to select out the identified dominant set from the graph and continue until the stopping criterion which checks if we have an empty set of vertices.

3.2 Identifying Dominant Sets With Replicator Dynamics

Pavan and Pelillo [PP07] showed the relationship between the notion of a cluster and dominant sets. In their paper they characterized the notion of dominant set in terms of continuous optimization problem. They stated that the notion of a cluster and its relationship to dominant sets were mathematically equivalent by formulating the optimization problem as a standard quadratic programme where

$$f(x) = X^T AX \tag{2.2.1}$$

is maximized subject to the constraint that $\mathbf{X}$ lies on the standard simplex$\triangle = \{x \in \mathbb{R}^{\kappa} X \geq 0$ and $\sum_n x_n = 1\}$. $\mathbf{A}$ in this case is defined as the similarity matrix of the graph and x is defined as the weighted characteristic vector and it is defined in terms of

the subset of vertices **S**,

$$x^S = \begin{cases} \frac{w_S(i)}{W(S)} & if\ i \in S; \\ 0 & otherwise. \end{cases} \tag{2.2.2}$$

where $W(S) = \sum_{i \in s} w_s(i)$ is the total weight, which must be greater than 0. Pavan and Pelillo [PP07] proved that by using this definition of **x**, the maximization of the objective function is the same as finding dominant sets. Further details of this proof can be found in [PP07]. To find a local solution of the objective function *f*, a method taken from evolutionary game theory, called replicator dynamics was used. The first-order replicator equations are defined as

$$x_i(t+1) = x_i(t)\frac{(AX(t))_i}{X(t)^T AX(t)} \tag{2.2.3}$$

And are applied to all nodes in the network in turn. Since **A** is symmetric, the replicator equations provide a strictly increasing update to the characteristic vector x, which converges upon the local solution of *f*. By taking the support or non-zero indices of the final **x**, we identify the elements of the graph that are a dominant set. That is, the solution of the replicator equations converges exactly on a characteristic vector that conforms exactly to conditions 2.1.4 and 2.1.5. In practice, **x** is initialized with uniform weights, which corresponds to the centroid of the standard simplex.

Since the replicator equations only converges on most dominant set of a particular graph, an effective way of identifying further clusters in the network is to apply a peeling strategy [PP07]. This involves finding a dominant set(cluster) using equation 2.2.3, removing the vertices in the cluster from the similarity graph, and then re-applying the replicator equations to the remaining vertices's. In practice, the elements of the characteristic vector rarely converged to exactly zero so a threshold was used to identify

numbers that were extremely close. The same threshold was also used to ensure that if the value of the maximized objective function was too small. We considered that all the remaining nodes in the graph were singletons.

3.3 Predicting Cluster Membership for Out-of-Sample Data

Dominant-set is efficient and novel graph-based clustering method, even if it has the problem of working efficiently in big data sets and in a dynamical situation; where the data set needs to be updated continually. Segmenting high resolution image and spatio-temporal data as well as applications like document classification and database visualization is the classical examples such type of problems.

To resolve this problem, pavan and pelillo proposed a new solution on their paper "Efficient Out-of-Sample Extension of Dominant-set Clusters" [PP04]. Given a dataset to be clustered, instead of doing the clustering for the whole dataset or repeating the whole clustering process when the new point is added; it is expensive in terms of cost. The authors basically propose to take some percent of the data set in case of static big data set or take the initial data set in case of dynamic situations, then performing the clustering process. Then for each new unseen instance(the instance not a member of the initial sample or new point to be added) they calculate the similarity between the unseen instance and the cluster detected previously and assign the point to the cluster which has highest positive weight(similarity) with the point.

They also stated in their paper in case if the weight between the unseen instance(point not clustered) and the entire cluster detected previously is negative it means that there exist a new unseen cluster.
The original idea the authors[PP04] stated in their paper is given below:

Assume we are Given a graph $G = (V, E, w)$ where V, E, w denotes the set of points to be clustered, the edges between the points and edge weight between the points respectively.

Let $S \subseteq V$ be a subset of vertices which is dominant in the original graph G(which formed clusters previously), $\hat{V}$ the set of unseen instances and let $i \subseteq \hat{V} \setminus V$ a member of unseen instances $\hat{V}$. In order to add(assign) the new point $i \subseteq \hat{V}$ to the dominant set(cluster) S; first we have to calculate $W_{s\cup\{i\}}(i)$ and then we have to examine the sign of result. According to their proposal [PP04] if the sign of $W_{s\cup\{i\}}(i)$ is positive it indicates that the point is tightly coupled with the vertices's in S where as; if the sign is negative it indicates that the point is loosely coupled to the vertices in S. Finally, they proposed the following rule for predicting cluster membership of unseen data:

$$\text{if } \mathrm{W}_{s\cup\{i\}}(i) > 0, \text{ then assign vertex i to cluster S .} \tag{6}$$

According to rule 6 the point assigned to the cluster which has positive weight with it. Note that, according to this rule the same point can be assigned to more than one class, thereby yielding a soft partition of the input data. However, to get a hard partition they recommend to use the cluster membership approximation measures. They also mentioned that it can also happen for some instance i where no cluster S satisfies rule(6), in which case the point gets unclassified(or assigned to an "outlier" group). This should be interpreted as an indication that either the point is too noisy or the cluster formation process was inaccurate. We will use this concept in the second step of our approach.

Chapter 4

Related works

The fundamental problem in cluster analysis is to know how many clusters are appropriate for describing a given system. It is also the basic input for many clustering algorithms. Different scholars proposed a variety of methods on how to estimate the number of clusters. Gordon [Gor99] groups the approaches for determining the number of clusters into global and local methods. The global method evaluates some measure over the entire dataset and optimize it as a function of the number of clusters. Where as the local consider individual pairs of clusters and test whether they should be merged or not.

The drawback of most global method is that there is no direction for whether the data should be partitioned (best number of cluster is greater than 1) or not. However, it will not be a problem if users have good reasons to believe that there are clusters present in the data. The local methods are intended to test the hypothesis that a pair of clusters should be merged or not. They are suitable for assessing only hierarchically-nested partitions. According to Gordon comments, the significance levels should not be interpreted strictly since multiple tests are involved in the procedure.

4.1 Global methods

Calinski and Harabasz's method

Calinski and Harabasz's [CH74] in their work proposed a method for determining number of clusters based on an index called calinski harasbansz's (CH(g)). Which is defined as follows:

$$CH\,(g)=\frac{B(g)/(g-1)}{W(g)/(n-g)}$$

Where; B(g) and W(g) are between-cluster and within-cluster sum of squared errors, for g clusters. The mathematical formulation for B(g) and W(g) is defined as follows: Suppose we have a multivariate data containing n objects in p dimensions. Each object can be expressed as $x_i=(x_{i1}, x_{i2}, ..., x_{ip})^{'}$, i = 1,..., n. The dispersion matrix for each group is defined as:

$$W_m = \sum_{l=1}^{n_m} (x_{ml} - \bar{x_m}) (x_{ml} - \bar{x_m})', m = 1, ..., g.$$

Then the pooled within-group dispersion matrix W is defined by:

$$W = \sum_{m=1}^{g} \sum_{l=1}^{n_m} (x_{ml} - \bar{x_m}) (x_{ml} - \bar{x_m})'$$

The between-group dispersion matrix is defined as:

$$B = \sum_{m=1}^{g} n_m (\bar{x}_m - \bar{x}) (\bar{x}_m - \bar{x})', \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Calinski and Harabasz's stated that, the value which maximizes the CH(g) over g is the optimum number of the clusters.

According to the comparative study conducted by Milligan and Cooper [MC85] on 30 methods of determining the number of clusters in data, this method generally outperformed the others[Yan05].

Hartigan's method

Hartigan [Har75] proposed a method which is based on the following index:

$$Har\,(g) = \left[\ \frac{W(g)}{W(g+1)} \text{ - } 1\ \right] / (n - g = 1)$$

In their work, [Har75] proposed to calculate the value of Har(g) starting from g=1 and adding the cluster if the value of Har(g) is significantly large. A more simple decision rule suggested by Hartigan is to add a cluster if Har(g) is greater than 10 [Yan05]. For more detail it is advisable to refer [Har75].

