T.C.
BAHÇEŞEHİR UNIVERSITY

AGGREGATING ADVANTAGES OF A SET OF CLUSTERINGS INTO A FINAL CLUSTERING USING OBJECT-WISE SIMILARITY GRAPH

Master of Science Thesis

Ertunç ERDİL

Istanbul, 2011

T.C.
BAHÇEŞEHİR UNIVERSITY
The Graduate School of Natural and Applied Sciences
Computer Engineering

AGGREGATING ADVANTAGES OF A SET OF CLUSTERINGS INTO A FINAL CLUSTERING USING OBJECT-WISE SIMILARITY GRAPH

Master of Science Thesis

Ertunç ERDİL

Supervisor: Asst. Prof. Dr. Selim Necdet MİMAROĞLU

Istanbul, 2011

T.C.
BAHÇEŞEHİR UNIVERSITY
The Graduate School of Natural and Applied Sciences
Computer Engineering

Title of the Master's Thesis	:	Aggregating Advantages of a Set of Clusterings into a Final Clustering Using Object-Wise Similarity Graph
Name/Last Name of the Student	:	Ertunç ERDİL
Date of Thesis Defense	:	17 June, 2011

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assoc. Prof. Dr. Tunç BOZBURA
Acting Director

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Commitee Members:

Asst. Prof. Dr. Selim Necdet MİMAROĞLU (Supervisor) :

Assoc. Prof. Dr. Yücel SAYGIN :

Asst. Prof. Dr. Tevfik Metin SEZGİN :

ACKNOWLEDGEMENTS

This thesis is dedicated to my family who has given me much needed support and believed in me when times were rough and tough.

I would like to thank my advisor Dr. Selim Mimaroglu for his priceless help and guidance. This thesis would not have been possible unless the constant support and encouragement of him. It was an honor to work with him. His contributions will be very useful for my future academic research career.

I thank my jury members, Dr. Yücel Saygın and Dr. Tevfik Metin Sezgin, for their helpful suggestions, constructive criticisms, and time. I also thank Dr. Taskin Kocak and all other faculty members of Bahçeşehir University Computer Engineering Department for their great support during my master studies and assistantship.

I am grateful to the researchers in my research group, M. Emin Akşehirli and Murat Yağcı. It was a privilege to work with them. I would also thank all of my great colleagues, Ceyhun Can Ülker, Jbid Arsenyan, Erdem Erzurum, Özgür Ateş, Bengi Aygün, Ali Karaali, Oğuz Mustapaşa, and Güneş Akşehirli, who make the days at Bahçeşehir University unforgettable.

17 June 2011 Ertunç ERDİL

ABSTRACT

AGGREGATING ADVANTAGES OF A SET OF CLUSTERINGS INTO A FINAL CLUSTERING USING OBJECT-WISE SIMILARITY GRAPH

Erdil, Ertunç

Computer Engineering
Supervisor: Asst. Prof. Dr. Selim Necdet MİMAROĞLU

June 2011, 63 Pages

Clustering is the process of grouping objects that are similar, where similarity between objects is usually measured by a distance metric. Clustering is a hard problem since the natural grouping of a data set is unknown. Clustering aims to divide a data set into meaningful groups where each group formed by a clustering method is referred as a cluster. Clustering is a useful starting point for different purposes such as data understanding and summarization. In the literature, there are numerous applications of clustering ranging from biology to economics.

Clustering has a long and rich history in a variety of scientific fields. The main contributing research areas to clustering methodology are Machine Learning, Data Mining, and Pattern Recognition. Each clustering technique possess some advantages and disadvantages. Some clustering algorithms may even require input parameters which strongly affect the outcome. Some clustering techniques make some assumptions about the properties of the data sets and good quality clusterings are obtained, when the assumption holds. Distance metric also plays an important role in the process of producing a clustering. Especially in high dimensional data sets, it is hard to identify similarity or distance between objects. In most cases, it is not possible to choose the best distance metric, the best clustering method, and the best input parameter values for an input data set. Therefore, multiple clusterings can be obtained on a data set. And, multiple clusterings can be combined into a new and better quality final clustering.

In this thesis, we propose a graph based combining multiple clusterings algorithm that is scalable, robust, and intuitive. Combining multiple clusterings requires reusing preexisting knowledge and producing a novel final clustering having better overall quality. Our new algorithm, COMUSA, works on an object-wise weighted similarity graph which is constructed by using the evidence accumulated from multiple input clusterings. COMUSA offers good quality final clusterings by working at object level in a short amount of time. Extensive experimental evaluations on some very challenging real, synthetically

generated and gene expression data sets from a diverse set of domains establish the usefulness of our methods in terms of both quality and execution time.

Keywords: Unsupervised Learning, Cluster Ensemble, Data Mining, Machine Learning and Pattern Recognition

ÖZET

BİR KÜMELENMELER KÜMESİNİN AVANTAJLARINI NESNELER ARASI BENZERLİK ÇİZGESİ KULLANARAK BİR SONUÇ KÜMELENMESİNDE BİRLEŞTİRMEK

Erdil, Ertunç

Bilgisayar Mühendisliği
Tez Danışmanı: Yrd. Doç. Dr. Selim Necdet MİMAROĞLU

Haziran 2011, 63 Sayfa

Kümelenme benzer nesnelerin gruplanması sürecidir, objeler arası benzerlik genellikle bir uzaklık ölçütü ile ölçülür. Kümelenme, veri kümesinin gerçek gruplanması bilinmediği için zor bir problemdir. Kümelenme, verileri anlamlı gruplara bölmeyi amaçlar ve bir kümelenme metoduyla oluşturulmuş grup küme olarak adlandırılır. Kümelenme, verilerin anlaşılması ve özetlenmesi gibi farklı amaçlar için yararlı bir başlangıç noktasıdır. Literatürde kümelenme, biyolojiden ekonomiye kadar çeşitli uygulamalara sahiptir.

Kümelenme, çeşitli bilimsel alanlarda uzun ve zengin bir geçmişe sahiptir. Kümelenme metodolojisine katkıda bulunan temel alanlar Makine Öğrenmesi, Veri Madenciliği ve Örüntü Tanımadır. Herbir kümelenme tekniği bazı avantajlar ve dezavantajlar sergiler. Bazı kümelenme algoritmaları sonucu fazlasıyla etkileyecek girdi parametrelerine bile ihtiyaç duyabilirler. Bazı kümeleme teknikleri veri kümesinin özellikleri ile ilgili kabullenmeler yapabilir ve iyi kalitede bir kümelenme yalnızca bu kabullenmeler sağlandığında beklenebilir. Uzaklık ölçütü de kümeleme oluşturma sürecinde önemli bir rol oynar. Özellikle yüksek boyutlu veri kümelerinde nesneler arası benzerliği veya uzaklığı tanımlamak zordur. Bir çok durumda bir girdi veri kümesi için, en iyi uzaklık ölçütünü, en iyi kümeleme metodunu ve en iyi girdi argümanlarını seçmek mümkün değildir. Bu yüzden, bir veri kümesi için çoklu kümelemeler elde edilebilir. Ve, çoklu kümelemeler yeni ve daha iyi kaliteye sahip bir sonuç kümelemesinde birleştirilebilir.

Bu tezde, çoklu kümelemelerin birleştirilmesi için çizge tabanlı, ölçeklenebilir, güçlü ve sezgisel bir algoritma öneriyoruz. Çoklu kümelemelerin birleştirilmesi, önceki bilgilerin tekrar kullanılmasını ve daha iyi kaliteye sahip yeni bir sonuç kümelemesi oluşturulmasını gerektirir. Yeni algoritmamız, COMUSA, nesnelerden oluşan, ağırlıklı ve girdi kümelen-

melerindeki kanıt biriktirilerek oluşturulmuş bir benzerlik çizgesi üzerinde çalışır. COMUSA nesneler seviyesinde çalışarak, kısa bir sürede iyi kaliteye sahip sonuç kümelemesi oluşturmayı önerir. Çok çeşitli alanlardan alınmış gerçek, sanal olarak üretilmis ve gen ifade eden zorlayıcı veri kümeleri üzerindeki geniş deneysel sonuçlar metodumuzun hem kalite hem de çalışma zamanı olarak kullanışlı olduğunu gösterir.

Anahtar Kelimeler: Denetlenmemiş Öğrenme, Çoklu Kümelenmelerin Birleştirilmesi, Veri Madenciliği, Makine Öğrenmesi ve Örüntü Tanıma

TABLE OF CONTENTS

LIST OF TABLES ..... **ix**
LIST OF FIGURES ..... **x**
LIST OF ABBREVIATIONS ..... **xi**
LIST OF SYMBOLS ..... **xii**

1. INTRODUCTION **1**

1.1 CLUSTERING **1**
1.2 CLUSTERING METHODS **3**
1.2.1 Partitioning Methods **3**
1.2.2 Hierarchical Methods **4**
1.2.3 Density-Based Methods **6**
1.2.4 Other Pioneering Clustering Methods **8**
1.3 CLUSTER EVALUATION **9**
1.3.1 Supervised Methods **10**
1.3.2 Unsupervised Methods **11**
1.3.3 Relative Methods **12**
1.4 COMBINING MULTIPLE CLUSTERINGS **12**
1.4.1 Formal Definition of the Problem **13**
1.4.2 Evaluating the Quality of Final Clustering **14**
1.4.3 Related Work **18**
1.5 THESIS OVERVIEW **23**

2. COMUSA: COMBINING MULTIPLE CLUSTERINGS USING SIMILARITY GRAPH **24**

2.1 COMUSA **24**
2.1.1 Relaxation **33**

3. DISCUSSION AND EXPERIMENTAL RESULTS **34**

3.1 DISCUSSION OF COMUSA **34**
3.2 EXPERIMENTAL EVALUATIONS **35**
3.2.1 Generating Cluster Ensembles **35**
3.2.2 Test Results of COMUSA on Real, Synthetically Generated, and Gene Expression Data Sets **36**

4. CONCLUSION **39**
REFERENCES **41**
APPENDICES **50**
APPENDIX A FIGURES **51**
APPENDIX B TABLES **55**

LIST OF TABLES

Table 1.1 : **Contingency table** **17**

Table 2.1 : **attachment values of figure 2.2 in decreasing order** **28**

Table B.1 : **Properties of multiple clusterings on real and synthetically generated data sets** **55**

Table B.2 : **Properties of gene expression data sets** **56**

Table B.3 : **Properties of multiple clusterings on gene expression data sets** **57**

Table B.4 : **COMUSA on 1-spiral data set** **58**

Table B.5 : **Cluster validity results on 2-spiral, 2-half rings, 2-Curve, and Syn5K data sets** **58**

Table B.6 : **Cluster validity results on 2D2K and 8D5K** **59**

Table B.7 : **Cluster validity results on Iris, Glass, Breast Cancer, and Image Segmentation data sets** **60**

Table B.8 : **Cluster validity results on gene expression data sets** **61**

Table B.9 : **Execution time results (ms)** **62**

Table B.10 : **Number of clusters** **63**

LIST OF FIGURES

Figure 1.1 : Different clusterings of a data set **2**

Figure 1.2 : The process of hierarchical clustering algorithms on a sample data set Source: Han and Kamber (2006) **5**

Figure 1.3 : Similarity criterias between clusters **6**

Figure 1.4 : Center based density **7**

Figure 1.5 : Labeling with parameters ε and $MinPts = 7$ **7**

Figure 1.6 : Graph-based representations of cluster cohesion and separation ... **11**

Figure 1.7 : Combining multiple clusterings **12**

Figure 1.8 : Binary representation of multiple clusterings, Π **14**

Figure 1.9 : Co-association matrix, SM, of figure 1.8 ..... **21**

Figure 2.1 : df and sw **25**

Figure 2.2 : Similarity graph of figure 1.9 ..... **26**

Figure 2.3 : Generating final clustering using COMUSA on figure 2.2 ..... **30**

Figure 2.4 : COMUSA on a data set **31**

Figure 2.5 : COMUSA on another data set **32**

Figure 2.6 : A partial similarity graph **34**

Figure A.1 : 1-spiral data set and a clustering **51**

Figure A.2 : 2-spiral data set **52**

Figure A.3 : 2-curve data set **52**

Figure A.4 : 2-half rings data set **53**

Figure A.5 : Partitions of 2-half rings data set generated with k-means **54**

LIST OF ABBREVIATIONS

Adjusted Rand Index	:	ARI
Agglomerative Nesting	:	AGNES
Average Normalized Mutual Information	:	ANMI
Bipartite Merger	:	BM
Clustering in Quest	:	CLIQUE
Cluster-Based Similarity Partitioning Algorithm	:	CSPA
Combining Multiple Clsuterings Using Similarity Graph	:	COMUSA
Density-Based Clustering	:	DENCLUE
Density-Based Spatial Clustering of Applications with Noise	:	DBSCAN
Divisive Analysis	:	DIANA
Evidence Accumulation	:	EAC
Hyper-Graph Partitioning Algorithm	:	HGPA
Inter Cluster Similarity	:	ECS
Intra Cluster Similarity	:	ICS
Meta-Clustering Algorithm	:	MCLA
Metis Merger	:	MM
Minimum Spanning Tree	:	MST
Normalized Mutual Information	:	NMI
Rand Index	:	RI
Selective Spectral Clustering Algorithm	:	SELSCE
Sum of Squared Error	:	SSE
The Link-Based Cluster Ensemble	:	LCE

LIST OF SYMBOLS

Attachment Value	:	attachment
Cluster Centroid that Represent C_i	:	c_i
Clustering of D	:	$\pi(D)$
Cluster Validity Measure	:	ϕ
Consensus Function	:	cns
Data Object	:	d
Data Set	:	D
Degree of Freedom	:	df
Entropy of Cluster C_i	:	$e(C_i)$
Epsilon	:	ε
Final Clustering	:	$\pi^{\star}(D)$
i^{th} Cluster of $\pi(D)$	:	C_i
i^{th} Clustering of $\Pi(D)$	:	$\pi_i(D)$
j^{th} Cluster of i^{th} Clustering	:	C_{ij}
Minimum Number of Points	:	$MinPts$
Multiple Clusterings of D	:	$\Pi(D)$
Mutual Information Between clusterings $\pi_i(D)$ and $\pi_j(D)$	:	$I(\pi_i(D), \pi_j(D))$
Number of Clusters	:	k
Relaxation Rate	:	r
Silhoutte Coefficient of i^{th} object	:	s_i
Similarity Matrix	:	SM
Similarity Graph	:	SG
Sum of Weights	:	sw
Total Entropy of Clustering $\pi(D)$	:	$e(\pi(D))$
True Class Membership of D	:	$\pi^{T}(D)$
Weight Function	:	weight

1. INTRODUCTION

This chapter provides information on clustering, cluster evaluation, and combining multiple clusterings which constitutes the basis of this thesis.

1.1 CLUSTERING

Clustering, which is also known as $unsupervised\ classification$, aims to group similar data objects into clusters. Therefore, it is expected that objects in the same cluster are similar to each other and they are dissimilar to the other objects in other clusters. Similarity is evaluated using a distance metric based on the attribute values describing the objects.

Data clustering is a major researh topic in a variety of disciplines. Contributing areas to this topic include data mining, machine learning, pattern recognition, statistics, mathematics, and bioinformatics. Increasing amount of data yields the need of cluster analysis. There have been many applications of cluster analysis to real life problems. Biologists apply clustering to create taxonomy of all living things and to analyze huge amounts of genetic data to find groups of genes having similar functions (Tan et al. 2006). Clustering is also a crucial part of medicine discovery process (MacCuish and Maccuish 2010). Image segmentation aims to represent a digital image in terms of clusters of pixels (Forsyth and Ponce 2002). Clustering texts and documents is one of the most important applications of clustering in Text Mining (Feldman and Sanger 2007).