Silhouette statistic

In order to estimate the optimum number of clusters of a dataset Kaufman and Rousseeuw [Rou87] proposed the silhouette index. The definition of the silhouette index is based on the silhouettes introduced by Rousseeuw [Rou87], which are constructed to visualize graphically how well each object is classified in a given clustering output. To plot the silhouette of the m^{th} cluster, for each object in C_m, it calculate s(i) in the following way:

$$S\,(i) = \frac{a(i) - b(i)}{max\{a(i), b(i)\}}$$

where:

a(i)=average dissimilarity of object i to all other objects in the m^{th} cluster.

$$\text{b(i)} = \min_{C \neq C_m} d(i, C)$$

d(i,C)=average dissimilarity of object i to all other objects in cluster C; C $\neq C_m$

Then we calculate the average of s(i)($\bar{S}$(g)) for all objects in the data which is also called the average silhouette width for the entire data set. This value reflects the within-cluster compactness and between-cluster separation of a clustering.

Compute $\bar{S}$(g) for g = 1, 2,...(for all number of clusters which is assumed to be optimum)

and select the one which maximizes $\bar{S}$(g). According to [Rou87] the value of g which maximizes the average silhouette index($\bar{S}$(g)) is the optimum number of cluster of a dataset. [Yan05]

$$Optimumnumberofcluster(\hat{G}) = \arg\max_g \bar{S}(\mathrm{g})$$

Gap method

Tibshirani et al. [TWH00] proposed an approach for estimating the number of clusters(k) in a data set via the gap statistic. The main idea of the gap method is to compare the within-cluster dispersions in the observed data to the expected within-cluster dispersions assuming that the data came from an appropriate null reference distribution. The best value of k is estimated as the value $\hat{k}$, such that log(W($\hat{k}$)) falls the far below its expected curve. The formulation for the gap method is defined as:

$$\mathrm{Gap}_n\,(k) = E_n^* log(W(k)) - log(W(k))$$

Where, $E_n^* log(W(k))$ indicates the expected value of log(W(k)) under the null distribution. The value of k which maximizes $Gap_n(k)$ is the optimum number of clusters, $\hat{k}$. For detail explanation look [TWH00]

4.2 Local methods

In this section we will discuss two local methods used for estimating the number of clusters, which are among the top 5 best performing algorithms according to the comparative study of milligan and cooper's [MC85]. The first one is proposed by Duda and Hart [DH73], in their method the null hypothesis that the mth cluster is homogeneous is tested against the alternative that it should be subdivided into two clusters. The test is based on comparing the within-cluster sum of squared errors of the mth cluster; $J_1^2(m)$ with the within-cluster sum of squared distances when the mth cluster is optimally divided into two $J_2^2(m)$. If the mth cluster contains n_m objects in p dimensions, then the null hypothesis will be rejected if:

$$J_1^2(m)/J_2^2(m) < 1 - 2/(\pi p) - z\left[2(1 - 8/(\pi^2 p))/(n_m p)\right]^{\frac{1}{2}}$$

Where z is the cutoff value from a standard normal distribution specifying the significance level[Yan05].

The second method proposed by Beale [Bea69] tests the same hypothesis with a pseudo-F statistic. Which is given by:

$$F \equiv \left(\frac{J_1^2(m) - J_2^2(m)}{J_2^2(m)}\right) \Big/ \left(\left(\frac{n_m - 1}{n_m - 2}\right) 2^{\frac{2}{p}} - 1\right)$$

The homogeneous cluster hypothesis is rejected if the value of the F statistic is greater than the critical value from the $F_p, (n_m - 1)p$ distribution. In both tests given the rejection of the null hypothesis, it follows that the subdivision of the mth cluster into two sub clusters is significantly better than treating it as a single homogeneous cluster [Yan05].

In addition to the approaches stated above, Broom et al [CVoSDoPS92] proposed an approach for determining the greatest possible number of local maxima that a quadratic form can have when the vector is constrained within the unit simplex. The quadratic program has the following form:

$$\mathrm{V}=\mathrm{p}^T\mathrm{Ap}$$

where: $(p_1, p_2, ...p_n) \in \triangle_n = \{\mathrm{x} \in \mathbb{R}^{\kappa} : \mathrm{x}_i \geq 0, \sum_i x_i = 1$ and $\mathrm{A}=(\mathrm{a}_{ij})$ is a real, symmetric matrix. for detail explanation refer to [CVoSDoPS92]

Chapter 5

Our contribution

5.1 The Proposed Approach

To achieve our objective, we implement a two-step approach where in the first step, we tried to detect the (minimum)number of cluster automatically, in the second step we cross checked if there exists a structure which has to be clustered but not yet detected in the first step. Finally, after meeting our objective, in order to check whether our approach determines the right number of clusters we propagated the class labels using graph transduction to unlabel instances and to analyze the corresponding output. The steps are explained in detail as follows:

Step 1:Detect Number of Cliques

Let $G = (V; E)$ be an undirected graph without self-loops, where $V = 1, 2, ..., n$ is the set of vertices and $E \subseteq V \times V$ the set of edges. We define the order of a graph G as the cardinality of V . Two vertices $u, v \in V$ are adjacent if $(u, v) \in E$. A subset C of vertices in G is called a clique if all its vertices are mutually adjacent. It is a maximal clique if it is not a subset of other cliques in G. It is a maximum clique if it has maximum cardinality. The cardinality of a maximum clique of G is also called clique number and denoted by $w(G)$, It should be mentioned that the number of maximal clique and stability number are not always equal. more precisely, stability number is lower or equal to the number of maximal clique

The adjacency matrix of G is the $n \times n$ symmetric matrix $A_G = (a_{ij})$, where $a_{ij} = 1$ if $(i, j) \in E$, $a_{ij} = 0, otherwise$. $\bar{A}$ is defined as dissimilarity matrix .

The adjacency matrix of an undirected graph can be regarded as the similarity matrix of a clustering problem and compliment of a graph G (disimilarity) is defined as $\bar{A} = 1 - A_G$ for unweighted case therefore our framework can be used to find the stability number.

Consider the following constrained quadratic program derived from weighted Motzkin-Strauss formulation.

$$\begin{aligned} \frac{1}{w(G)} = \text{minimize} \quad & x^T(\bar{A} + \alpha I)x \\ \text{subject to} \quad & X \in \triangle \subset \mathbb{R}^n \end{aligned} \tag{5.1}$$

With $$\triangle = (x \geq 0 \quad and \quad e^T = 1)$$

where n is the order of G, I the identity matrix, α is a real parameter and $\triangle$ is the standard simplex of the n-dimensional Euclidean space.

In 1965, Motzkin and Straus [MS65] established a connection between the maximum clique problem with $\alpha = 0$. Specifically, they related the clique number of G to global solutions x^* of the program through the formula $w(G) = (1 - f_0(x^*))^{-1}$, and showed that a subset of vertices C is a maximum clique of G if and only if its characteristic vector $x^C \in \triangle$ is a global maximizer of f_0 on $\triangle$. Pelillo and Jagota [PJ95], extended the Motzkin-Straus theorem by providing a characterization of maximal cliques in terms of local maximizers of f_0 in $\triangle$.

A drawback of the original Motzkin-Straus formulation is the existence of spurious solutions, maximizers of f_0 over $\triangle$ that are not in the form of characteristic vectors. This was observed empirically by Pardalos and Phillips [PP90] and formalized later by Pelillo and Jagota [PJ95]. In principle, spurious solutions represent a problem, while providing information about the order of the maximum clique, does not allow us to easily extract its vertices. Fortunately, there is a straightforward solution to this

problem which has been introduced by Bomze [Bom97] . He, indeed, suggested to add a constant α on the diagonal of the adjacency matrix of the graph and basically proved that for $0 < \alpha < 1$ all local maximizer of 5.1 are strict and in one-to-one correspondence with the characteristic vectors of the maximal cliques of G. In our case as reported in the paper of Pelillo and Jagota [PJ95], α sets to a value close to 1.

In the weighted case, compliment Graph or weighted disimilarity matrix is calculated by

$$\bar{A_{ij}} = exp(-\frac{||F(i) - F(j)||^2}{\sigma^2}) \tag{5.2}$$

Equation 5.1 is designed for unweighted matrices by motzkin-strauss. We have extended their work to the weighted version. We have observed that in this way the minimum number of maximal cliques can be obtained.

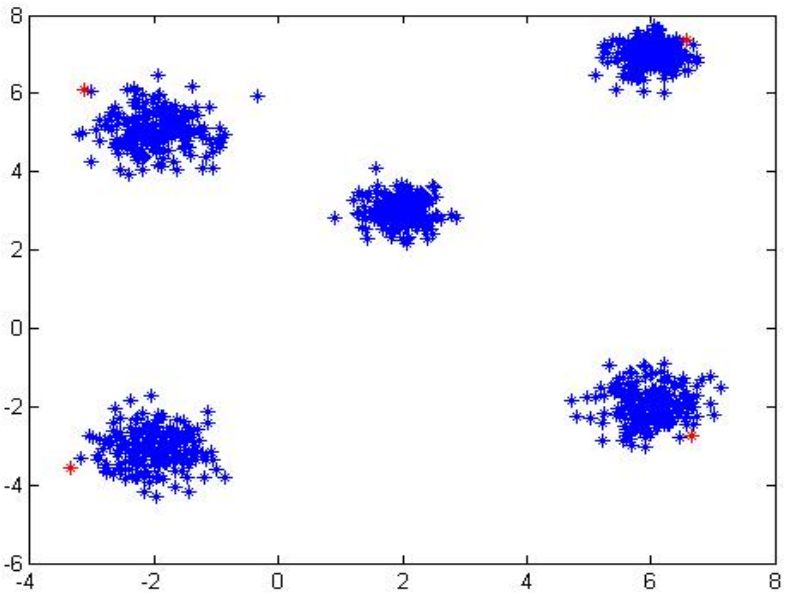

Fig. 5.1: *cluster representatives detected at step 1*

Step 2:checking the existence of new unseen cluster

As we have seen from multiple execution outputs on different computer generated datasets, of the first step we analyzed that there is a situation where we could not be able to detect all cluster representatives. For example if there exist a cluster between clusters. In order to detect such type of clusters we have implemented the idea of Efficient Out-of-Sample Extension of Dominant-Set Clusters which enables us to indirectly detect if there exists new (unseen) cluster representative. As stated in the paper [PP04] while predicting the class membership of new instances of the dataset, if the instances do not have positive similarity weight (W(s)i) with one of the cluster which is already known it means that the classification is inaccurate or there is a cluster which is unseen.

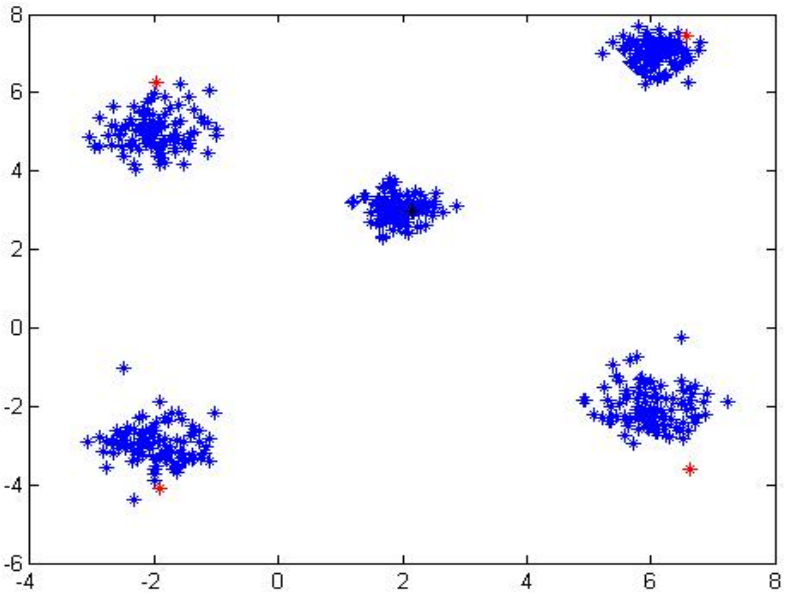

Fig. 5.2: *cluster representative detected using step 2*

Note: the black dot over the figure is the cluster representative detected by applying the idea of Efficient Out-of-sample Extension of Dominant-Set

5.2 Why Efficient Out-of-Sample Extension of Dominant-Set Clusters?

As we can see from the above figure, efficient-out-of-sample serves our purpose very well. It answers one of our basic question "is there any new unseen cluster or not?". As it has been explained in the paper [PP04], in order to predict the class membership of new instance (node), which is unseen, we have to check the sign of the weight of the node to be added to the existing cluster. While the node with positive weight is added to the its corresponding cluster, the cluster which gives the highest weight to it, the node with negative weight is considered as either an outlier or a representative for other unseen cluster group. For detail explanation we refer the reader to [PP04].

5.3 Experiments and Experimental Results:

In order to test the effectiveness of our approach, we have conducted an experiment on different toy, generated using different Matlab functions, real, downloaded from UCI machine learning page, `ics.uci.edu/ml/datasets.html` and social network datasets. For all the datasets our framework is compared against well known clustering algorithms: Normalized Cut and K-Means, and the results are very encouraging. While our first part of the experiment covers the experiments done on toy datasets, the second part consists the real dataset descriptions and the experiments done on real datasets. The last part consists of social network dataset descriptions and experiments done on them. For those datasets that have ground truth, the label which help us to know how much our framework is accurate, we first remove their labeling information before doing any experiment.

In all parts of our experiment, we have tried to show the detail procedure of the experimental process and their outputs, summarized in different tables. The algorithm, to find dominant set clusters, follows easy steps: It first identifies the number of clusters, it then use graph transduction to get the whole clustering result which we used to show effectiveness of our approach by comparing it against K-Means and Normalized Cut algorithms.

The datasets used to test the performance of our approach are summarized below.

		Data sets			
Data source	Computer generated	Name	Instances	Features	number of clusters
		FG	500	2	5
		Banana	500	2	2
		TGC	450	2	3
		SG	700	2	7
	UCI	Iris	150	4	3
		Ionosphere	351	33	2
		Pima	768	8	2
		Ecoli	272	7	2
		Soybean	136	35	4
		Liver	345	6	2
		Haberman	306	3	2
	social network	Karate	34	-	2
		Dolphins	62	-	2
		Food	45	-	7
		Collaboration	235	-	-
		Jazz Musician	198	-	-

Table 5.1: *List of datasets used for testing our approach*

When we come to our first part of the experiment, as we have discussed above, our first move is to test our framework on different toy datasets generated by ourselves. For this purpose we have generated three different multivariate normal distribution datasets with three, five and seven cluster groups each and one elongated banana shaped dataset.

Our first trial was on the first toy dataset which is a multivariate normal distribution data, with five cluster groups, generated using Matlab *mvnrnd*() function with means(mu1=[0 9],mu2=[-7,-5], mu3=[17 4], mu4=[4 1], and mu5=[12 -6]), and covariance of [1 0; 0 1]. Each of the five clusters has 100 instances and 2 features. In order to investigate the result of our approach we have done several experiments.

Five Multivariate Gaussian dataset(FG)

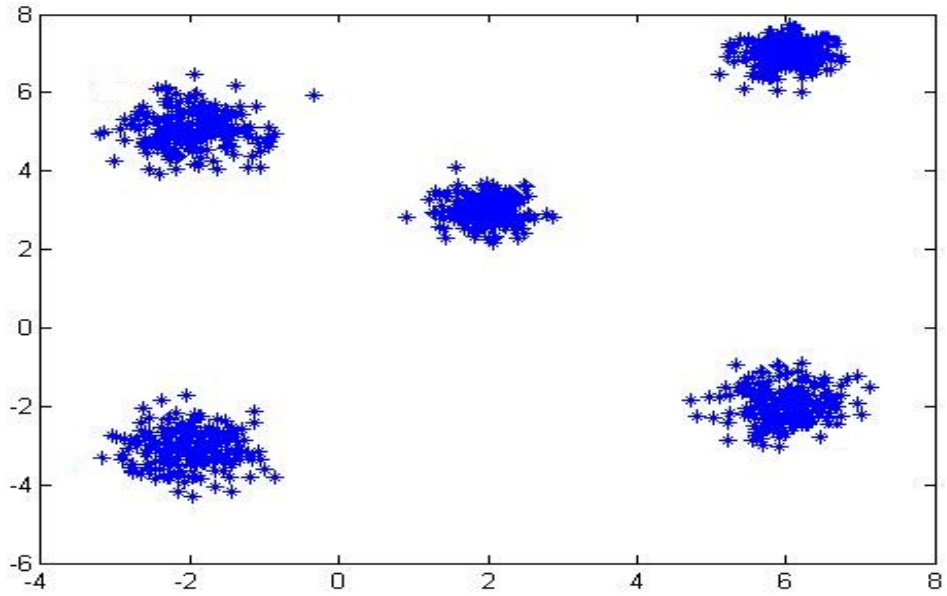

Fig. 5.3: *five gaussians dataset generated using matlab mvnrnd() function*

As we have discussed above, to clarify our experimental part in an easy way, we have tried to show outputs of different parts of the algorithm. In the first step, the algorithm identifies the farthest points detected which are cluster representatives. As we may not find all the cluster representatives in the first step, we have done a second step as described in 5.1.