Clustering is an ill-defined problem because the correct clustering of a data set is not obvious in most cases. Data sets may have varying properties and the properties of a data set is generally unknown. So, it is hard knowing the most appropriate clustering algorithm to apply on the data set. Everitt (1974) defines the problem as connected regions of a multi-dimensional space containing a relatively high density of points, seperated from other such regions by a region containing a relatively low density of points. This definition assumes that data objects to be clustered can be represented as points in space and clusters may be identified with the eye. However, it is still not very clear how we identify the clusters due to the fact that clusters may be perceived differently in the human mind. Let us consider the data set in Figure 1.1a which is plotted in two dimensional space. The question is “How many clusters are there in this data set?”. When we look at the

data set, we may identify three clusters as shown with different colors in Figure 1.1b. On the other hand, the clustering in Figure 1.1c which has eight clusters also can be perceived by the human mind. Which one is the correct clustering? The answer depends on the similarity threshold that we observe the data set. At a higher level of similarity threshold, it is expected to perceive a clustering like in Figure 1.1b, but at a lower level similarity threshold data objects with higher similarity locate in the same cluster and more clusters are formed as illustrated in Figure 1.1c. Thus, one of the most crucial problems in clustering is to specify an appropriate similarity metric. This makes clustering problem even harder for high dimensional data sets (Bellman 2003).

Yet, determining natural number of clusters also poses a challenging issue for clustering methods. Most of the existing methods need number of clusters as a user specified input parameter. The parameter may enable some methods to produce better clusterings when only it is provided correctly. For example, in Figure 1.1a, if number of clusters were specified as 3, the clustering in Figure 1.1b would have been inferred easily.

In the literature, clustering problem is discussed extensively. Some detailed information about clustering can be found in Tan et al. (2006), Han and Kamber (2006), Jain and Dubes (1988), Alpaydin (2004), Jain (2010), Bishop (2006).

Figure 1.1a Sample data set

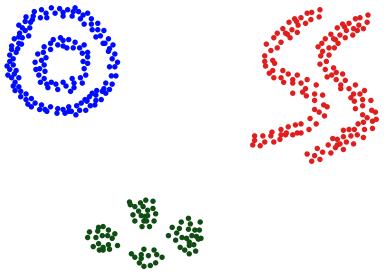

Figure 1.1b A clustering on data set

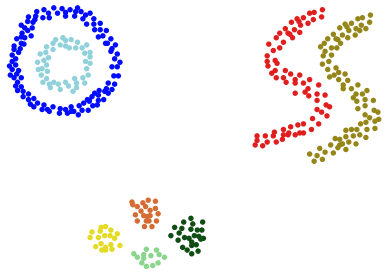

Figure 1.1c Another clustering on data set

Figure 1.1: Different clusterings of a data set

1.2 CLUSTERING METHODS

As we mentioned in the previous section, clustering is a hard to accomplish task because of its ill-defined nature. There are thousands of clustering algorithms in the literature; each makes some assumptions about the underlying data set. Good quality clusterings can be expected when the assumptions hold. Since the characteristic of the data set is generally unknown, more than often, the assumption do not hold, which in turn means bad quality clusterings will be obtained. On this basis, Jain (2010) emphasizes that cluster analysis is an exploratory tool; the output of clustering algorithms only suggest hypotheses.

Categorization of clustering algorithms is difficult since these categories may overlap (Han and Kamber 2006). In this section, we provide a non-exhaustive survey of pioneering and state-of-the-art clustering methods.

1.2.1 Partitioning Methods

Partitioning clustering methods divides a data set into non-overlapping clusters such that each data object is assigned into only one cluster. Number of clusters (partition) is specified with a parameter by the user usually. k-means (first introduced by Lloyd (1982)) is a well-known partitioning method and is commonly used.

k-means

The k-means clustering algorithm takes the input parameter, k, and partitions the data set into k clusters such that the data objects in the same cluster are most similar to the cluster centroids.

The basic k-means algorithm is given in Algorithm 1 and it proceeds as follows. First, k data objects are randomly selected which represent the cluster centroids initially. Each data object is assigned into the most similar cluster, and similarity is measured by using the distance between the data objects and their corresponding centroids. Then, cluster centroids are recomputed. These steps are repeated until the cluster centroids do not change. Thus, k clusters are taken their final form. Clustering can also be formulated

Algorithm 1: k-means Algorithm

Input: D: Data Set, k: Number of Clusters
Output: k Clusters

1 Select k points randomly as initial centroids;
2 **repeat**
3 From k clusters by assigning each point to its closest centroid;
4 Recompute the centroid of each cluster;
5 **until** *Centroids do not change* ;

as an optimization problem with an objective function, and algorithm iterates until the function converges. Generally, sum of the squared error is used and defined as follows:

$$SSE = \sum_{i=1}^{k} \sum_{d \in C_i} distance(c_i, d)^2 \tag{1.1}$$

where d is a data object, C_i is the i^{th} cluster and c_i is the centroid that represents C_i.

Although the basic k-means algorithm is used extensively, there are some issues and ties that may need to be solved. One of that obtaining the singleton clusters when no points are allocated to a cluster during the assignment step. Yet, k-means is affected by the outliers because outliers change the centroid of the clusters considerably.

1.2.2 Hierarchical Methods

Hierarchical clustering algorithms can be divided into two categories: agglomerative and divisive. Hierarchical methods construct nested clusters that can be represented with a tree, called **Dendogram**. A meaningful clustering can be obtained by cutting the Dendongram at a certain level. Divisive clustering algorithms start with one big cluster containing all data objects and splits it until all clusters become singleton. Just the opposite agglomerative clustering algorithms start by placing each object in its own cluster and merges the most similar clusters iteratively. AGNES and DIANA, which are visualized in Figure 1.2, are well-known agglomerative and divisive clustering algorithms, respectively.

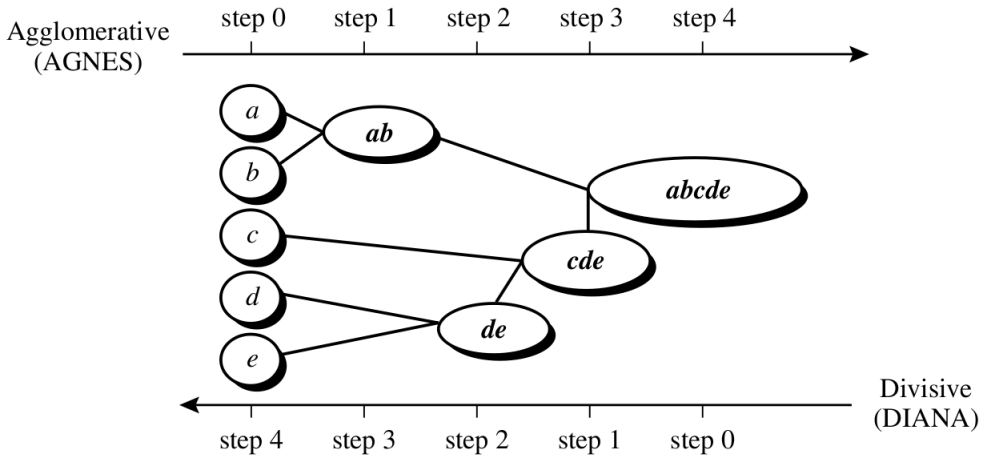

Figure 1.2: The process of hierarchical clustering algorithms on a sample data set Source: Han and Kamber (2006)

Algorithm 2: Basic Agglomerative Clustering Algorithm

Input: D: Data Set
Output: Dendogram

```
1 Compute the similarity matrix ;
2 Initialize the dendogram such that each data object in its own cluster ;
3 repeat
4     Merge two most similar clusters ;
5     Update similarity matrix to reflect the similarity between new cluster and the other
      clusters;
6     Update Dendogram;
7 until All clusters become merged ;
```

Most of the existing hierarchical clustering algorithms in the literature are agglomerative. Basic agglomerative clustering algorithm is given in Algorithm 2. As we mentioned earlier, the algorithm starts with singleton clusters and merges the most similar cluster in a greedy manner until all data objects are placed in one cluster. The key step of Algorithm 2 is the determination of the similarity criteria between two clusters. **Single link** defines similarity between clusters as the highest similarity between two data objects from different clusters. It is good at handling arbitrary shape data sets, but is sensitive to noise and outliers. **Complete link** between two clusters is defined as the lowest similarity between data objects that are in different clusters. Complete link works well on globular shape data sets and less sensitive to noise and outliers. Alternatively, **group average** is the average of the pairwise similarities of all data objects in different clusters. Figure 1.3

illustrates the three similarity criterias: single link, complete link, and group average. Another technique, **Ward's method** (Ward 1963), assumes that a cluster is represented by its centroid and attempts to minimize SSE after merging two clusters.

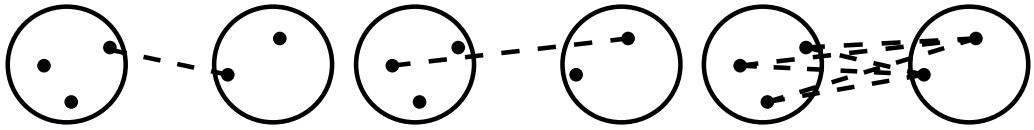

Figure 1.3a Single link **Figure 1.3b** Complete link **Figure 1.3c** Group average

Figure 1.3: Similarity criterias between clusters

1.2.3 Density-Based Methods

Density-based methods assume cluster as a high density region that is seperated from other low density regions (Tan et al. 2006). DBSCAN (Density Based Spatial Clustering of Applications with Noise) is a simple and effective density-based clustering algorithm that is designed for identifying arbitrary shape clusters and noise (Ester et al. 1996).

The main operation of density-based methods is defining density, which is not trivial. There are several distinct approaches proposed for this purpose. Center-based density approach is the basis for the DBSCAN algorithm. It classifies a data object as core point, border point or noise with respect to two user specified parameters: ε and $MinPts$. An object is a core point if the number of objects with its ε radius is at least $MinPts$. A border point is not a core point, but is located within the ε radius of a core point. A noise point is an object that is neither a core point nor a border point. In Figure 1.4, ε radius of data object p contains 7 data objects. In Figure 1.5, p_1 is a core point, p_2 is a border point, and p_3 is noise with respect to ε and $MinPts = 7$.

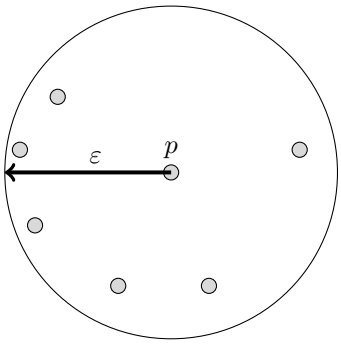

Figure 1.4: Center based density

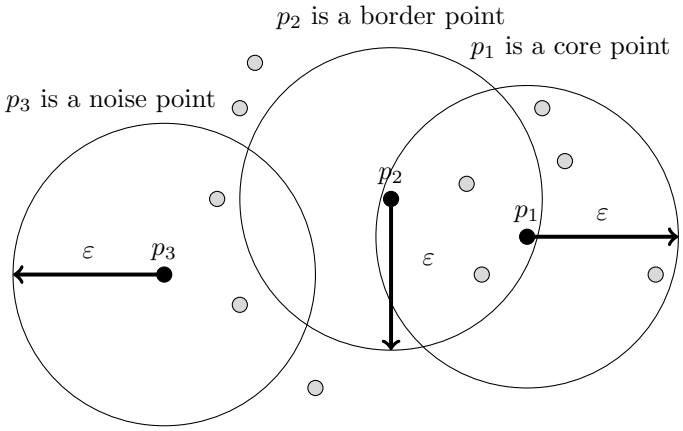

Figure 1.5: Labeling with parameters ε and $MinPts = 7$

Algorithm 3: DBSCAN Algorithm

Input: D: Data Set, ε, $MinPts$
Output: Clustering of D

1 Mark all data objects as core points, border points, or noise with respect to ε and $MinPts$;
2 Put an edge between core points that are in the ε neighborhood of each others;
3 Assign connected core points into a cluster;
4 Assign all of the border points within the ε neighborhood of a cluster into the same cluster;

DBSCAN is given in Algorithm 3. Initially, each data object is labeled as core, border, or noise point. Core points that are within the ε neighborhood of each other are assigned in the same cluster. Similarly, any border point that in the ε radius of a core point is put in the same cluster as core point. Noise points are eliminated.

There are, of course, some issues and shortcomings of DBSCAN like any other clustering method. Determining the input parameters, ε and $MinPts$, is one of the main problem in DBSCAN. Although there are rule of thumbs for determining these parameters, they

are not efficient for data sets with varying density. Therefore, DBSCAN does not provide good results on such data sets. Reducing execution time of DBSCAN is also very challenging. Zhou et al. (2000), Borah and Bhattacharyya (2004), Tsai and Sung (2010), and Mimaroglu and Aksehirli (2011) propose some improvements on DBSCAN in terms of execution time.

1.2.4 Other Pioneering Clustering Methods

The k-medoids algorithm is designed by Kaufman et al. (1990), to solve the noise sensitivity issue of k-means. It suggests to take real data objects as representative (medoid) of clusters instead of taking the mean value of objects in a cluster. The remaining data objects are clustered with the medoid where it is the most similar. The algorithm proceeds to minimize the sum of the dissimilarities within a cluster which requires analyzing all possible pairs of objects. CLARA (Kaufman et al. 1990) and CLARANS (Ng and Han 1994) are also partitioning methods and works based on the idea in k-medoids.

Grid-based clustering algorithms breaks the data space into grids and then forms dense grids as a cluster if the density is over a certain threshold. Therefore, they are considered as density-based according to some sources. CLIQUE (Clustering in Quest) (Gunopulos and Raghavan 1998) is a grid-based algorithm that provides an efficient approach to cluster high-dimensional data sets. Other examples for this type clustering are proposed in Hinneburg and Keim (1999), Schikuta and Erhart (1997), and Sheikholeslami et al. (1998). The main of grid-based methods are defining the size of grid cells and specifying a threshold for density. DENCLUE (Hinneburg and Keim 1998) is a kernel based scheme for density-based clustering. It computes the overall density using a mathematical function, called **influence function**, and clusters are formed by identifying the local maxima of the function.

Graph-based methods are also widely used for clustering purpose as well. In a graph-based method, data objects correspond to vertices of the graph and the similarity between two data objects are represented by a weighted edge. Minimum spanning tree (MST) clustering (Everitt et al. 2011) constructs a dissimilarity graph of data objects and constructs a minimum spanning tree of this graph. Then, it proceeds by breaking the edge having the largest dissimilarity at each iteration until singleton clusters remain. OPOSSUM, proposed by Strehl and Ghosh (2000), is designed to identify clusters of sparse and high dimensional data sets. It partitions the similarity graph using a graph parti-

tioning package, METIS (Karypis and Kumar 1998). Chameleon (Karypis et al. 1999) uses METIS package to partition the similarity graph as well. Unlike OPOSSUM, it then merges partitions obeying a similarity criteria. Although METIS package is designed for graph partitioning purpose, clustering can also be conducted using METIS.

Crisp partitions of well-seperated clusters can be found easily. However, in most cases, it is hard to assign data objects into a particular cluster. Fuzzy clustering methods gives a membership value to each data object and assign them into clusters with respect to their membership values utilizing the fuzzy theory (Lee 2005). In brief, a data object may belong to several different clusters with varying memberships. Fuzzy c-means (Höppner et al. 1999) is the most well-known fuzzy clustering method in the literature. In is undoubtedly true that, more than often, data sets are generated as an output of a statistical process. Therefore, it is not surprising to find a statistical model that fits on the data set. Mixture models work based on this assumption. Expectation-Maximization (EM) algorithm (Dempster et al. 1977) is widely used to find mixture model parameters using maximum likelihood principle.

Liu et al. (2009) proposes a clustering algorithm which represents a cluster by multiple prototype. A graph-based approach for clustering dense graphs is introduced in Moussiades and Vakali (2010). A method designed for document clustering purpose (Kalogeratos and Likas 2011) clusters data sets using synthetic cluster prototypes. A new mixture model for clustering high-dimensional micro array data is proposed in Baek and McLachlan (2011).