Step 1: Identifying the farthest points,

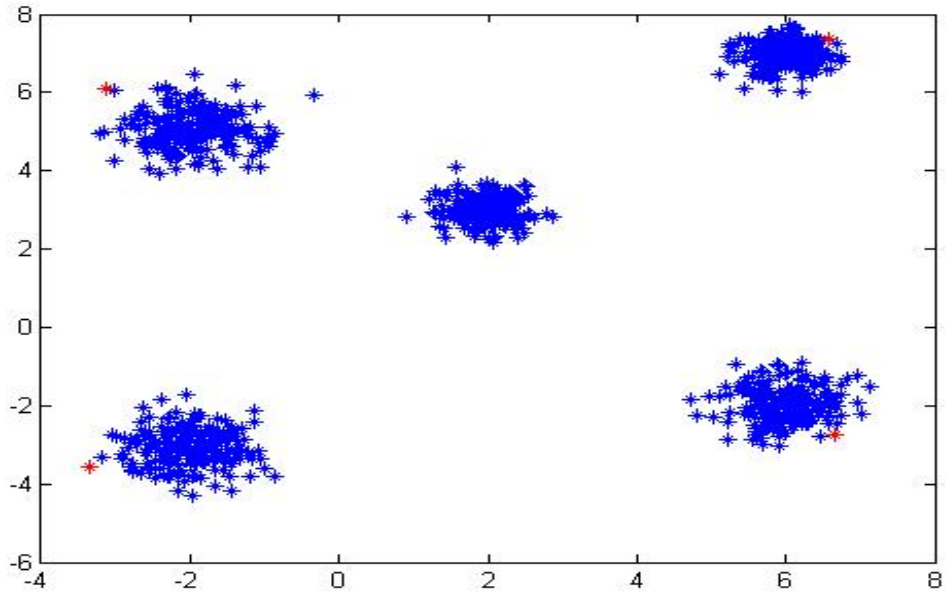

Fig. 5.4: *Cluster representatives detected by applying step 1 on FG data set*

As we can see in the figure the four red dotes plotted over the cluster are the farthest points detected using first step (5.1) of our algorithm, the first step can't get the whole cluster representatives.

Step 2:

From fig 5.4 we can see that there is one cluster representative which is not detected in the first step of our approach. So, in order to detect such a cluster representative, we apply the second step of our approach (i.e. 5.1). The result is shown in fig 5.5

From fig 5.5 we can see that our approach is able to determine the right number of clusters for five multivariate Gaussians datasets. After determining number of cluster (i.e after getting the cluster representative points) we use graph transduction to get the whole cluster structure. The result is plotted as shown in figure 5.6.

From this figure, we can see that our approach determines the right number of cluster and a very encouraging clustering result.

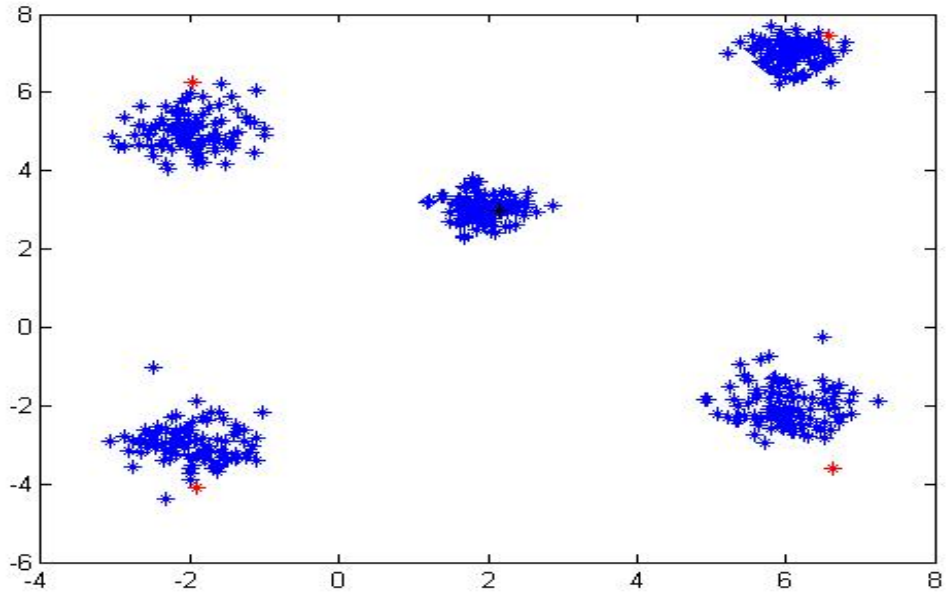

Fig. 5.5: *Cluster representatives detected by applying step 2 on FG data set; the black dot over the cluster are the cluster representative detected by applying step 2 on FG dataset*

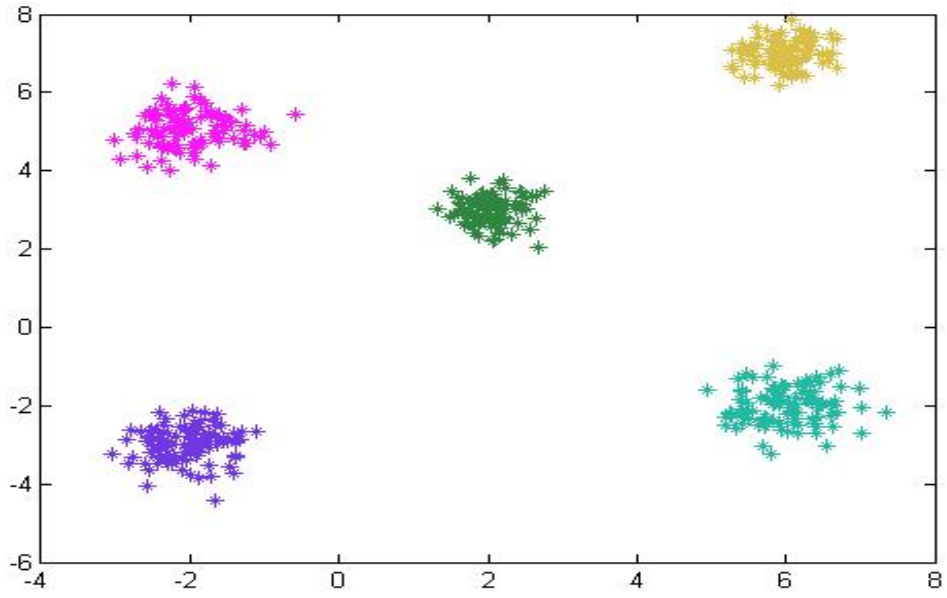

Fig. 5.6: *clusters recovered using graph transduction for Five Gaussians (FG) dataset*

Seven Multivariate Gaussian data set(SG)

Below is experimental result of applying our approach on seven Gaussians dataset

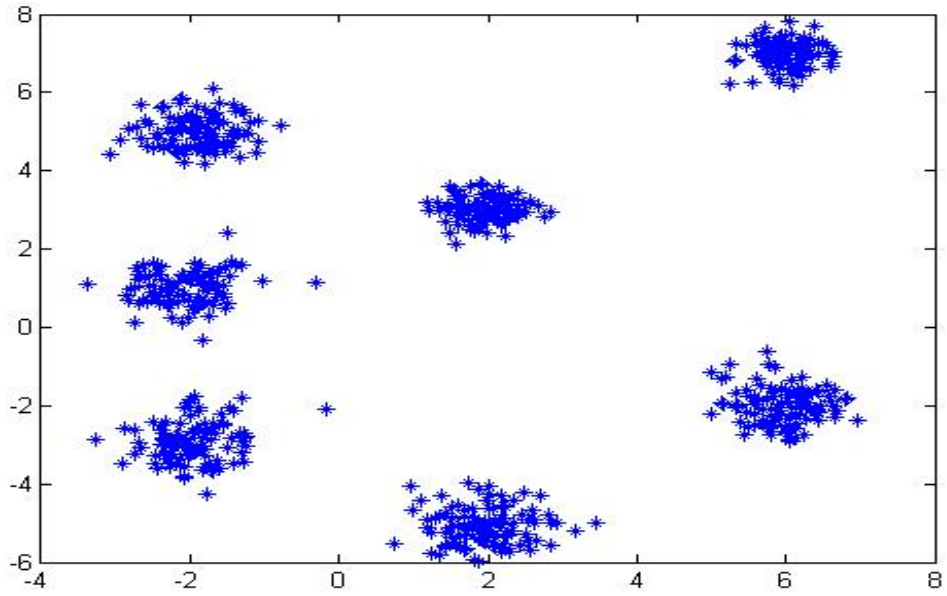

Fig. 5.7: *Seven Gaussians dataset generated using matlab mvnrnd() function*

The second toy dataset, as of the first is a multivariate normal distribution data but with seven cluster groups and different mean and variance, is also generated using *mvnrnd*() Matlab function with means(mu1=[2 3],mu2=[-2,-3], mu3=[6 7], mu4=[-2 5], mu5=[6 -2], mu6=[2 -5], mu7=[-2 1]), and covariance of (SIGMA1 = [0.1 0; 0 0.1], SIGMA2 = [0.2 0; 0 0.2],SIGMA3 = [0.1 0; 0 0.1], SIGMA4 = [0.2 0; 0 0.2],SIGMA5 = [021 0; 0 0.2], SIGMA6 = [0.2 0; 0 0.2],SIGMA7 = [0.2 0; 0 0.2]) for cluster 1 to 7 respectively. The dataset has seven well separated clusters. Each of them has 100 instances and 2 features. We have done several experiments on this data set and have got the following results. As of the previous experiment, we have tried to show the detail of each steps:

Step 1: Finding the farthest points

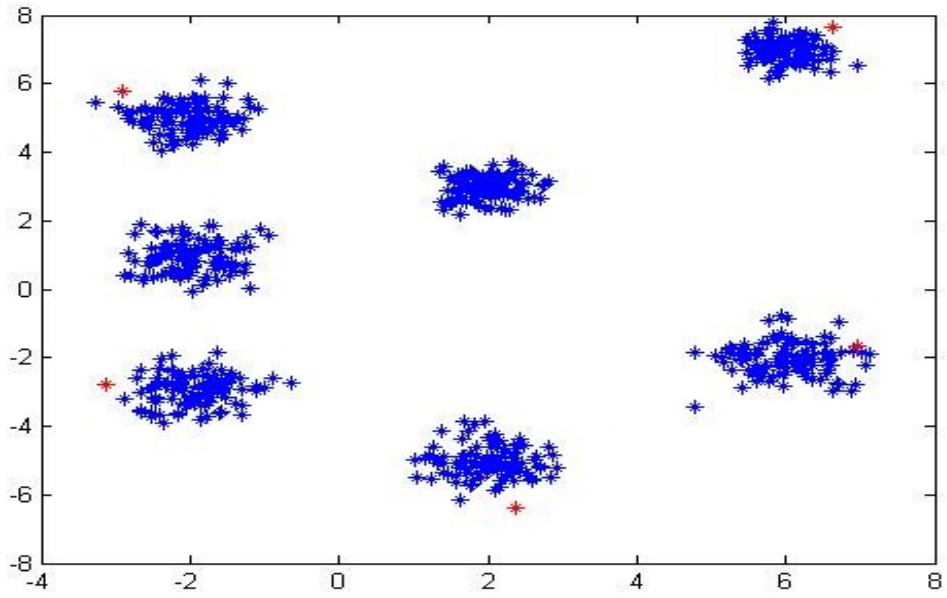

Fig. 5.8: *Cluster representatives detected by applying first step of our algorithm on seven gaussians(SG) dataset, the red dot over the figure is the cluster representatives detected by applying step 1 on seven gaussians(SG) dataset*

From the figure we can see five red dotes plotted over the clusters, these points are the cluster representatives detected by applying the first step (5.1) of our algorithm.

Step 2:

As we can see in the figure 5.8 there are two clusters representatives that are not detected in first step of our approach. So, in order to detect such cluster representatives, we applied the second step of our approach and we got the result shown in fig 5.9.

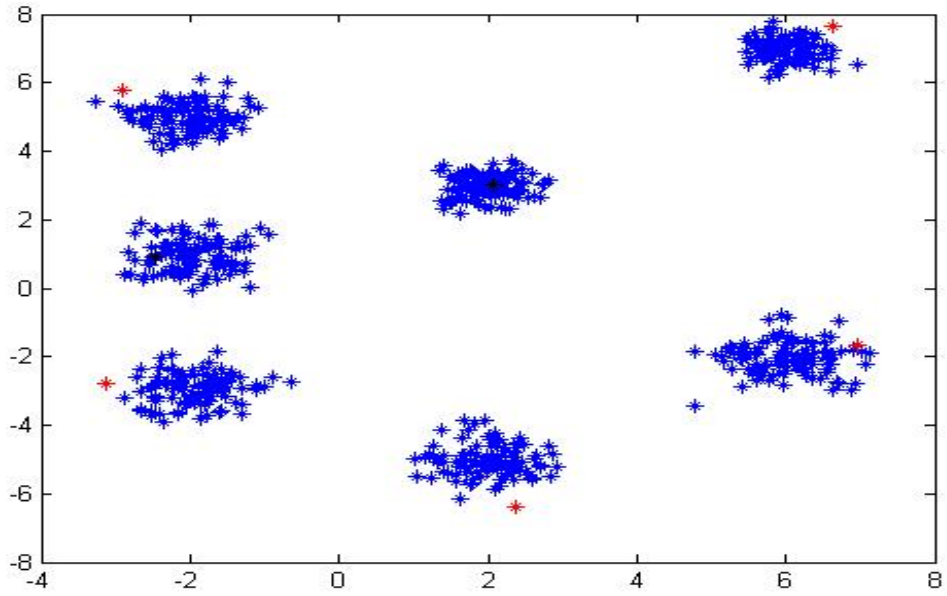

Fig. 5.9: *Cluster representatives detected by applying second step of our algorithm on SG dataset, the black dot over the figure is the cluster representatives detected by applying step 2 on the SG dataset*

From fig 5.9 we can see that, as of the first experiment, our approach is able to determine the right number of clusters for the SG datasets. This cluster representatives are used as a label for the graph transduction algorithm which help us get the final clustering result. The result of the second toy is plotted as shown in fig 5.10. We can see from this figure our approach, using the label information and graph transduction, is able to determine the right number of cluster and we get almost the same cluster structure as of the original one (fig 5.7)

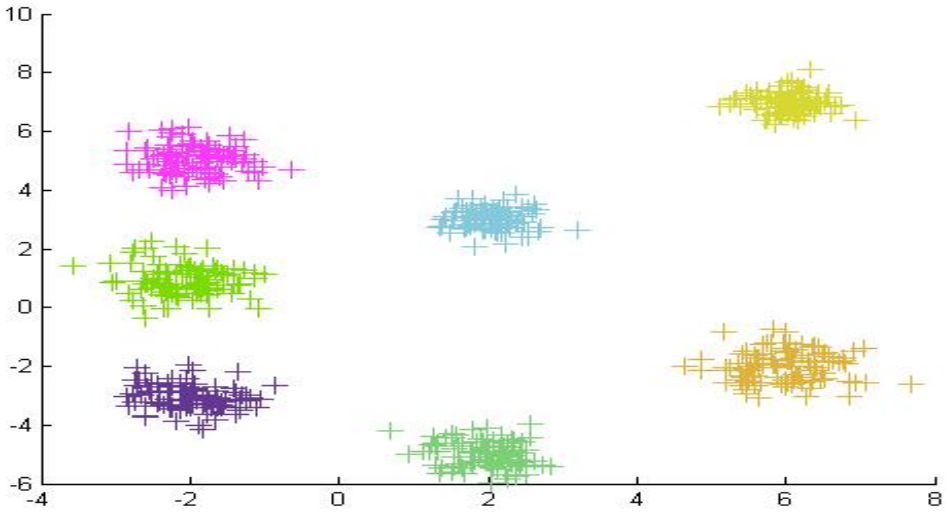

Fig. 5.10: *Cluster structure recovered using graph transduction for seven gaussians (SG) dataset*

Three clusters close to each other

The third toy dataset is generated to show how much powerful our framework is for a very near cluster groups and some overlaps as noise. The result of the experiment is shown in fig 5.11.