1.3 CLUSTER EVALUATION

Cluster evaluation, or cluster validation, is not a well-developed or widely used branch of cluster analysis because of its unsupervised nature. Nonetheless, there are many methods for cluster evaluation in the literature. In this section, we briefly mention the important features of existing methods for evaluating validity of clusters.

Each clustering algorithm perceives the notion of cluster from different angles. They all make some assumptions about the underlying data set. Good clusterings can be expected when the assumption holds. However, generally, assumptions about the data set do not hold, which in turn means bad clusterings are generated. It is very hard to select an appropriate clustering method, because the natural grouping is unknown. Many clus-

tering algorithms effect the result by taking input parameters, e.g. k-means, k-medoids, DBSCAN, etc. So, we have to evaluate the better clustering in some way.

Jain and Dubes (1988) groups cluster validation methods into three types as follows:

1.3.1 Supervised Methods

Supervised methods measure cluster validity by using external information. Often, this information is true class labels of the data set. Supervised methods are widely used to evaluate the performance of a classification model. Therefore, they also known as classification oriented measures.

Entropy is a well-known approach used in information theory which provides useful descriptions of long term behavior of random processes (Gray 2010). Given a data set D, a clustering $\pi(D) = \{C_1, C_2, \ldots, C_{|\pi(D)|}\}$, and true class memberships of D $\pi^T(D) = \{C_1^T, C_2^T, \ldots, C_{|\pi(D)|}^T\}$, class entropy of each cluster, C_j, is computed using the Formula 1.2.

$$e(C_j) = -\sum_{i=1}^{|\pi^T(D)|} \frac{|C_i^T \cap C_j|}{C_j} \log_2 \frac{|C_i^T \cap C_j|}{C_j} \tag{1.2}$$

Total entropy of the clustering $\pi(D)$ is computed as the sum of the entropies of each cluster weighted by the size of each cluster as shown in Formula 1.3.

$$e(\pi(D)) = \sum_{j=1}^{|\pi(D)|} \frac{|C_j|}{|D|} e(C_j) \tag{1.3}$$

Purity, Precision, Recall, and F-measure are the examples of other common supervised measures of the extent to which a cluster contains objects of a single class.

1.3.2 Unsupervised Methods

Unsupervised methods measures the validity of a clustering without making use of any external information, which is usually measured by using the data set itself or the similarity matrix. Unsupervised methods are often divided into two categories: cluster cohesion and cluster separation. **Cluster cohesion** determines the compactness of a single cluster. **Cluster separation**, conversely, measures the distinctness or how isolated a cluster is from other clusters. Cluster cohesion and separation approaches can be graph-based or prototype-based. A hybrid method known as the **Silhouette Coefficient**, combines both

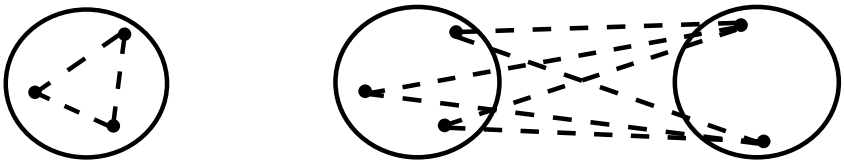

Figure 1.6a Cluster cohesion

Figure 1.6b Cluster separation

Figure 1.6: Graph-based representations of cluster cohesion and separation

cohesion and separation. Silhouette coefficient is computed for a data object with three steps as follows (Tan et al. 2006):

- Compute the average distance of the i^{th} data object to all other objects in its clusters, and call this value a_i.
- Compute the average distances of the i^{th} data object to any cluster not containing the object. Find the minimum average value among all clusters and call this value b_i.
- The silhouette coefficient for the i^{th} object is $s_i = (b_i - a_i)/max(a_i, b_i)$.

1.3.3 Relative Methods

Relative methods compare two different clusterings or clusters instead of measuring their validity. For this purpose, both supervised and unsupervised evaluation measures are utilized. Thus, relative methods can be considered as specific type of cluster evaluation measures, not as a seperate type of measure.

1.4 COMBINING MULTIPLE CLUSTERINGS

Combining multiple clusterings into a final clustering having better overall quality is a growing research topic in machine learning, pattern recognition, and data mining. The problem is also known as *cluster ensemble*, *cluster fusion*, and *consensus clustering* in the literature. Multiple clusterings can be obtained by using distinct clustering methods, or by providing varying input parameters to a clustering method. In some cases, human experts can also produce clusterings. Therefore, we can have multiple clusterings on an input data set, and utilize this valuable information for obtaining a final clustering. The

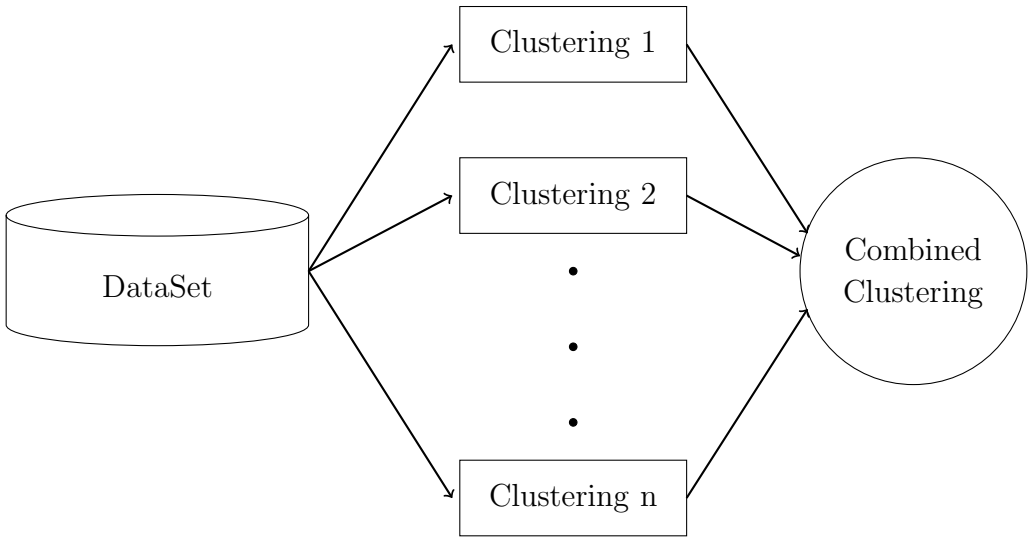

Figure 1.7: Combining multiple clusterings

schema that represents combining multiple clustering process is shown in Figure 1.7. The goal of combining multiple clusterings is to produce a new final clustering by aggregating the advantages and reducing the disadvantages.

Many consensus functions have been proposed in the literature. In Filkov and Skiena (2004) and Cristofor and Simovici (2002), consensus functions based on median partition approach have been proposed. These approaches search differences between the clusterings by working on a coarser level. At another direction, in Strehl and Ghosh (2003), consensus functions based on hypergraphs have been proposed. In this technique, a hyperedge represents a cluster, and a hypergraph represents a clustering. An evolutionary and kernel function based algorithms are proposed in Mohammadi et al. (2008) and (Vega-Pons et al. 2010), respectively. An information-theoretical framework is capable to identify clusters with arbitrary shapes (Ana and Jain 2003).

1.4.1 Formal Definition of the Problem

Let D be a data set. A clustering (partition) of D, $\pi(D)$, can be stated as follows:

$$\pi(D) = \{C_1, C_2, \ldots, C_{|\pi(D)|}\},$$

where C_i is a cluster (block) of $\pi(D)$, $1 \leq i \leq |\pi(D)|$, and

$$D = \bigcup_{i=1}^{|\pi(D)|} C_i$$

Note that we have a partial clustering (i.e. not complete) when $\bigcup_{i=1}^{|\pi(D)|} C_i \subset D$. Given a set of clusterings $\Pi(D) = \{\pi_1(D), \pi_2(D), \ldots, \pi_{|\Pi(D)|}(D)\}$, the problem of combining multiple clusterings is defined as finding a new clustering $\pi^\star(D) = \{C_1^\star, C_2^\star, \ldots, C_{|\pi^\star(D)|}^\star\}$ by using the information provided by $\Pi(D)$. This is achieved by using a consensus function $cns(\Pi(D)) = \pi^\star(D)$ such that

$$\forall i(\phi(\pi^\star(D)) \geq \phi(\pi_i(D))), 1 \leq i \leq |\Pi(D)| \tag{1.4}$$

where function ϕ is a cluster validity measure. Exhaustively searching all the possible clusterings for finding *the best* clustering is not an option, since there are $\frac{1}{k!}\sum_{l=1}^{k}\binom{k}{l}(-1)^{k-l}l^n$ possible clusterings where k is the number of final clusters and n is the number of objects (Strehl and Ghosh 2003). Three clusterings on a data set are

Clustering	**Cluster**	$\mathbf{d_1}$	$\mathbf{d_2}$	$\mathbf{d_3}$	$\mathbf{d_4}$	$\mathbf{d_5}$	$\mathbf{d_6}$	$\mathbf{d_7}$	$\mathbf{d_8}$
$\pi_1(\mathbf{D})$	$\mathbf{C_{11}}$	1	0	1	0	0	1	0	0
	$\mathbf{C_{12}}$	0	0	0	1	1	0	0	0
	$\mathbf{C_{13}}$	0	1	0	0	0	0	1	1
$\pi_2(\mathbf{D})$	$\mathbf{C_{21}}$	1	1	0	1	0	0	0	0
	$\mathbf{C_{22}}$	0	0	0	0	0	0	0	1
	$\mathbf{C_{23}}$	0	0	0	0	1	0	1	0
	$\mathbf{C_{24}}$	0	0	1	0	0	1	0	0
$\pi_3(\mathbf{D})$	$\mathbf{C_{31}}$	0	0	1	0	0	1	0	0
	$\mathbf{C_{32}}$	1	1	0	1	0	0	0	1
	$\mathbf{C_{33}}$	0	0	0	0	1	0	1	0

Figure 1.8: Binary representation of multiple clusterings, Π

presented in Figure 1.8. In this form, each cluster is represented by its characteristic bit vector which is as long as the size of the data set, $|D|$. For example, $\mathbf{C_{11}}$ cluster has d_1, d_3, and d_6 objects as shown below.

$\mathbf{C_{11}}$

1	0	1	0	0	1	0	0

1.4.2 Evaluating the Quality of Final Clustering

The quality of final clustering can be evaluated using several cluster objective measures such as inter-cluster similarity (ECS), intra-cluster similarity (ICS) (Mimaroglu and Yagci 2009), rand index (RI) (Rand 1971), adjusted rand index (ARI) (Hubert and Arabie 1985), normalized mutual information (NMI) and average normalized mutual information (ANMI) (Strehl and Ghosh 2003),and jaccard index (Denud and Gunoche 2006). Inter and intra cluster similarities, normalized and average normalized mutual information, and adjusted rand index are explained in this section.

Normalized and Average Normalized Mutual Information

NMI is a cluster validity measure that compares two clusterings. Let $\pi_i(D)$ and $\pi_j(D)$ be two clusterings of a data set, D. $I(\pi_i(D), \pi_j(D))$ is defined as the mutual information between $\pi_i(D)$ and $\pi_j(D)$, and $e(\pi_i(D))$ states the entropy of $\pi_i(D)$. Yet, the NMI is defined as in Equation 1.5.

$$NMI(\pi_i(D), \pi_j(D)) = \frac{I(\pi_i(D), \pi_j(D))}{\sqrt{e(\pi_i(D))e(\pi_j(D))}} \tag{1.5}$$

Note that NMI can be used as a supervised cluster validity measure if the natural class labels are provided to NMI as a clustering. NMI takes the value 1 when the perfect matching is obtained, which is desired.

Average normalized mutual information is an unsupervised cluster validity measure. ANMI takes the average of all NMI values between the final clustering $\pi^\star(D)$ and each clustering in multiple clusterings $\Pi(D)$ (see Equation 1.6).

$$ANMI(\pi^\star(D), \Pi(D)) = \frac{1}{|\Pi(D)|} \sum_{\pi_i(D) \in \Pi(D)} NMI(\pi^\star(D), \Pi(D)) \tag{1.6}$$

Intra and Inter Cluster Similarities

Intra-cluster similarity measures the inner similarity of a cluster, where large values are preferred. For a clustering, $\pi(D) = \{C_1, C_2, \ldots, C_{|\pi(D)|}\}$, intra-cluster similarity is defined as follows:

$$ICS(\pi(D)) = \sum_{i=1}^{|\pi(D)|} \frac{1}{|C_i|^2} \sum_{d,d' \in C_i} sim(d, d') \tag{1.7}$$

In (1.7) $sim(d, d')$ is the similarity of the objects d and d'. When working on multiple clusterings, evidence accumulated in $\Pi(D)$ is used as a similarity measure as mentioned earlier. Intra-cluster similarity of a final clustering $\pi^\star(D)$ with respect to multiple clusterings $\Pi(D)$ is shown in (1.8) (Mimaroglu and Yagci 2009).

$$ICS_\Pi(\pi^\star(D)) = \sum_{k=1}^{|\pi^\star(D)|} \frac{1}{|C_k^\star|^2} \sum_{i=1}^{|\Pi|} \sum_{j=1}^{|\pi_i|} \binom{|C_k^\star \wedge C_{ij}|}{2} \tag{1.8}$$

Inter-cluster similarity of a cluster $\pi(D)$ is defined as follows:

$$ECS(\pi(D)) = \sum_{i=1}^{|\pi(D)|} \sum_{j=i+1}^{|\pi(D)|} \frac{1}{|C_i||C_j|} \sum_{d \in C_i, d' \in C_j} sim(d, d') \tag{1.9}$$

Low values of (1.9) indicate that $\pi(D)$ has isolated clusters, which is preferred. Inter-cluster similarity of a final clustering $\pi^\star(D)$ with respect to multiple clusterings $\Pi(D)$ is shown in Formula 1.10 (Mimaroglu and Yagci 2009).

$$\begin{aligned} ECS_\Pi(\pi^\star(D)) = & \sum_{k=1}^{|\pi^\star(D)|} \sum_{l=k+1}^{|\pi^\star(D)|} \frac{1}{|C_k^\star||C_l^\star|} \sum_{i=1}^{|\Pi|} \sum_{j=1}^{|\pi_i|} \binom{|(C_k^\star \vee C_l^\star) \wedge C_{ij}|}{2} \\ & - \binom{|C_k^\star \wedge C_{ij}|}{2} - \binom{|C_l^\star \wedge C_{ij}|}{2} \end{aligned} \tag{1.10}$$

Inter-cluster similarity and intra-cluster similarity can be combined to form a clustering validity function as shown in Equation 1.11.

$$\phi(\pi^{\star}(D)) = k_1.ICS(\pi^{\star}(D)) + k_2.ECS(\pi^{\star}(D)) \tag{1.11}$$

,where $k_1 > 0$ and $k_2 < 0$. In our tests, we use Formula 1.11 as an unsupervised cluster evaluation technique and we refer to it as ICS+ECS.

Adjusted Rand Index(ARI)

Adjusted Rand Index (ARI) can be used for both checking the validity of a clustering algorithm or a combining multiple clusterings algorithm. ARI measures the extent to which the discovered clustering structure matches some external criteria, i.e. class labels. Given a data set $D = \{d_1, \ldots, d_n\}$, suppose $U = \{u_1, \ldots, u_r\}$ represents classes, and $V = \{v_1, \ldots, v_p\}$ represents a clusterings of the D.

$$\bigcup_{i=1}^{r} u_i = D = \bigcup_{j=1}^{p} v_j \tag{1.12}$$

and $u_i \cap u_j = \emptyset$ for $1 \leq i, j \leq r$ and $i \neq j$. Also, $v_i \cap v_j = \emptyset$ for $1 \leq i, j \leq p$ and $i \neq j$.