From fig 5.11 b, we can see that our approach is able to determine the right number of clusters for clusters closed to each other. From fig 5.11c we can see that by applying graph transduction we are able to get the cluster structure which are almost similar to the original one(fig 4.11 a), which is on the other hand the proof of our prediction.

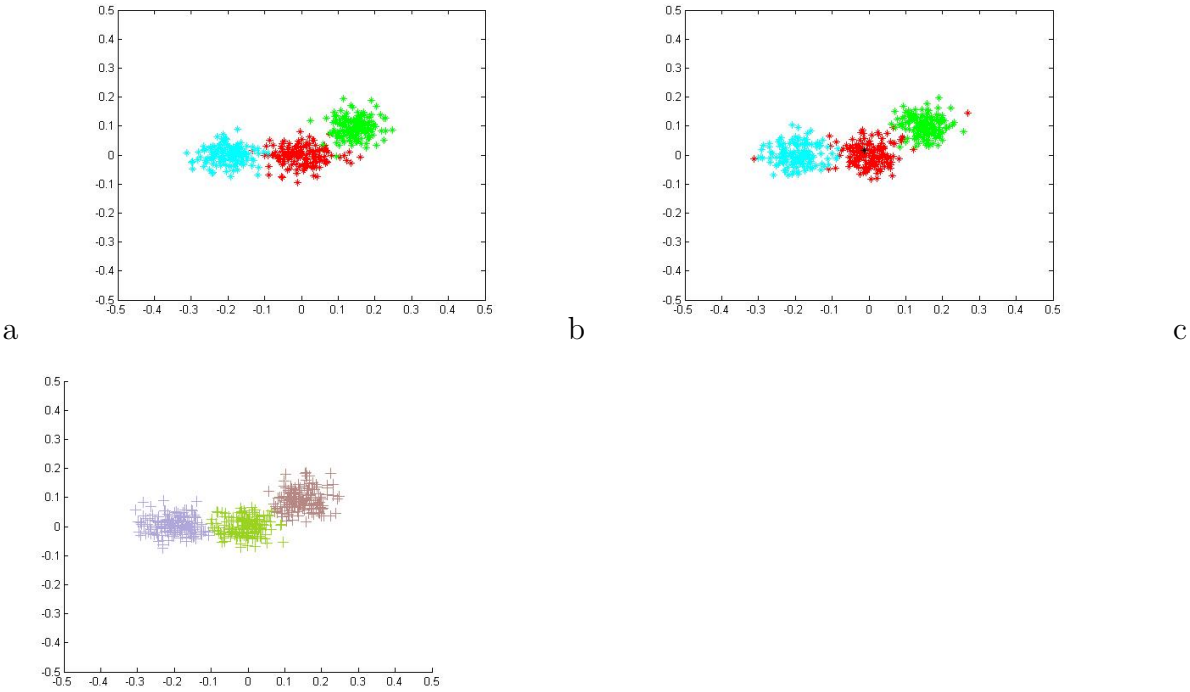

Fig. 5.11: *an example of determining the number of cluster automatically and forming the cluster by labeling using graph transduction. a)original cluster structure generated using matlab mvnrnd() function b)cluster representatives detected by applying our approach. note: the red dot is the one detected in first step and black dot is the one detected in second step. c)cluster structure recovered using graph transduction*

Elongated cluster

In this first part of our experiment using toy datasets, our last trial was on elongated structure. For this purpose we have considered the banana shape dataset from PRTools, http://prtools.org. The data generated is a 2-dimensional 2-class dataset of banana shaped with a uniform distribution of the data along the elongated structure.

For our purpose, we have generated the banana shaped data with a variance of 0.5 and 250 instances. We have done several experiment on this data set and the results are tabulated.

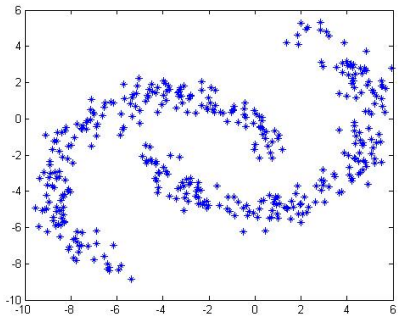

Fig. 5.12: *Banana structure dataset generated using matlab gendatb() function*

Step 1:

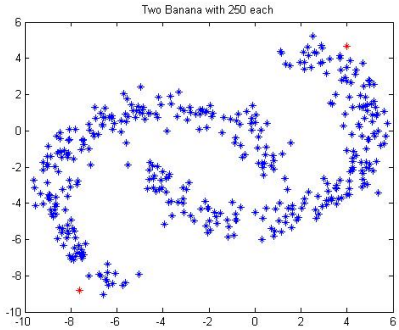

Fig. 5.13: *Cluster representatives detected by applying first step of our algorithm on banana data set, the red dotes over the figure is the cluster representatives detected by applying step 1 on banana data set*

From the output of the above figure we can see that all the cluster representatives are identified in the first step.

Step 2:

Although, it is trivial to see from fig 5.13 that there is no unseen cluster, to see what will happen in such a situation we run the second step.

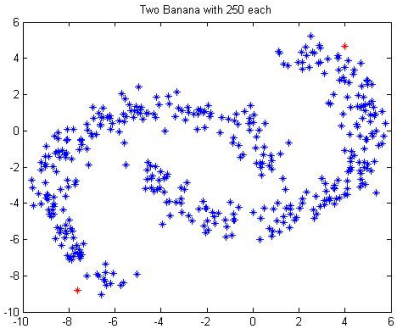

Fig. 5.14: *Cluster representatives detected by applying second step of our algorithm on banana data set*

From this result we can also conclude that the second step of our approach detects the cluster representatives only if there exist unseen cluster from the first step.

After getting the clusters representatives, we use graph transduction to diffuse the information and get the whole cluster structure. The result is plotted below.

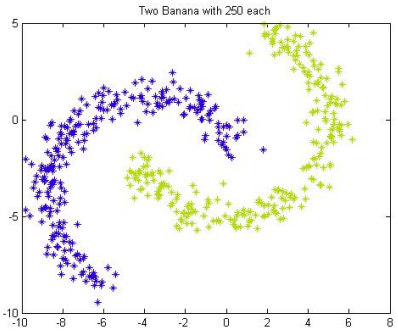

Fig. 5.15: *Cluster structure recovered using graph transduction for banana dataset*

From this experiment we can conclude that our approach is also effective for elongated structures.

To see the stability of our framework, we have run the algorithm different number of times on all toy datasets and see how many times it detects the correct number of cluster representatives. The result, which is really promising and interesting, is tabulated

below and we can see from the table most of the time our algorithm predict the number of clusters correctly.

dataset name	no of true cluster	no of run	no of times correctly predicted
FG	5	10	10
		50	48
		100	95
		200	190
		500	480
Banana	2	10	10
		50	50
		100	100
		200	200
		500	500
TGC	3	10	10
		50	48
		100	94
		200	195
		500	482
SG	7	10	10
		50	46
		100	94
		200	195
		500	484

Table 5.2: *Performance test of our approach on toy datasets(FG, SG, banana, TGC)*

In the next part of our experiment we are going to see the performance of our algorithm on real datasets from UCI. Before doing an experiment on the UCI datasets listed in Table 5.1 on page 30 it is better to give the description of them, it helps the reader to know the performance of the framework very well.

Iris dataset: this dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two which are not linearly separable from each other. Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
    - Iris Setosa
    - Iris Versicolour
    - Iris Virginica

To say more about this dataset, many people have been using it as a test case for many classification and clustering techniques in machine learning. When we see it as two cluster (based on the linearly separable regions) one of the clusters contains Iris Setosa, while the other cluster contains both Iris Virginica and Iris Versicolor and is not separable without the species information. Here is the 3D plot of the dataset with the first three columns as points in the feature space and the fourth column as colour.

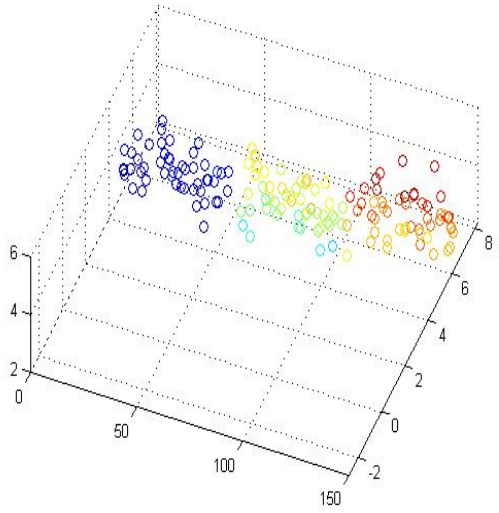

Fig. 5.16: *Iris Dataset Plot*

Wine Dataset. This dataset is the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. It consists of three different varieties of wine and the analysis determined the quantities of 13 constituents found in each of the three types of wines. The dataset can be used in many classification and clustering analysis. Attribute information:

1. Alcohol
2. Malic acid
3. Ash
4. Alcalinity of ash
5. Magnesium
6. Total phenols
7. Flavanoids

8. Nonflavanoid phenols
9. Proanthocyanins
10. Color intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline
14. class:
    - class 1
    - class 2
    - class 3

Ecoli dataset: This dataset is a dataset used for protein classification. It includes 336 instances. Each of the attributes used is a score(between 0 and 1) corresponding to a certain feature of the protein sequence. The higher the score is, the more possible the protein sequence has such feature. In this dataset, seven features(attributes) are used: mcg, gvh, lip, chg, aac, alm1, alm2. Proteins are classified into 8 classes: cytoplasm(cp), inner membrane without signal sequence(im), perisplasm(pp), inner membrane with uncleavable signal sequence(imU), outer membrane(om), outer membrane lipoprotein(omL), inner membrane lipoprotein(imL), inner membrane with cleavable signal sequence(imS). Before applying our algorithm on this dataset we remove instances which belong to class outer membrane (om), outer membrane lipoprotein (omL), inner membrane lipoprotein (imL), inner membrane with cleavable signal sequence (imS) which has 20, 5, 2, 2 instances respectively.

Let's say something about other datasets from UCI used on our experiment. The dataset Pima is the result of diabetes test on at least 21 years old females pima heritage indians. This data is used to classify whether the person is diabetes positive or not based on 8 different factors. The other dataset is the Haberman which contains cases from study conducted on the survival of patients who had undergone surgery for breast cancer and it has two classes. The two classes are if the patient survived 5 years or longer or if he died within 5 year. The Ionosphere dataset, is dataset for classification of radar returns from the ionosphere. The radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere whereas, "Bad" returns are those that do not; their signals pass through the ionosphere.

A similar step-by-step experimental analysis, as on the toy dataset, have been conducted on the real datasets: Cluster representatives are identified by the two step processes and the information we got are diffused to the rest of the samples using graph trasduction technique which help us get the final clusterings which we have used to compare against other well known clustering algorithms: Normalized Cut and K-means. The result is table or shown in fig 5.19:

dataset name	no of true cluster	no of run	no of times correctly predicted
Iris	3	10	9
		50	47
		100	95
		200	190
		500	480
Ionosphere	2	10	10
		50	48
		100	97
		200	195
		500	485
Haberman	2	10	10
		50	48
		100	94
		200	195
		500	482
Pima	2	10	10
		50	46
		100	94
		200	195
		500	484
liver	2	10	9
		50	49
		100	47
		200	180
		500	454
Ecoli	3	10	10
		50	46
		100	94
		200	195
		500	484
Soybean	4	10	10
		50	45
		100	90
		200	185
		500	480

Table 5.3: *Performance test of our approach on real data set(UCI) i.e. Iris, Ionosphere, Pima, Haberman, Wine, Ecoli, Soybean and Liver dataset*

In the last part of our experiment we evaluated our algorithm on social network data set from UCI network repository, http://www-personal.umich.edu/~mejn/netdata/. Before doing an experiment on the social network dataset listed in Table 5.1 we think it is better to make the reader familiar at least with some of them .

Karate: is the data set of zachary's network of karate club members [Zac77], a well-known graph regularly used as a benchmark to test community detection algorithms. It consists of 34 vertices, the members of a karate club in the United States, who were observed during a period of three years. Edges connect individuals who were observed to interact outside the activities of the club. At some point, a conflict between the club president and the instructor led to the fission of the club in two separate groups, supporting the instructor and the president, respectively (indicated by blue and red circles in fig 5.17). [For10]

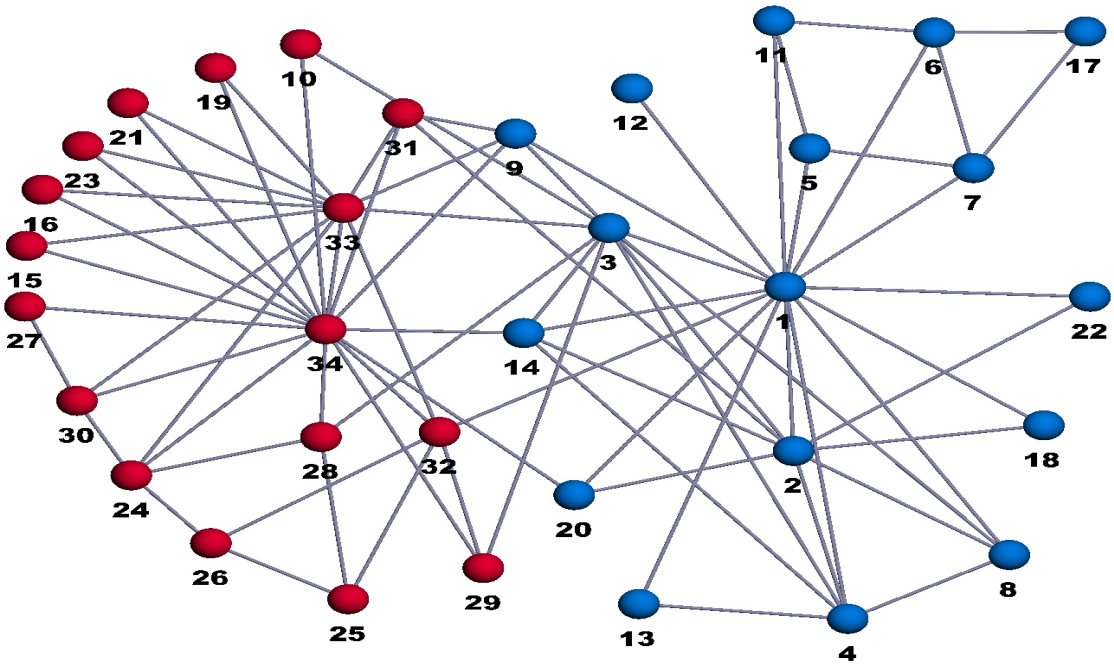

Fig. 5.17: *Zachary's karate club network*

Dolphins: is the data set of the network of bottlenose dolphins living in Doubtful Sound (New Zealand) analyzed by Lusseau. There are 62 dolphins and edges were set between animals that were seen together more often than expected by chance. The dolphins separated in two groups after a dolphin left the place for some time (squares and circles in the figure,fig 5.18). Lusseau's dolphins' network, like Zachary's karate club, is often used to test algorithms for community detection. [For10]