Table 1.1: Contingency table

Class / Cluster	$\mathbf{v_1}$	$\mathbf{v_2}$	$\dots$	$\mathbf{v_p}$	**Sums**
$\mathbf{u_1}$	n_{11}	n_{12}	$\dots$	n_{1p}	$n_{1.}$
$\mathbf{u_2}$	n_{21}	n_{22}	$\dots$	n_{2p}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	
$\mathbf{u_r}$	n_{r1}	n_{r2}	$\dots$	n_{rp}	$n_{r.}$
Sums	$n_{.1}$	$n_{.2}$		$n_{.p}$	$n_{..} = n$

In Table 1.1, $n_{ij} = |u_i \cap v_j|, n_{i.} = \sum_{j=1}^{p} n_{ij}$, and $n_{.j} = \sum_{i=1}^{r} n_{ij}$

ARI can be formulated as follows:

$$\frac{\sum_{i,j} \binom{n_{ij}}{2} - \left(\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2}\right) / \binom{n}{2}}{\frac{1}{2}\left(\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}\right) - \left(\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2}\right) / \binom{n}{2}} \tag{1.13}$$

ARI takes maximum value at 1, which indicates perfect match to the external criteria.

1.4.3 Related Work

In this section some of the important methods for combining multiple clusterings are explained.

The link-based cluster ensemble (LCE)

The link-based cluster ensemble (LCE), which is presented in Iam-On et al. (2010), starts with a bipartite membership graph of objects and clusters and builds up a dense graph with implied similarities between every cluster and every object. LCE produces a final clustering on this structure by spectral graph partitioning technique. LCE, which is designed to work on gene expression data sets, produces good results on biological and non-biological data sets.

Cluster-Based Similarity Partitioning Algorithm (CSPA)

CSPA, which is introduced in Strehl and Ghosh (2003), is based on co-association matrix, and METIS (Karypis and Kumar 1998). CSPA is shown in Algorithm 4.

Algorithm 4: Cluster-Based Similarity Partitioning Algorithm **CSPA**

Input: $\Pi(D)$: Multiple Clusterings of a Data Set D,
k: Number of Clusters In the Final Clustering
Output: $\pi^{\star}(D)$: Final Clustering

1 Compute co-association matrix, SM, using $\Pi(D)$;
`// Partition the Similarity Graph of` SM `into` k `components using METIS`
2 $\pi^{\star}(D) = \mathbf{METIS}(SM, k)$;
3 **return** $\pi^{\star}(D)$;

Hyper-Graph Partitioning Algorithm (HGPA)

HGPA is introduced in Strehl and Ghosh (2003) as well: Multiple clusterings construct a hyper-graph where each object is a vertex, and each cluster is an hyper-edge. Main idea is to have k unconnected components of the hyper-graph by using HMETIS (Karypis et al. 1997). Combining multiple clusterings problem is formulated as partitioning the hyper-graph by cutting a minimal number of hyper-edges. A set of hyper-edges are removed and k unconnected components are obtained, which provides the final clustering. HGPA is shown in Algorithm 5.

Meta-Clustering Algorithm (MCLA)

In Meta-Clustering Algorithm (MCLA) (Strehl and Ghosh 2003), which is shown in Algorithm 6, a meta-cluster is defined as a cluster of clusters. MCLA constructs a meta-

Algorithm 5: Hyper-Graph Partitioning Algorithm **(HGPA)**

Input: $\Pi(D)$: Multiple Clusterings of a Data Set D,
k: Number of Clusters In the Final Clustering
Output: $\pi^{\star}(D)$: Final Clustering

Construct a hyper-graph, HG, using multiple clusterings and data objects
`// Partition the hyper-graph,` HG `into` k `components using HMETIS`
$\pi^{\star}(D) = \mathbf{HMETIS}(HG, k)$;
return $\pi^{\star}(D)$;

graph where each vertex is a cluster and each edge is the similarity between clusters which is measured using Jaccard measure. MCLA is composed of following three steps:

- Constructing the meta-graph,
- Partitioning the meta-graph
- Computing cluster members

Algorithm 6: Meta-Clustering Algorithm **MCLA**

Input: $\Pi(D)$: Multiple Clusterings of a Data Set D
k: Number of Clusters In the Final Clustering
Output: $\pi^{\star}(D)$: Final Clustering

`//` G `is a meta-graph, construct it`
$G = (V, E)$;
foreach $c \in \Pi(D)$ **do**
    Add c as a vertex to V;
foreach $v_1 \in V$ **do**
    foreach $v_2 \in V$ **do**
        if $v_1 \neq v_2$ **then**
            `// label the edge` (v_1, v_2)
            $label(v_1, v_2) = |v_1 \cap v_2|$;
$\pi^{\star}(D) = \mathbf{METIS}(G, k)$;
foreach $obj \in D$ **do**
    `//modify` $\pi^{\star}(D)$ `as follows`
    assign obj to its most associated cluster in $\pi^{\star}(D)$
return $\pi^{\star}(D)$;

Combining Multiple Clusterings Using Evidence Accumulation (EAC)

Evidence Accumulation (EAC) (Fred and Jain 2005) accumulates the evidence in each cluster to form a co-association matrix and it is provided to an agglomerative clustering algorithm, as shown in Algorithm 7.

In order to compute the co-association matrix, SM, enumerating all the pairs of objects at each cluster is necessary. Each pair updates (i.e. increments by 1) the corresponding entry in the co-association matrix. In other words, each entry in this matrix SM_{ij} is the number of times that objects i and j are assigned to the same clusters. For example, cluster $\mathbf{C_{11}}$ in Table 1.8 contributes following pairs to the co-association matrix: (d_1, d_1), (d_1, d_3), (d_1, d_6), (d_3, d_3), (d_3, d_6), and (d_6, d_6). This computation has quadratic time complexity.

Figure 1.9 shows the co-association matrix of Figure 1.8. Note that, Figure 1.9 represents the evidence accumulated by the pre-existing multiple clusterings (Fred and Jain 2005, Topchy et al. 2003, Fred and Jain 2002).

	$\mathbf{d_1}$	$\mathbf{d_2}$	$\mathbf{d_3}$	$\mathbf{d_4}$	$\mathbf{d_5}$	$\mathbf{d_6}$	$\mathbf{d_7}$	$\mathbf{d_8}$
$\mathbf{d_1}$	3	2	1	2	0	1	0	1
$\mathbf{d_2}$	2	3	0	2	0	0	1	2
$\mathbf{d_3}$	1	0	3	0	0	3	0	0
$\mathbf{d_4}$	2	2	0	3	1	0	0	1
$\mathbf{d_5}$	0	0	0	1	3	0	2	0
$\mathbf{d_6}$	1	0	3	0	0	3	0	0
$\mathbf{d_7}$	0	1	0	0	2	0	3	1
$\mathbf{d_8}$	1	2	0	1	0	0	1	3

Figure 1.9: Co-association matrix, SM, of figure 1.8

Algorithm 7: Evidence Accumulation **EAC**

Input: $\Pi(D)$: Multiple Clusterings of a Data Set D,
n: Number of Objects
Output: $\pi^{\star}(D)$: Final Clustering

```
Initialize SM to a n × n matrix ;
foreach π_i(D) ∈ Π(D) do
    Update SM ;
Run Agglomerative Clustering on SM to construct π*(D);
return π*(D);
```

Bipartite Merger (BM) and Metis Merger (MM)

Data may be distributed at different sites, in this case a distributed clustering solution with a final merging of clusters is needed. Hore et al. (2009) proposes two approaches (BM and MM) for combining clusters, represented by sets of cluster centers. Using cluster centers (prototypes) instead of clusters reduces computation and memory requirements. BM works on several clusterings each having n clusters. It groups the centroids according to their similarity and merges them to have a final clustering with n clusters. MM uses METIS, and it is more flexible: clusterings can have different number of clusters. Good results of both BM and MM are reported in Hore et al. (2009).

Some Recent Works for Combining Multiple Clusterings

Ayad and Kamel (2010) propose a voting-based cluster ensemble algorithm, which is a cumulative voting scheme. It generalizes the formulation of the voting problem as a regression problem with multiple input variables. A genetic algorithm, called MOCLE, which uses multiple objective function is introduced in Faceli et al. (2009). MOCLE uses clustering validation measures as objective function and it combines pairs of partitions in an optimization process at each iteration. Coelho et al. (2011) suggests a genetic approach as well. Selective spectral clustering algorithm (SELSCE), which uses a bagging technique to pick the good clustering, is proposed in Jia et al. (2011).

Weaknesses of Related Work

CSPA, HGPA, MCLA, EAC, and LCE require the number of final clusters in advance. EAC and CSPA do not scale very well, because they all work at object level. These techniques may not accurately capture the relationship between clusters, which is another disadvantage. Although HGPA is very fast, it is not very accurate due to the degenerative effect of noise clusters. MCLA uses Jaccard measure, which only captures syntactical similarity between clusters. LCE starts with a bipartite membership graph of objects and clusters. But, LCE builds up a dense graph with implied similarities between every cluster and every object which needs a lot of computation.

Although median partition methods implicitly estimate the number of clusters, finding the median partition is a very complex problem. These methods suffer from slow execution times, mainly because they work on object level. Noisy input clusterings may considerably affect median partition based clustering ensemble methods, which is another disadvantage.

Genetic methods suffer from long execution times. In the domain of clustering ensemble, determining chromosome encoding, crossover, mutation, and the fitness function are not immediate and trivial.

1.5 THESIS OVERVIEW

This dissertation proposes a novel and efficient similarity-graph based method for combining multiple clusterings.

Chapter 1 provides preliminaries and non-exhaustive literature survey of clustering and combining multiple clusterings, for better understanding and completeness of the dissertation.

The following chapter, Chapter 2, introduces the novel similarity-graph based algorithm for combining multiple clusterings. The algorithm works on a similarity-graph and is very efficient to construct a final clustering. Chapter 3, provides discussion of the algorithm and experimental results on real, synthetically generated, and gene expression data sets.

Concluding remarks, discussions and future work is presented in Chapter 4.

2. COMUSA: COMBINING MULTIPLE CLUSTERINGS USING SIMILARITY GRAPH

In this chapter, we introduce a novel graph-based method, COMUSA (Mimaroglu and Erdil 2011), for combining multiple clusterings. The problem of combining multiple clusterings into a final clustering has gained importance recently. Each clustering technique possess advantages and disadvantages: most of them are strongly affected by input parameter values and makes an assumption about the data set. Different distance metrics may also lead to obtain different clusterings. In brief, there are several reasons to obtain multiple clusterings of a data set. It is beneficial to construct multiple clusterings using several clustering methods, several distance metrics, and several input parameters. Combining multiple clusterings enables aggregating the benefits of pre-existing knowledge and producing a better quality final clustering. Also, it is expected that the final clustering is novel, robust, and scalable. In order to solve this challenging problem we propose a new graph-based method. Our method accumulates the evidence using the input multiple clusterings, and produces a novel final clustering which has better overall quality. The number of clusters in the final clustering is detected automatically; this is another big advantage of our technique. Extensive experimental test results on real, synthetically generated, and gene expression data sets demonstrate the effectiveness of our new method.

2.1 COMUSA

In this section, we explain the working principles of our algorithm, COMUSA, with details. COMUSA operates on a similarity graph. The similarity graph is an undirected and weighted graph that represents co-association (similarity) matrix. The co-association matrix is obtained by using evidence accumulation from the multiple clusterings as explained in Section 1.4.3.

The similarity graph, $SG = (D, E)$, constructed in COMUSA is object-wise which means that it represents the similarities between objects. Each edge, (d_i, d_j), of the graph has a weight which corresponds to the entry SM_{ij} in the co-association matrix. For simplicity of the similarity graph, we omit the edges having 0 weight and edge labels of value 1, i.e $SM_{ij} = 1$. Also, self loops (i.e. all the edges (d_i, d_i)) are disregarded in COMUSA,

because this information is redundant in the process of constructing a final clustering as well.

The definitions, which play a major role for understanding COMUSA, are given below.

Definition 2.1.1. *The **degree of freedom** of a vertex d_i is:*

$$\mathsf{df}(d_i) = |\{d_j | (d_j, d_i) \in E\}|$$

Definition 2.1.2. *For a data set D, and a family of clusterings $\Pi(D)$, let SM be the corresponding co-association matrix. Edges are labeled by the function weight defined by*

$$\mathsf{weight}(d_i, d_j) = SM_{ij},$$

where SM_{ij} is the entry at row i and column j of SM.

Definition 2.1.3. *The **sum of weights** of edges incident to a vertex d_i is the*

$$\mathsf{sw}(d_i) = \sum_{j=1, j\neq i}^{|D|} \mathsf{weight}(d_j, d_i).$$

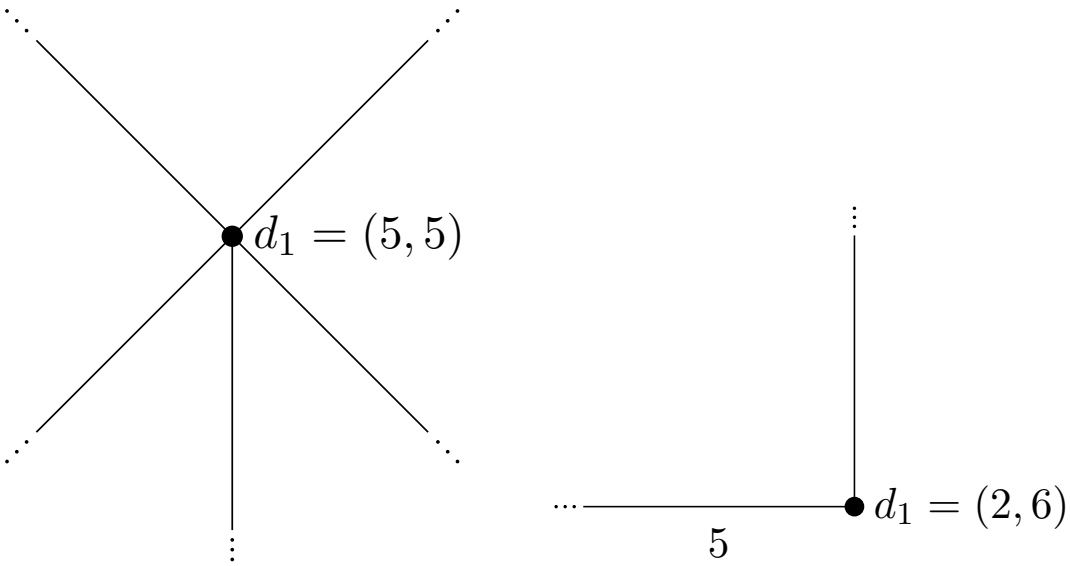

Figure 2.1a $\mathsf{df}(d_1) = \mathsf{sw}(d_1) = 5$ **Figure 2.1b** $\mathsf{df}(d_1) = 2$, $\mathsf{sw}(d_1) = 6$

Figure 2.1: df and sw

Lemma 2.1.4. *For a vertex d_i of a similarity graph we have*

$$\mathsf{sw}(d_i) \geq \mathsf{df}(d_i).$$

Proof: The inequality is immediate from the definitions of df and sw. ∎

Definition 2.1.5. *The* attachment *of a vertex d_i is given by:*

$$\mathit{attachment}(d_i) = \frac{\mathsf{sw}(d_i)}{\mathsf{df}(d_i)}.$$

There may be isolated nodes having 0 degree of freedom and 0 sum of weights. By convention, attachment value of such a vertex is considered as 0.

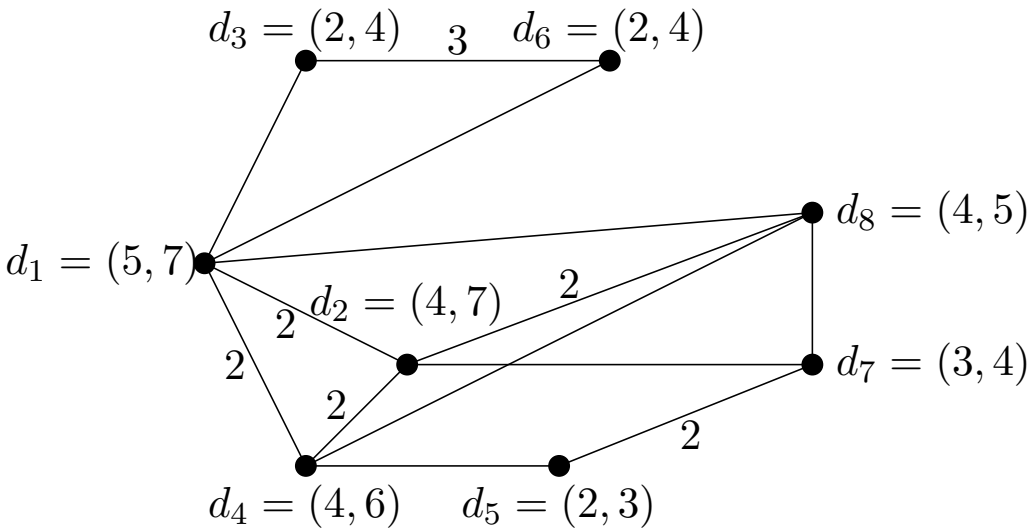

Figure 2.2: Similarity graph of figure 1.9

Each vertex, d_i, of the similarity graph is labeled by the degree of freedom and sum of weights in form $(\mathsf{df}(d_i), \mathsf{sw}(d_i))$. Similarity between two data objects d_i and d_j is illustrated with an edge labeled by the value $\mathsf{weight}(d_i, d_j)$. Representations of df, sw, and weight are sampled on two partial similarity graph in Figure 2.1. Low values of $\mathsf{df}(d_i)$ means that d_i is connected to less number of vertices. In a similar manner, high values of sum of weights of d_i, $\mathsf{sw}(d_i)$, indicate that d_i is connected to its neighbors strongly.

Let us consider a data object d_i. Low value of $\mathsf{df}(d_i)$ and high value of $\mathsf{sw}(d_i)$ is desirable because this gives us a useful information about the tendency of the data point. It means that, d_i is strongly connected to small number of objects and most probably they will be clustered together. Therefore, it can be beneficial to initialize a cluster by starting such data objects. We suggest attachment (see Definition 2.1.5) to initiate new clusters; an object, which have not been assigned into a cluster (unmarked), having the highest attachment is selected as a pivot data point as a singleton cluster in COMUSA. Then, the pivot object expands the cluster at hand as much as possible as explained below.

The algorithm of **COMUSA**, **Co**mbining **M**ultiple Cl**u**sterings Using **S**imilarity Gr**a**ph, is given in Algorithm 8.

Algorithm 8: **Co**mbining **M**ultiple Cl**u**sterings Using **S**imilarity Gr**a**ph **COMUSA**

Input: $\Pi(D)$: Multiple Clusterings
Output: $\pi^{\star}(D)$: Final Clustering

Initialize an empty queue Q;
$clusterId = 1$;
Construct similarity graph $SG = (D, E)$ using $\Pi(D)$, and D;
Sort D in decreasing order with respect to attachment ;
while *there are unmarked objects* **do**
    Add unmarked object, d_i, with highest $\mathsf{attachment}(d_i)$ to Q ;
    while *Q is not empty* **do**
        `// pivot object`
        $v =$ remove first element from Q ;
        Add v to cluster $clusterId$;
        Mark v ;
        foreach $(w, v) \in E$ **do**
            if w *is marked* **then**
                continue ;
            else
                $strWeight = \mathsf{weight}(w, v)$;
                $isMax = true$;
                foreach $(z, w) \in E$ **do**
                    `// maximum constraint`
                    if $strWeight \ngeq \mathsf{weight}(z, w)$ **then**
                        $isMax = false$;
                        break ;
                if $isMax$ **then**
                    Add w to Q ;
    $clusterId$++ ;

The family of clusterings of a data set having 8 data objects is shown in Figure 1.8. The co-association matrix of this multiple clusterings and the corresponding similarity graph are shown in Figure 1.9 and Figure 2.2, respectively. We demonstrate COMUSA on this similarity graph for understanding of the algorithm better. Initially, attachment values are computed in line 4 of Algorithm 8 for each data object to find the pivot object as shown in Table 2.1. An unmarked object having the highest attachment value is chosen as pivot. In this example, d_3 and d_6 have the highest attachment values; we randomly pick d_3, initiate a new cluster $C_1^\star$, and assign d_3 into $C_1^\star$. COMUSA proceeds

Table 2.1: attachment values of figure 2.2 in decreasing order

vertex	attachment$(vertex)$
d_3	2.00
d_6	2.00
d_2	1.75
d_4	1.50
d_5	1.50
d_1	1.40
d_7	1.33
d_8	1.25

to discover the data objects that will be assigned into the same cluster with d_3. First, pivot object, d_3, checks its neighbors. Immediate neighbors of d_3 are d_1 and d_6. $C_1^\star$ cannot be expanded by d_1 because it does not have the maximum edge weight with d_3 (weight$(d_3, d_1) \not\geq$ weight(d_1, d_4), similarly weight$(d_3, d_1) \not\geq$ weight(d_1, d_2)). d_6 is assigned into d_3's cluster because the inequality weight$(d_3, d_6) \geq$ weight(d_6, d_1) satisfies the maximum constraint (line 18) in the algorithm. Now, d_6 acts like pivot object and tries to expand the cluster $C_1^\star$. The only unmarked neighbors of d_6 is d_1. However, d_1 does not have its maximum connection with d_6 as well. Since d_6 does not have any further neighbor, the algorithm backtracks one step and again d_3 becomes pivot. But, d_3 also does not have any unchecked neighbor, means that $C_1^\star$ cannot be expanded anymore. Our first cluster, shown with dashed pattern in Figure 2.3a, forms with two objects $C_1^\star = \{d_3, d_6\}$.

Since there are some objects that are not marked (not assigned into any cluster), the algorithm keeps running by choosing a new pivot object. d_2 has the highest attachment value among unmarked objects and is selected as pivot. A new cluster, $C_2^\star$, is created and COMUSA expands it similarly. The unmarked neighbors of d_2 are d_1, d_4, d_7, and d_8. d_2 includes d_1 because weight$(d_1, d_2) = 2$ which is one of the maximum connection of d_1. Then, d_1 becomes pivot and d_4 is assigned into $C_2^\star$ in a similar manner. But, d_4 cannot expand the cluster any further. Again, d_2 becomes acting pivot object and evaluates its unchecked neighbors d_7 and d_8. d_7 cannot be added into second cluster since weight$(d_2, d_7) \leq$ weight(d_7, d_5). Next, the object d_8 is included in the cluster since among all the edges passing through d_8, weight(d_2, d_8) has the maximum value. d_8 becomes an acting pivot but cannot expand $C_2^\star$. Since there are no unchecked neighbor of acting pivots remain, further expansion is not possible: we have $C_2^\star = \{d_2, d_1, d_4, d_8\}$ which is depicted with dotted pattern in Figure 2.3b.

There are only two unmarked objects left which are d_5 and d_7. The object d_5 has the highest attachment value so it is chosen as a pivot object. Its only unmarked neighbor is d_7. weight$(d_5, d_7) \geq$ weight(d_7, d_2) and weight$(d_5, d_7) \geq$ weight(d_7, d_8), therefore d_7 and d_5 are clustered together, so $C_3^\star = \{d_5, d_7\}$. All the objects are marked, COMUSA terminates. Final clustering having three clusters are shown in Figure 2.3.

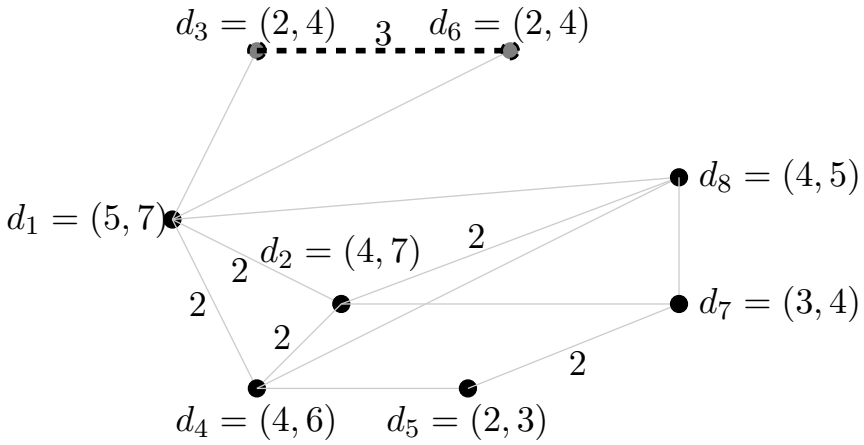

Figure 2.3a First cluster in dashed pattern

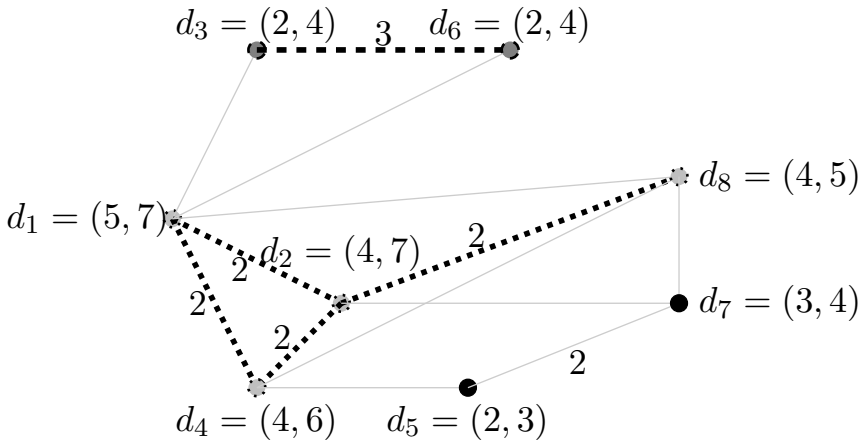

Figure 2.3b Second cluster in dotted pattern

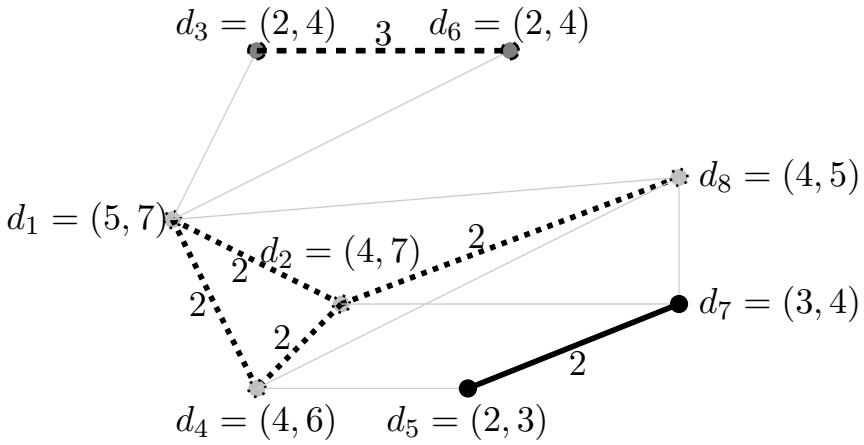

Figure 2.3c Third cluster in straight bold line

Figure 2.3: Generating final clustering using COMUSA on figure 2.2

We also perform COMUSA on two more toy examples. Next two examples show that COMUSA is robust, and intuitive with respect to similarity graph.

Example 2.1.6. *Let us consider the similarity graph shown in Figure 2.4a. Notice that all the edge labels are* 2*, and* attachment *values for all the vertices are constant. Each vertex is qualified to be a pivot, and no matter what vertex is selected as the pivot we end up with one big cluster having all the vertices:* $C_1^\star = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\}$ *as shown in Figure 2.4b.*

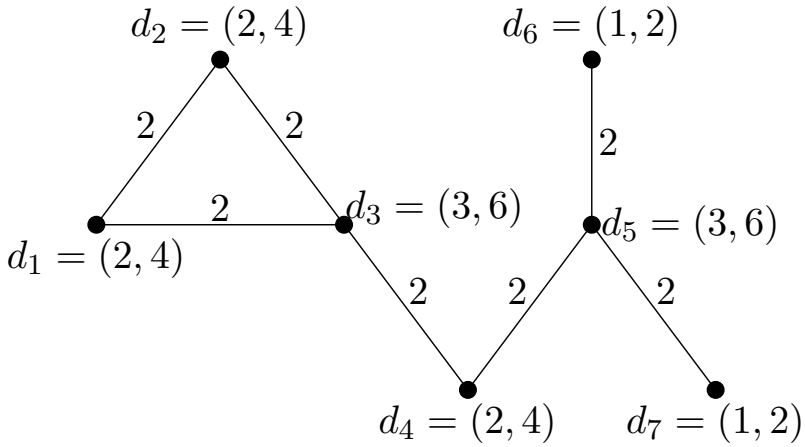

Figure 2.4a Similarity graph of a data set

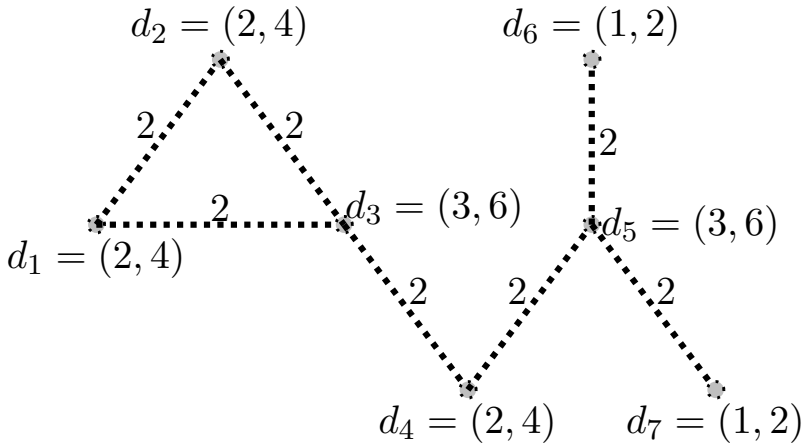

Figure 2.4b Final clustering of Figure 2.4a in dashed pattern

Figure 2.4: COMUSA on a data set

Example 2.1.7. *Running COMUSA on the similarity graph shown in Figure 2.5a generates* 4 *clusters:* $C_1^\star = \{d_3, d_8\}$*,* $C_2^\star = \{d_1, d_4, d_5\}$*,* $C_3^\star = \{d_2, d_6\}$*, and* $C_4^\star = \{d_7\}$*. This result is very intuitive too, objects having high values of similarity are grouped in the same clusters. Also, note that isolated object* d_7 *is left by itself in a cluster. Final clustering produced by COMUSA is shown in Figure 2.5b.*

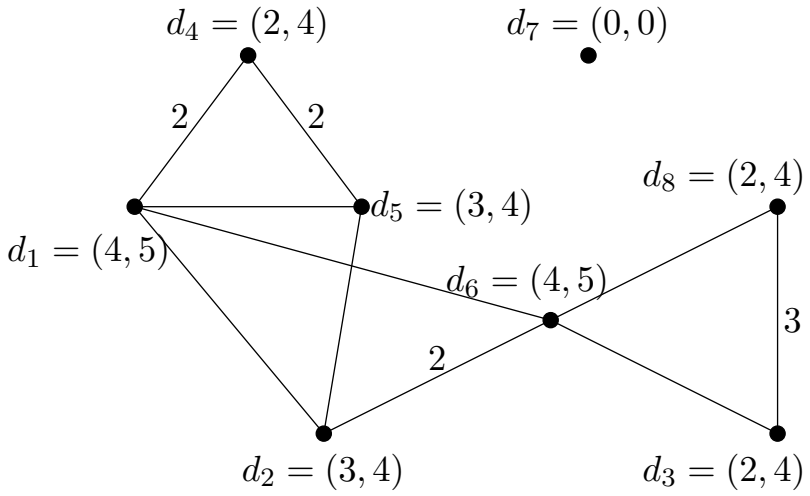

Figure 2.5a Similarity graph of another data set

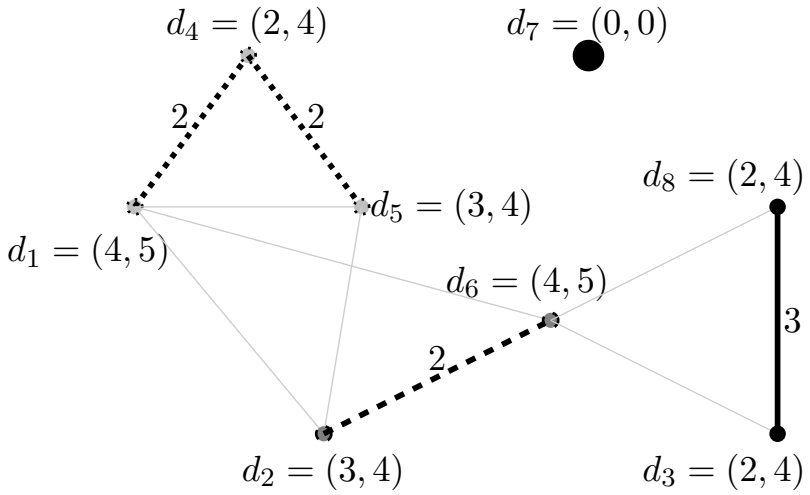

Figure 2.5b Final clustering of Figure 2.5a having 4 clusters. 3 clusters are shown with distinct patterns, d_7 is a singleton cluster.