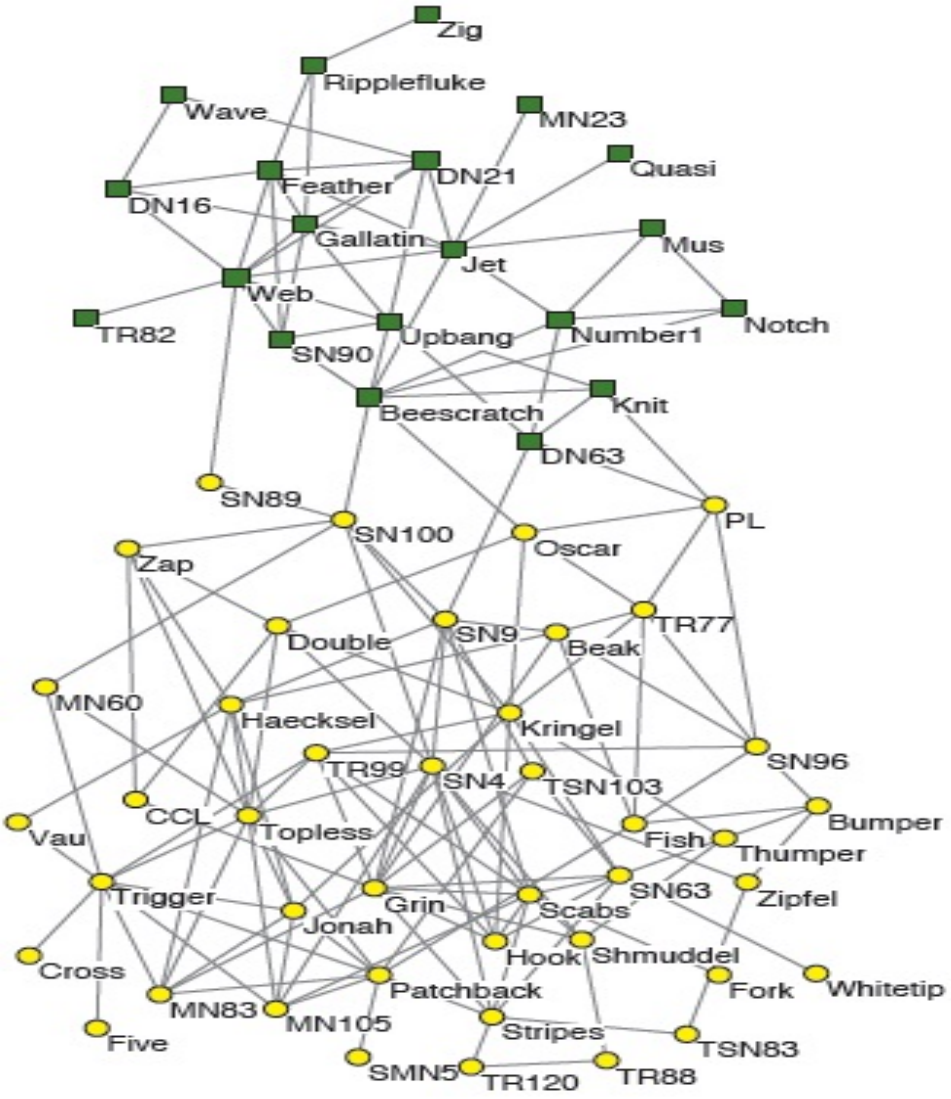

Fig. 5.18: *Lusseau's network of bottlenose dolphins.*

A similar step-by-step experimental analysis, as on the toy and real dataset, have been conducted on the social network datasets: Cluster representatives are identified by the two step processes and the information we got are diffused to the rest of the samples using graph transduction technique, which help us get the final clusterings. However, in this case the accuracy of final clustering result is compared only against Normalized Cut since, K-means does not support dataset represented in terms of graph. The final result is tabulated in fig 5.19:

		Datasets						
				Number of classes		Accuracy		
		Name	Instances	Original	Detected	Our method	K-means	N-Cut
Data Source	Toy	FG	500	5	5	99.8	77	100
		Banana	500	2	2	94	75	94
		TGC	450	3	3	98	98	67
		SG	700	7	7	99.6	73	100
	UCI	Wine	178	3	3	69	60	71
		Iris	150	3	3	86	86	90
		Ionosphere	351	2	2	70	71	68
		Pima	768	2	2	67	66	61
		Ecoli	272	4	4	94	78	76
		Soybean	136	4	4	78	68	82
		Liver	345	2	2	58	55	53
		Haberman	306	2	2	75	52	51
		Auto_mpeg	398	3	3	74	64	70
	Social network	Karate	34	2	2	88		85
		Dolphin	62	2	3	78		78

Fig. 5.19: *Overall Experimental result: this table shows the number of cluster determined by our approach and the accuracy of cluster formed by graph transduction , K-Means and N-Cut for toy, real and social network datasets.*

To sum up, as can be seen from all the experimental results, our framework performs very well for all types of datasets, specially for the social

Chapter 6

Conclusion and Future work

We presented a system which performs automatic determination of number of cluster using the concept of Dominant set approach as stated on the papers of Massimiliano Pavan and Marcello Pelillo [PP07]and [PP04]. We showed qualitative and quantitative output results of our algorithm by making tests on different computer generated datasets, UCI repository datasets and social network datasets and we got an interesting and promising result.

Social network analysis is the study of relationship among entities in a certain society. It aims at extracting group of entities called communities, which have high relationship among each other and less with the others. In the last decade, numerous classic graph clustering methods have been adapted for community detection which are namely: random walks, spectral clustering, modularity maximization, differential equations and statistical mechanics. One of the main drawbacks exhibited in most of the methods for community detection is that, the number of community k should be specified in advance. Therefore as a future work we want to extend our approach for detecting the number of communities or cluster before the community detection process takes place.

To see the effectiveness of our future work, we have done a preliminary experiment on karate dataset(zachary's network of karate club members) and dolphin dataset(network of bottlenose dolphins) ,social network datasets, and we got promising result.

Bibliography

[Bea69] E.M.L. Beale. *Euclidean Cluster Analysis*. Scientific Control Systems Limited, 1969. 23

[Bom97] Immanuel M. Bomze. Evolution towards the maximum clique. *J. of Global Optimization*, 10(2):143–164, March 1997. 27

[CH74] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-Simulation and Computation*, 3(1):1–27, 1974. 20

[CVoSDoPS92] C. Cannings, G.T. Vickers, University of Sheffield. Department of Probability, and Statistics. *On the Number of Local Maxima of a Constrained Quadratic Form*. University of Sheffield, Department of Probability and Statistics, 1992. 24

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Willey & Sons, New Yotk, 1973. 2, 23

[EP12] Aykut Erdem and Marcello Pelillo. Graph transduction as a noncooperative game. *Neural Computation*, 24(3):700–723, 2012. 10

[For10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010. 47, 48

[Gor99] A.D. Gordon. *Classification, 2nd Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1999. 19

[Har75] John A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition, 1975. 21

[JD88] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988. 7

[MC85] Glenn Milligan and Martha Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985. 20, 23

[MS65] Theodore S Motzkin and Ernst G Straus. Maxima for graphs and a new proof of a theorem of turán. *Canad. J. Math*, 17(4):533–540, 1965. 26

[PJ95] Marcello Pelillo and Arun Jagota. Feasible and infeasible maxima in a quadratic program for maximum clique. *J. Artif. Neural Networks*, 2:411–420, 1995. 26, 27

[PP90] Panos M Pardalos and AT Phillips. A global optimization approach for solving the maximum clique problem. *International Journal of Computer Mathematics*, 33(3-4):209–216, 1990. 26

[PP04] Massimiliano Pavan and Marcello Pelillo. Efficient out-of-sample extension of dominant-set clusters. In *NIPS*, 2004. 2, 17, 18, 28, 29, 50

[PP07] Massimiliano Pavan and Marcello Pelillo. Dominant sets and pairwise clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):167–172, 2007. 13, 14, 15, 50

[Rou87] Peter Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, November 1987. 21, 22

[SM97] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997. 9

[TWH00] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a dataset via the gap statistic. 63:411–423, 2000. 2, 22

[Yan05] Mingjin Yan. *Methods of Determining the Number of Clusters in a Data Set and a New Clustering Criterion.* PhD thesis, Virginia Polytechnic Institute and State University,Blacksburg, Virginia, 2005. 20, 21, 22, 23

[Zac77] W.W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977. 47