Figure 2.5: COMUSA on another data set

COMUSA initiates a new cluster with an object having the highest attachment value, then extends the cluster in a greedy manner. In a similarity graph, neighbors of a pivot are checked with respect to their similarity to the pivot. Then, each neighbor is considered as an acting pivot. In a final clustering the number of clusters depends on the data set: COMUSA detects this number automatically, which is a big advantage.

2.1.1 Relaxation

Expansion of a cluster depends on the maximum constraint in COMUSA. However, maximum constraint may frustrate the objects placing in the same cluster even if they are very similar. Moreover, in some cases larger clusters may be desired, which cannot be possible due to the condition.

Maximum constraint can be relaxed with a user specified ratio called **relaxation**, r. Therefore, the condition in if statement (in line 18) become $strWeight + strWeight.r \not\geq$ weight(z, w). By increasing relaxation ratio, fewer clusters having larger size are obtained. Thus, the parameter may contribute for finding correct number of clusters efficiently. Experimental results demonstrate that adjusting the relaxation ratio affects the quality of final clustering. However, there is no rule of thumb for ideal relaxation value in advance, it depends on the input information.

We explain relaxation ratio by performing COMUSA on a partial similarity graph that is shown in Figure 2.6. d_1 is the vertex having the highest attachment value, so it is selected as pivot. COMUSA tries to extend the cluster with d_2, but this is not possible since $weight(d_2, d_3) > weight(d_1, d_2)$. Let us assume that the relaxation ratio is specified as 25%. In this scenario, d_2 is assigned into d_1's cluster since $weight(d_1, d_2) + weight(d_1, d_2).25\% \geq weight(d_2, d_3)$. Then, d_2 becomes acting pivot and d_3 is included into the same cluster as well. Notice that COMUSA with a positive relaxation value produces larger, and fewer clusters.

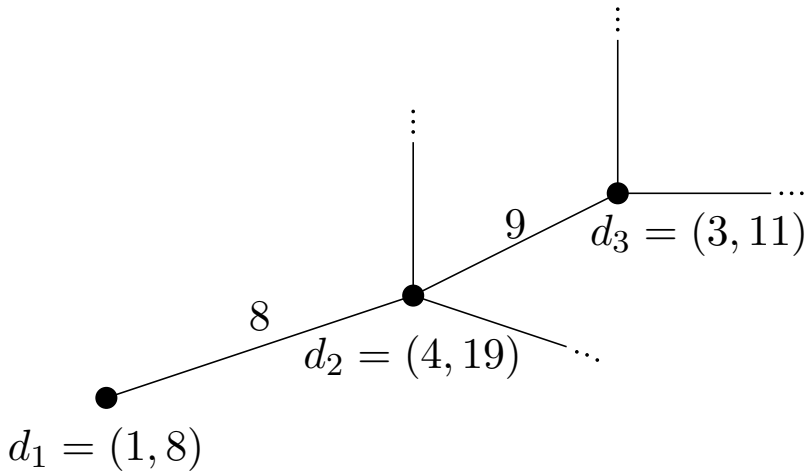

Figure 2.6: A partial similarity graph

3. DISCUSSION AND EXPERIMENTAL RESULTS

In this section, we discuss the important features of COMUSA. We also provide our experimental results on real, synthetically generated, and gene expression data sets.

3.1 DISCUSSION OF COMUSA

COMUSA initiates a new cluster by selecting a pivot (seed) object. This step is crucial in COMUSA because pivot objects are good starting points. High values of sum of weights and low values of degree of freedom indicate high attachment values which means that such objects are strongly attached (connected) to its neighbors. Therefore, clustering objects starting from pivot object enables cluster compactness which is expected.

The process of expanding a cluster is at least as important as initiating a new cluster. Pivot object expands the cluster by considering all the immediate neighbors. A neighbor is assigned into the pivot's cluster when only it is most similar to the pivot. In other words, pivot object always tries to pull its neighbors into its own cluster. If a neighbor is included, it is marked and then acts like a pivot. New pivot also considers its immediate neighbors for further expansion. Therefore, there may remain some unchecked neighbors of old pivot. These neighbors are checked in further steps. If a pivot cannot expand a cluster any more, previous pivot becomes pivot again and algorithm iterates by checking its unchecked neighbors. Finally, all the neighbors of all the pivot objects are checked and then expansion of a cluster comes to an end.

Each data object that is assigned into a cluster becomes marked in COMUSA. After a cluster is formed, there may still remain some unclustered (unmarked) objects. COMUSA keeps constructing other clusters by selecting a new pivot among unmarked objects and they are expanded similarly. COMUSA terminates when all the data objects are marked, i.e. belong to a cluster.

Arbitrary shape clusters can be found by our algorithm, we do not make any assumptions about the input data set. COMUSA works very well because pivot objects are good starting points, and an object is included into a cluster if the object is most similar to a pivot in that cluster. Experimental results show that in a short amount of time COMUSA creates

very good quality clusters on real and synthetic data sets, even on very challenging ones (see Figure A.1a, A.2, A.3, A.4), therefore remedies the weaknesses of the related work.

3.2 EXPERIMENTAL EVALUATIONS

This section includes experimental results of COMUSA on varying data sets from different domains and having different properties. Generating cluster ensembles and properties of test data sets are also presented in the section for prior knowledge.

3.2.1 Generating Cluster Ensembles

Combining multiple clusterings techniques take a collection of clusterings of a data set. Therefore, approaches for generating cluster ensembles play an important role in combining multiple clusterings process. In our experiments, we generated cluster ensembles with three different approaches: manually constructing clusters, randomly constructing clusters or randomly injecting error into the original clusters, and using k-means algorithm with varying k-values. The main benefit of using different approaches is to produce a diverse set of clusterings having different properties and qualities. Note that, the diversity and quality of a cluster ensemble affects the final clustering's quality.

We used 4 real and 7 synthetic data sets in our experiments. The properties of input multiple clusterings for real and synthetically generated data sets are presented in Table B.1. For example, the input generated from Breast Cancer data set has 5 clusterings, each clustering with 2 to 5 clusters, and each clustering is generated by k-means, manually or at random. The min, max and average quality of input clusterings are given in the table as well. For Breast Cancer data set, min clustering quality is 0.077, max clustering quality is 0.525, and average clustering quality is 0.309.

We also evaluated the performance of COMUSA on 34 gene expression data sets. The properties of gene expression data sets and input multiple clusterings can be seen in Table B.2 and Table B.3, respectively.

3.2.2 Test Results of COMUSA on Real, Synthetically Generated, and Gene Expression Data Sets

We have conducted experiments on a computer having 2.8GHz processor with 4GB of main memory, running on Linux kernel 2.6. Our choice of implementation language is Java, which provides built-in support for bit vectors, and operations on bit vectors. COMUSA, MCLA, and EAC are all implemented in Java and are tested with Java Development Kit 1.6.0_16. We obtained PMETIS, KMETIS, and HMETIS from the corresponding authors. PMETIS and KMETIS belong to the METIS package and are implemented in C language. HMETIS is also implemented in C. LCE in implemented in MATLAB.

A synthetically generated data set 1-spiral contains 100 data objects. 2-spiral, 2-half rings, 2-curve data sets are also synthetically generated and contain 200, 118, and 192 objects respectively. Although these are small and low dimensional (2 dimensions only) data sets, identifying correct clusterings of these data sets is very challenging for both some clustering and combining multiple clusterings methods.

2D2K and 8D5K data sets are taken from Strehl and Ghosh (2003). 2D2K contains 500 points each of two 2-dimensional Gaussian clusters with means $(-0.227, 0.077)$ and $(0.095, 0.323)$ and diagonal covariance matrices with 0.1 for all diagonal elements. 8D5K contains 1000 points from five multivariate Gaussian distributions (200 points each) in 8-dimensional space. The clusters all have the same variance (0.1), but different means. Means were drawn from a uniform distribution within the unit hypercube. Syn5K data set is also artificially generated and contains 5000 data objects 5 classes.

Real data sets that we used in our experiments are obtained from University of California Irvine Machine Learning Repository (A. Asuncion 2007). Iris, Glass, Breast Cancer, and Image Segmentation data sets are all multivariate. Iris has 4 dimensions, 150 objects, and 3 classes. Glass data set has 10 dimensions, 214 objects, and 6 classes. Breast Cancer is a data set having 9 attributes, 286 objects and 2 classes. Last, Image Segmentation has 19 real attributes with 2310 objects and 7 classes.

COMUSA is tested on 1-spiral data set, shown in Figure A.1a, with two different input multiple clusterings: 1-spiral, hand clustered and 1-spiral, k-means clustered. The input 1-spiral, hand clustered consists of two partial clusterings. These clusters are shown in Figure A.1b, where a clustering is represented with rectangular shape, and another clustering is represented with elliptical shape. Notice that both of clusterings are partial.

COMUSA successfully finds the 1-spiral data set for the input 1-spiral, hand clustered. 1-spiral, k-means clustered is produced by performing k-means algorithm on 1-spiral data set with inputs $k = 2$ and $k = 3$ twice for each. Thus, we obtained 4 different clusterings. COMUSA successfully discovered the natural clusters with 34% relaxation. Results of COMUSA on 1-spiral data set are shown in Table B.4. Note that COMUSA is not compared with another combining multiple clusterings methods. Because all of them take number of clusterings as input, so it is meaningless to provide number of clusters as 1.

Figures A.2, A.3 and A.4 demonstrate the 2-spiral, 2-curve, and 2-half rings data sets, respectively. Multiple clusterings of these data sets are obtained using different approaches as well. Partitions generated by k-means on the 2-half rings data set are shown in Figure A.5. The results of COMUSA, PMETIS, KMETIS, HMETIS, MCLA, and EAC are compared for cluster validity, as shown in Table B.5. COMUSA produces perfect outputs on all the data sets. For these data sets PMETIS, KMETIS, HMETIS, MCLA and EAC are requested to produce 2 clusters for fairness. The results of Syn5K data set are also shown in Table B.5.

ECS+ICS validity measure results of 2D2K and 8D5K data sets are compared to PMETIS, KMETIS, HMETIS, MCLA, and EAC results as shown in Table B.6. Different number of clusters, k, including the correct number of clusters are provided to PMETIS, KMETIS, HMETIS, MCLA, and EAC which can also be seen in the table. Definitely, the quality of final clustering constructed by COMUSA is superior to other final clustering produced by other methods.

COMUSA produces good quality final clusterings on real data sets as well. As shown in Table B.7, COMUSA produces better results on Glass and Breast Cancer data sets. On the remaining data sets, the results of COMUSA is very close to the highest results.

COMUSA can also be used in Bioinformatics domain to perform combining multiple clusterings on biological data sets (Mimaroglu and Erdil 2010). We conduct experiments on 34 gene expression data sets. The results of COMUSA are only compared with LCE, because LCE is designed to work on Bioinformatics domain. As it can clearly be seen from Table B.7 COMUSA is superior to LCE on 21 data sets.

We also compared the execution time results (see Table B.9) of COMUSA, PMETIS, KMETIS, HMETIS, MCLA, and EAC for the data sets and input clusterings in Tables B.5, B.6, and B.7. Gene expression data sets are not included to time results since

they are very small; all methods run fast on these data sets. Clearly, COMUSA is faster than both MCLA and EAC on all the data sets except Image Segmentation and Syn5K. It is also faster than PMETIS and KMETIS for 2-spiral, 2-curve, 2D2K, 8D5K, Glass, and Breast Cancer data sets. For 3 data sets, COMUSA is faster than HMETIS. Shortly, COMUSA is comparable to METIS package algorithms except very large data sets. Note that COMUSA is implemented in Java, which is known to be slower than C language implementations. COMUSA iterates over all the edges of the similarity graph regardless of the relaxation input. Therefore, performing COMUSA with different relaxation values does not effect the execution time considerably.

As we mentioned before, COMUSA does not take number of clusters in the final clustering as its input; detects it automatically. In Table B.10, number of clusters obtained by COMUSA is compared with natural number of clusters. According to experiments conducted for all the data sets, COMUSA is able to find correct number of clusters or close to this number.

All the test data sets and COMUSA implementation are available at
`akademik.bahcesehir.edu.tr/~eerdil/comusa`.

4. CONCLUSION

In this thesis, we introduced a novel method for combining multiple clusterings. COMUSA takes a collection of clusterings as its input and produces a good quality final clustering. Relaxation rate parameter which affects the quality of final clusterings can also be provided to COMUSA. COMUSA does not take the number of clusters in the final clustering; this number is automatically computed by COMUSA.

Our algorithm constructs a similarity graph of objects using multiple input clusterings where similarity graph is the backbone structure for discovering connected components.

Automatically finding the number of clusters in the final clusterings is one of the most important feature of COMUSA. This feature of COMUSA can be explained as follows: A pivot object includes another object into the cluster if it is more closely connected to the pivot than to any other unmarked vertices. Therefore, after several iterations, a cluster cannot be expanded further and forms a cluster automatically. Since all clusters are formed in this manner, COMUSA does not the need number of clusters as input parameter. COMUSA comes to an end when all data objects belong to a cluster.

The quality of input clusterings impact both the quality of final clustering and the number of clusters in the final clustering. Therefore, ensemble generation methods impact the outcome.

COMUSA is partitional, novel, and complete. Extensive experimental evaluations on many real, synthetically generated and gene expression data sets demonstrate that COMUSA: (1) works well on arbitrary shape clusters, (2) is not affected by the cluster size, (3) is not affected by noise and outliers, (4) is not affected by the sparseness of the data set, (5) is order independent,and (6) is deterministic.

The similarity graph constructed in COMUSA is object-wise, where each vertex of the graph represents an object. Since data sets may have wide range of data objects, the similarity graph can be very large and dense. COMUSA iterates over all the edges and vertices of the similarity graph while constructing a final clustering which is very costly. Therefore, COMUSA suffers from long execution time for very large data sets. This is the most important shortcoming of our method. As future work, our aim is to make COMUSA feasible for very large data sets. Constructing a cluster-wise similarity graph instead

of object-wise similarity graph may reduce the execution time of COMUSA. Because, number of clusters in the multiple clusterings is generally much less than the number of objects.

REFERENCES

Books

Alpaydin, E.: 2004, *Introduction to machine learning*, The MIT Press.

Bellman, R.: 2003, *Dynamic programming*, Dover Pubns.

Bishop, C.: 2006, *Pattern recognition and machine learning*, Vol. 4, Springer New York.

Everitt, B., Landau, S., Leese, M. and Stahl, D.: 2011, *Cluster analysis 5th Edition*, John Wiley & Sons.

Everitt, B. S.: 1974, *Cluster analysis*, John Wiley & Sons.

Feldman, R. and Sanger, J.: 2007, *The text mining handbook: advanced approaches in analyzing unstructured data*, Cambridge Univ Pr.

Forsyth, D. and Ponce, J.: 2002, *Computer vision: a modern approach*, Prentice Hall Professional Technical Reference.

Gray, R.: 2010, *Entropy and information theory*, Springer Verlag.

Han, J. and Kamber, M.: 2006, *Data mining: concepts and techniques*, Morgan Kaufmann.

Höppner, F., Klawonn, F., Kruse, R. and Runkler, T.: 1999, *Fuzzy cluster analysis: methods for classification, data analysis, and image recognition*, John Wiley & Sons.

Jain, A. and Dubes, R.: 1988, *Algorithms for clustering data*.

Kaufman, L., Rousseeuw, P. and Corporation, E.: 1990, *Finding groups in data: an introduction to cluster analysis*, Vol. 39, Wiley Online Library.

Lee, K.: 2005, *First course on fuzzy theory and applications*, Vol. 27, Springer.

MacCuish, J. and Maccuish, N.: 2010, *Clustering in Bioinformatics and Drug Discovery*, Chapman & Hall/CRC mathematical and computational biology series, Taylor & Francis Group.

Tan, P., Steinbach, M., Kumar, V. et al.: 2006, *Introduction to data mining*, Pearson Addison Wesley Boston.

Periodicals

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O. and Staudt, L. M.: 2000, Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling, *Nature* **403**(6769), 503–511.

Ana, L. and Jain, A.: 2003, Robust data clustering, **2**, II–128 – II–133 vol.2.

Armstrong, S. A., Staunton, J. E., Silverman, L. B., Pieters, R., den Boer, M. L., Minden, M. D., Sallan, S. E., Lander, E. S., Golub, T. R. and Korsmeyer, S. J.: 2002, MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia., *Nat Genet* **30**(1), 41–47.

Ayad, H. G. and Kamel, M. S.: 2010, On voting-based consensus of cluster ensembles, *Pattern Recognition* **43**(5), 1943 – 1953.

Baek, J. and McLachlan, G. J.: 2011, Mixtures of common t-factor analyzers for clustering high-dimensional microarray data, *Bioinformatics* **27**(9), 1269–1276.

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E. J., Lander, E. S., Wong, W., Johnson, B. E., Golub, T. R., Sugarbaker, D. J. and Meyerson, M.: 2001, Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses., *Proc Natl Acad Sci U S A* **98**(24), 13790–13795.

Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., Hendrix, M., Radmacher, M., Simon, R., Yakhini, Z., Ben-Dor, A., Sampas, N., Dougherty, E., Wang, E., Marincola, F., Gooden, C., Lueders, J., Glatfelter, A., Pollock, P., Carpten, J., Gillanders, E., Leja, D., Dietrich, K., Beaudry, C., Berens, M., Alberts, D. and Sondak, V.: 2000, Molecular classification of cutaneous malignant melanoma by gene expression profiling., *Nature* **406**(6795), 536–540.

Borah, B. and Bhattacharyya, D.: 2004, An improved sampling-based dbscan for large spatial databases, *Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on*, IEEE, pp. 92–96.

Bredel, M., Bredel, C., Juric, D., Harsh, G. R., Vogel, H., Recht, L. D. and Sikic, B. I.: 2005, Functional network analysis reveals extended gliomagenesis pathway maps and three novel MYC-interacting genes in human gliomas., *Cancer Res* **65**(19), 8679–8689.

Chen, X., Cheung, S. T., So, S., Fan, S. T., Barry, C., Higgins, J., Lai, K., Ji, J., Dudoit, S., Ng, I. O. L., van de Rijn, M., Botstein, D. and Brown, P. O.: 2002, Gene expression patterns in human liver cancers, *Mol. Biol. Cell* **13**(6), 1929–1939.

Chowdary, D., Lathrop, J., Skelton, J., Curtin, K., Briggs, T., Zhang, Y., Yu, J., Wang, Y. and Mazumder, A.: 2006, Prognostic gene expression signatures can be measured in tissues collected in RNAlater preservative., *J Mol Diagn* **8**(1), 31–39.

Coelho, A. L., Fernandes, E. and Faceli, K.: 2011, Multi-objective design of hierarchical consensus functions for clustering ensembles via genetic programming, *Decision Support Systems* **In Press, Corrected Proof**, –.

Cristofor, D. and Simovici, D.: 2002, Finding median partitions using information-theoretical-based genetic algorithms, *Journal of Universal Computer Science* **8**(2), 153–172.

Dempster, A., Laird, N. and Rubin, D.: 1977, Maximum likelihood from incomplete data via the em algorithm, *Journal of the Royal Statistical Society. Series B (Methodological)* **39**(1), 1–38.

Denud, L. and Gunoche, A.: 2006, Comparison of distance indices between partitions, *Data Science and Classification* pp. 21–28.

Dyrskjot, L., Thykjaer, T., Kruh0ffer, M., Jensen, J. L., Marcussen, N., Hamilton-Dutoit, S., Wolf, H. and Orntoft, T. F.: 2003, Identifying distinct classes of bladder carcinoma using microarrays., *Nat Genet* **33**(1), 90–96.

Ester, M., Kriegel, H., Sander, J. and Xu, X.: 1996, A density-based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, Vol. 1996, Portland: AAAI Press, pp. 226–231.

Faceli, K., de Souto, M. C., de Araújo, D. S. and de Carvalho, A. C.: 2009, Multi-objective clustering ensemble for gene expression data analysis, *Neurocomputing* **72**(13-15), 2763 – 2774. Hybrid Learning Machines (HAIS 2007) / Recent Developments in Natural Computation (ICNC 2007).

Filkov, V. and Skiena, S.: 2004, Heterogeneous data integration with the consensus clustering formalism, pp. 110–123.

Fred, A. and Jain, A.: 2002, Data clustering using evidence accumulation, **4**, 276–280.

Fred, A. and Jain, A.: 2005, Combining multiple clusterings using evidence accumulation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 835–850.

Garber, M. E., Troyanskaya, O. G., Schluens, K., Petersen, S., Thaesler, Z., Pacyna-Gengelbach, M., van de Rijn, M., Rosen, G. D., Perou, C. M., Whyte, R. I., Altman, R. B., Brown, P. O., Botstein, D. and Petersen, I.: 2001, Diversity of gene expression in adenocarcinoma of the lung., *Proc Natl Acad Sci U S A* **98**(24), 13784–13789.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. and Lander, E. S.: 1999, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring., *Science* **286**(5439), 531–537.

Gordon, G. J., Jensen, R. V., Hsiao, L., Gullans, S. R., Blumenstock, J. E., Ramaswamy, S., Richards, W. G., Sugarbaker, D. J. and Bueno, R.: 2002, Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma., *Cancer Res* **62**(17), 4963–4967.

Gunopulos, R. and Raghavan, P.: 1998, Automatic subspace clustering of high dimensional data for data mining applications, *Proceedings of ACM SIGMOD International Conference on Management of Data*, Citeseer.

Hinneburg, A. and Keim, D.: 1998, An efficient approach to clustering in large multimedia databases with noise, *Knowledge Discovery and Data Mining* **5865**, 58–65.

Hinneburg, A. and Keim, D.: 1999, Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering, *Proceedings of the 25th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc., p. 517.

Hore, P., Hall, L. O. and Goldgof, D. B.: 2009, A scalable framework for cluster ensembles, *Pattern Recognition* **42**(5), 676 – 688.

Hubert, L. and Arabie, P.: 1985, Comparing partitions, *Journal of classification* **2**(1), 193–218.

Iam-On, N., Boongoen, T. and Garrett, S.: 2010, Lce: a link-based cluster ensemble method for improved gene expression data analysis, *Bioinformatics* **26**(12), 1513.

Jain, A. K.: 2010, Data clustering: 50 years beyond k-means, *Pattern Recognition Letters* **31**(8), 651 – 666. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR), 19th International Conference in Pattern Recognition (ICPR).

Jia, J., Xiao, X., Liu, B. and Jiao, L.: 2011, Bagging-based spectral clustering ensemble selection, *Pattern Recognition Letters* **32**(10), 1456 – 1467.

Kalogeratos, A. and Likas, A.: 2011, Document clustering using synthetic cluster prototypes, *Data & Knowledge Engineering* **70**(3), 284 – 306.

Karypis, G., Aggarwal, R., Kumar, V. and Shekhar, S.: 1997, Multilevel hypergraph partitioning: application in vlsi domain, pp. 526–529.

Karypis, G., Han, E. and Kumar, V.: 1999, Chameleon: Hierarchical clustering using dynamic modeling, *Computer* **32**(8), 68–75.

Karypis, G. and Kumar, V.: 1998, Multilevel k-way partitioning scheme for irregular graphs, *Journal of Parallel and Distributed Computing* **48**(1), 96–129.

Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C. and Meltzer, P. S.: 2001, Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks., *Nat Med* **7**(6), 673–679.

Laiho, P., Kokko, A., Vanharanta, S., Salovaara, R., Sammalkorpi, H., Jarvinen, H., Mecklin, J., Karttunen, T. J., Tuppurainen, K., Davalos, V., Schwartz, S., Arango, D., Makinen, M. J. and Aaltonen, L. A.: 2007, Serrated carcinomas form a subclass of colorectal cancer with distinct molecular basis., *Oncogene* **26**(2), 312–320.

Lapointe, J., Li, C., Higgins, J. P., van de Rijn, M., Bair, E., Montgomery, K., Ferrari, M., Egevad, L., Rayford, W., Bergerheim, U., Ekman, P., DeMarzo, A. M., Tibshirani, R., Botstein, D., Brown, P. O., Brooks, J. D. and Pollack, J. R.: 2004, Gene expression profiling identifies clinically relevant subtypes of prostate cancer., *Proc Natl Acad Sci U S A* **101**(3), 811–816.

Liang, Y., Diehn, M., Watson, N., Bollen, A. W., Aldape, K. D., Nicholas, M. K., Lamborn, K. R., Berger, M. S., Botstein, D., Brown, P. O. and Israel, M. A.: 2005, Gene expression profiling reveals molecularly and clinically distinct subtypes of glioblastoma multiforme., *Proc Natl Acad Sci U S A* **102**(16), 5814–5819.

Liu, M., Jiang, X. and Kot, A. C.: 2009, A multi-prototype clustering algorithm, *Pattern Recognition* **42**(5), 689 – 698.

Lloyd, S.: 1982, Least squares quantization in pcm, *Information Theory, IEEE Transactions on* **28**(2), 129–137.

Mimaroglu, S. and Aksehirli, E.: 2011, Improving dbscan's execution time by using a pruning technique on bit vectors, *Pattern Recognition Letters* **32**(13), 1572 – 1580.

Mimaroglu, S. and Erdil, E.: 2010, Obtaining better quality final clustering by merging a collection of clusterings, *Bioinformatics* **26**(20), 2645–2646.

Mimaroglu, S. and Erdil, E.: 2011, Combining multiple clusterings using similarity graph, *Pattern Recognition* **44**(3), 694 – 703.

Mimaroglu, S. and Yagci, A.: 2009, A binary method for fast computation of inter and intra cluster similarities for combining multiple clusterings, pp. 452–456.

Mohammadi, M., Nikanjam, A. and Rahmani, A.: 2008, An evolutionary approach to clustering ensemble, **3**, 77–82.

Moussiades, L. and Vakali, A.: 2010, Clustering dense graphs: A web site graph paradigm, *Information Processing & Management* **46**(3), 247 – 267.

Ng, R. and Han, J.: 1994, Efficient and effective clustering methods for spatial data mining, *Proceedings of the International Conference on Very Large Data Bases*, Citeseer, pp. 144–144.

Nutt, C. L., Mani, D. R., Betensky, R. A., Tamayo, P., Cairncross, J. G., Ladd, C., Pohl, U., Hartmann, C., McLaughlin, M. E., Batchelor, T. T., Black, P. M., von Deimling, A., Pomeroy, S. L., Golub, T. R. and Louis, D. N.: 2003, Gene expression-based classification of malignant gliomas correlates better with survival than histological classification., *Cancer Res* **63**(7), 1602–1607.

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., Kim, J. Y. H., Goumnerova, L. C., Black, P. M., Lau, C., Allen, J. C., Zagzag, D., Olson, J. M., Curran, T., Wetmore, C., Biegel, J. A., Poggio, T., Mukherjee, S., Rifkin, R., Califano, A., Stolovitzky, G., Louis, D. N., Mesirov, J. P., Lander, E. S. and Golub, T. R.: 2002, Prediction of central nervous system embryonal tumour outcome based on gene expression., *Nature* **415**(6870), 436–442.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C. H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P., Poggio, T., Gerald, W., Loda, M., Lander, E. S. and Golub, T. R.: 2001, Multiclass cancer diagnosis using tumor gene expression signatures., *Proc Natl Acad Sci U S A* **98**(26), 15149–15154.

Rand, W.: 1971, Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical association* **66**(336), 846–850.

Risinger, J. I., Maxwell, G. L., Chandramouli, G. V. R., Jazaeri, A., Aprelikova, O., Patterson, T., Berchuck, A. and Barrett, J. C.: 2003, Microarray analysis reveals distinct gene expression profiles among different histologic types of endometrial cancer., *Cancer Res* **63**(1), 6–11.

Schikuta, E. and Erhart, M.: 1997, The bang-clustering system: grid-based data analysis, *Advances in Intelligent Data Analysis Reasoning about Data* pp. 513–524.

Sheikholeslami, G., Chatterjee, S. and Zhang, A.: 1998, Wavecluster: A multi-resolution clustering approach for very large spatial databases, *Proceedings of the International Conference on Very Large Data Bases*, Citeseer, pp. 428–439.

Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Kutok, J. L., Aguiar, R. C. T., Gaasenbeek, M., Angelo, M., Reich, M., Pinkus, G. S., Ray, T. S., Koval, M. A., Last, K. W., Norton, A., Lister, T. A., Mesirov, J., Neuberg, D. S., Lander, E. S., Aster, J. C. and Golub, T. R.: 2002, Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning., *Nat Med* **8**(1), 68–74.

Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D'Amico, A. V., Richie, J. P., Lander, E. S., Loda, M., Kantoff, P. W., Golub, T. R. and Sellers, W. R.: 2002, Gene expression correlates of clinical prostate cancer behavior., *Cancer Cell* **1**(2), 203–209.

Strehl, A. and Ghosh, J.: 2000, A scalable approach to balanced, high-dimensional clustering of market-baskets, *High Performance Computing—HiPC 2000* pp. 525–536.

Strehl, A. and Ghosh, J.: 2003, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, *The Journal of Machine Learning Research* **3**, 583–617.

Su, A. I., Welsh, J. B., Sapinoso, L. M., Kern, S. G., Dimitrov, P., Lapp, H., Schultz, P. G., Powell, S. M., Moskaluk, C. A., Frierson, H. F. and Hampton, G. M.: 2001, Molecular classification of human carcinomas by use of gene expression signatures., *Cancer Res* **61**(20), 7388–7393.

Tomlins, S. A., Mehra, R., Rhodes, D. R., Cao, X., Wang, L., Dhanasekaran, S. M., Kalyana-Sundaram, S., Wei, J. T., Rubin, M. A., Pienta, K. J., Shah, R. B. and Chinnaiyan, A. M.: 2007, Integrative molecular concept modeling of prostate cancer progression., *Nat Genet* **39**(1), 41–51.

Topchy, A., Jain, A. and Punch, W.: 2003, Combining multiple weak clusterings, pp. 331–338.

Tsai, C. and Sung, C.: 2010, Eidbscan: An extended improving dbscan algorithm with sampling techniques, *International Journal of Business Intelligence and Data Mining* **5**(1), 94–111.

Vega-Pons, S., Correa-Morris, J. and Ruiz-Shulcloper, J.: 2010, Weighted partition consensus via kernels, *Pattern Recognition* **43**(8), 2712 – 2724.

Ward, J.: 1963, Hierarchical grouping to optimize an objective function, *Journal of the American statistical association* **58**(301), 236–244.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J. A., Marks, J. R. and Nevins, J. R.: 2001, Predicting the clinical status of human breast cancer by using gene expression profiles., *Proc Natl Acad Sci U S A* **98**(20), 11462–11467.

Yeoh, E., Ross, M. E., Shurtleff, S. A., Williams, W. K., Patel, D., Mahfouz, R., Behm, F. G., Raimondi, S. C., Relling, M. V., Patel, A., Cheng, C., Campana, D., Wilkins, D., Zhou, X., Li, J., Liu, H., Pui, C., Evans, W. E., Naeve, C., Wong, L. and Downing, J. R.: 2002, Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling., *Cancer Cell* **1**(2), 133–143.

Zhou, A., Zhou, S., Cao, J., Fan, Y. and Hu, Y.: 2000, Approaches for scaling dbscan algorithm to large spatial databases, *Journal of computer science and technology* **15**(6), 509–526.

Other References

A. Asuncion, D. N.: 2007, UCI machine learning repository.
URL: *http://www.ics.uci.edu/∼mlearn/MLRepository.html*

APPENDICES

APPENDIX A. FIGURES

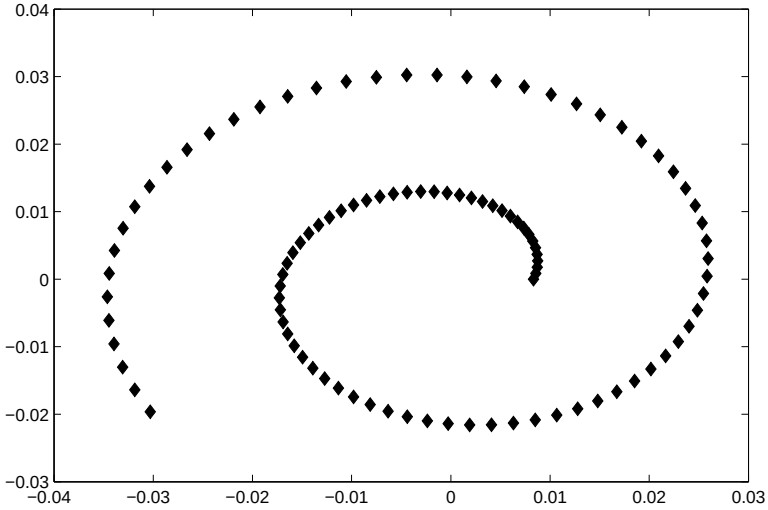

Figure A.1a 1-spiral data set

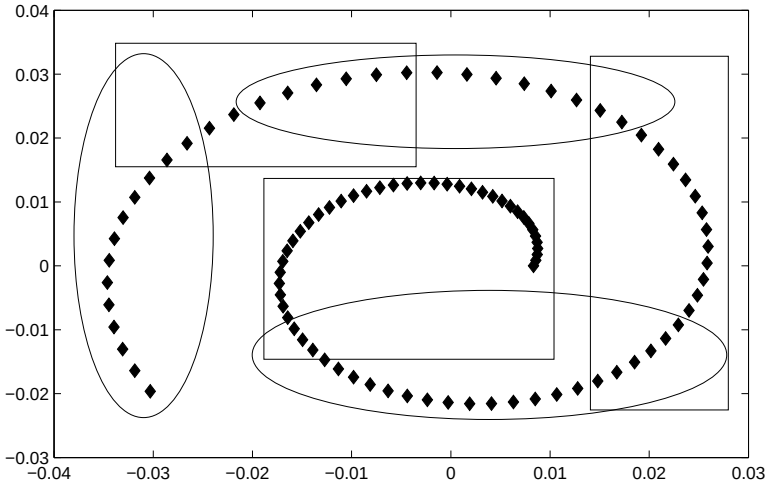

Figure A.1b Clustering on 1-spiral data set

Figure A.1: 1**-spiral data set and a clustering**

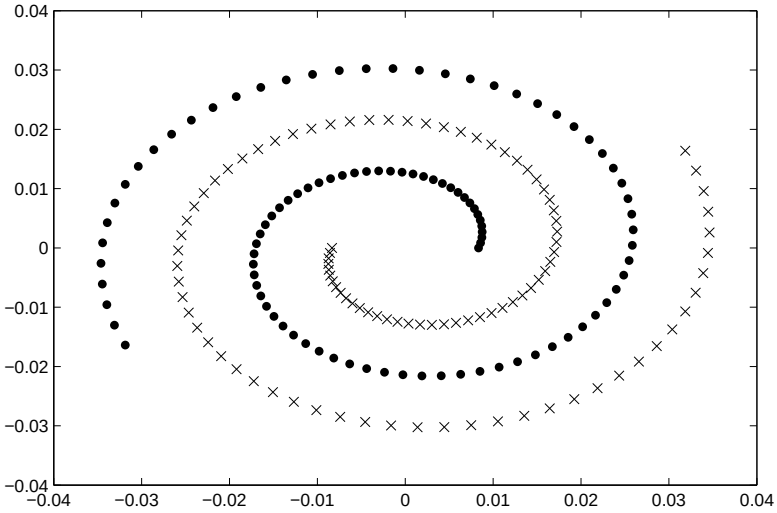

Figure A.2: 2**-spiral data set**

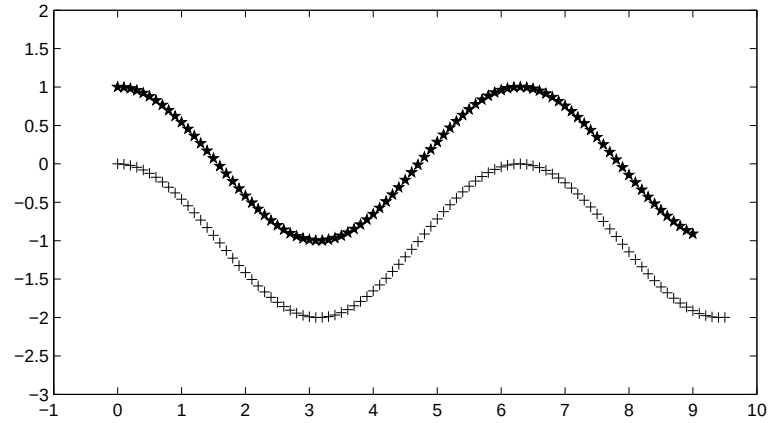

Figure A.3: 2**-curve data set**

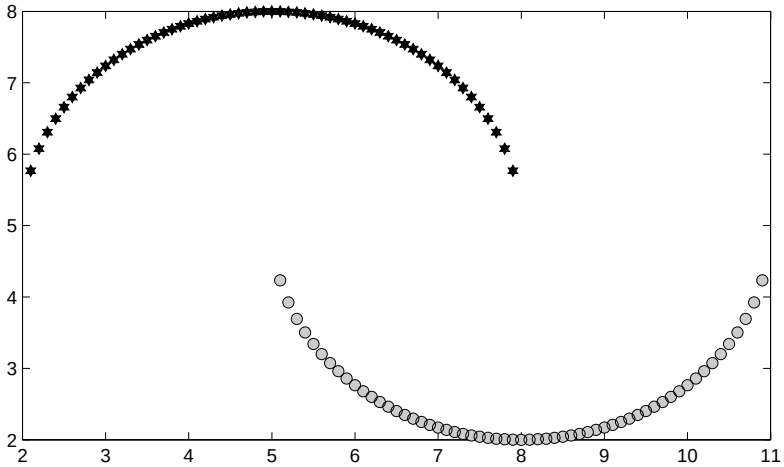

Figure A.4: 2**-half rings data set**

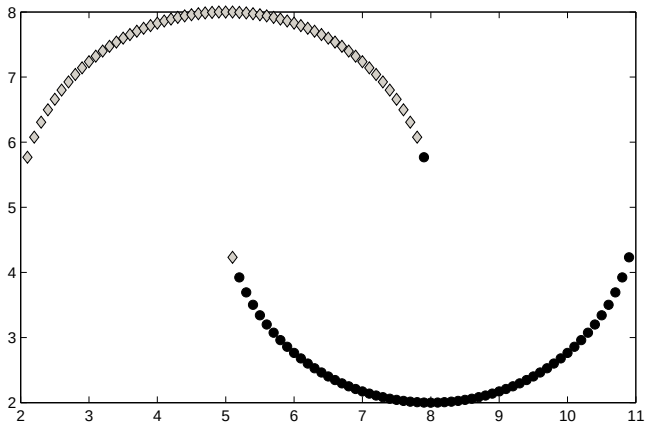

Figure A.5a $k = 2$

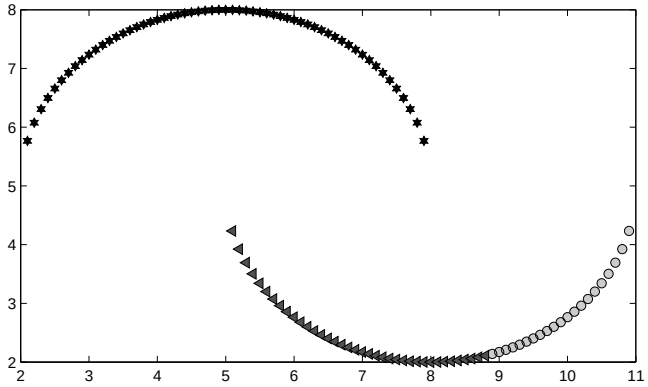

Figure A.5b $k = 3$

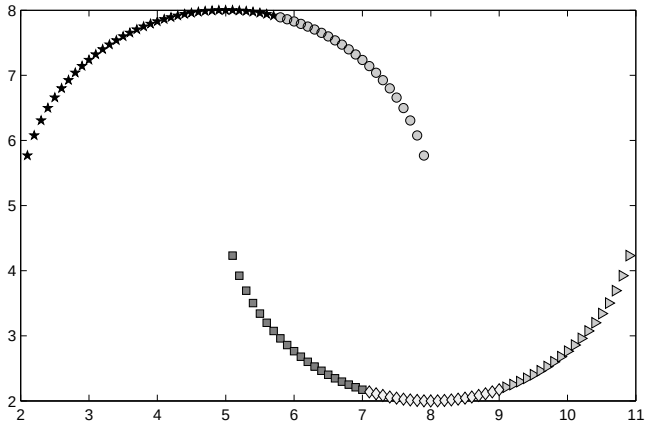

Figure A.5c $k = 5$

Figure A.5: Partitions of 2-half rings data set generated with k-means

APPENDIX B. TABLES

Table B.1: Properties of multiple clusterings on real and synthetically generated data sets

INPUT	$\lvert\Pi\rvert$	$\lvert\pi\rvert$	**Method**	**ARI**		
				min	**max**	**average**
1-**spiral, hand clustered**	2	3	manually	0.0	0.0	0.0
1-**spiral,** k**-means clustered**	4	[2, 3]	k-means	0.0	0.0	0.0
2-**spiral**	2	[2, 3]	manually	0.549	0.611	0.580
2-**half rings**	3	[2, 5]	k-means	0.414	0.933	0.699
2-**curve**	2	[2, 8]	k-means, randomly, manually	-0.005	0.301	0.103
2D2K	3	[2, 3]	k-means	0.656	0.788	0.744
8D5K	3	[3, 5]	k-means	0.547	0.739	0.654
Iris	3	[2, 3]	k-means	0.539	0.697	0.644
Glass	4	[6, 8]	k-means, randomly, manually	0.386	1.0	0.555
Breast Cancer	5	[2, 5]	k-means, randomly, manually	0.077	0.525	0.309
Image Segmentation	10	7	randomly	0.416	0.923	0.749
Syn5K	10	5	randomly	0.811	0.835	0.821

Table B.2: Properties of gene expression data sets

Data Set	Array Type	Tissue	Total samples	Num of classes	Total Genes	Selected # of Genes
Bladder carcinoma (Dyrskjot et al. 2003)	Affymetrix	Bladder	40	3	7129	1203
Breast Cancer (West et al. 2001)	Affymetrix	Breast	49	2	7129	1198
Breast-Colon tumors (Chowdary et al. 2006)	Affymetrix	Breast, Colon	104	2	22283	182
Carcinomas (Su et al. 2001)	Affymetrix	Multi-tissue	174	10	12533	1571
Central nervous system-1 (Pomeroy et al. 2002)	Affymetrix	Brain	34	2	7129	857
Central nervous system-2 (Pomeroy et al. 2002)	Affymetrix	Brain	42	5	7129	1379
Endometrial cancer (Risinger et al. 2003)	Double Channel	Endometrium	42	4	8872	1771
Glioblastoma multiforme (Liang et al. 2005)	Double Channel	Brain	37	3	24192	1411
Gliomagenesis (Bredel et al. 2005)	Double Channel	Brain	50	3	41472	1739
Gliomas-1 (Nutt et al. 2003)	Affymetrix	Brain	50	4	12625	1377
Gliomas-2 (Nutt et al. 2003)	Affymetrix	Brain	28	2	12625	1070
Gliomas-3 (Nutt et al. 2003)	Affymetrix	Brain	22	2	12625	1152
Hepatocellular carcinoma (Chen et al. 2002)	Double Channel	Liver	178	2	22699	85
Leukemia-1 (Yeoh et al. 2002)	Affymetrix	Bone Marrow	248	2	12625	2526
Leukemia-2 (Yeoh et al. 2002)	Affymetrix	Bone Marrow	248	6	4022	1095
Leukemia-3 (Armstrong et al. 2002)	Affymetrix	Blood	72	2	12582	1081
Leukemia-4 (Armstrong et al. 2002)	Affymetrix	Blood	72	3	12582	2194
Leukemia-5 (Golub et al. 1999)	Affymetrix	Bone Marrow	72	2	7129	1877
Leukemia-6 (Golub et al. 1999)	Affymetrix	Bone Marrow	72	3	7129	1877
Lung tumor-1 (Bhattacharjee et al. 2001)	Affymetrix	Lung	203	5	12600	1543
Lung tumor-2 (Garber et al. 2001)	Double Channel	Lung	66	4	24192	4553
Lymphoma-1 (Alizadeh et al. 2000)	Double Channel	Blood	42	2	4022	1095
Lymphoma-2 (Alizadeh et al. 2000)	Double Channel	Blood	62	3	4022	2093
Lymphoma-3 (Shipp et al. 2002)	Affymetrix	Blood	77	2	7129	798
Melanoma (Bittner et al. 2000)	Double Channel	Skin	38	2	8067	2201
Mesothelioma (Gordon et al. 2002)	Affymetrix	Lung	181	2	12533	1626
Multi-tissue (Ramaswamy et al. 2001)	Affymetrix	Multi-tissue	190	14	16063	1363
Prostate cancer-1 (Tomlins et al. 2007)	Double Channel	Prostate	104	5	20000	2315
Prostate cancer-2 (Tomlins et al. 2007)	Double Channel	Prostate	92	4	20000	1288
Prostate cancer-3 (Lapointe et al. 2004)	Double Channel	Prostate	69	3	42640	1625
Prostate cancer-4 (Lapointe et al. 2004)	Double Channel	Prostate	110	4	42640	2496
Prostate cancer-5 (Singh et al. 2002)	Affymetrix	Prostate	102	2	12600	339
Round blue-cell tumor (Khan et al. 2001)	Double Channel	Multi-tissue	83	4	6567	1069
Serrated carcinomas (Laiho et al. 2007)	Affymetrix	Colon	37	2	22883	2202

Table B.3: Properties of multiple clusterings on gene expression data sets

Data Set	Method	Features	$\|\pi\|$	$\|\Pi\|$	ARI		
					Min	Max	Average
Bladder carcinoma	k-means	25% - 50%	2 - 6	10	0.18	0.64	0.39
Breast Cancer	k-means	25% - 50%	2 - 7	10	0.08	0.42	0.25
Breast-Colon tumors	k-means	25% - 50%	2 - 10	10	0.11	0.92	0.43
Carcinomas	k-means	25% - 50%	2 - 13	10	0.10	0.63	0.42
Central nervous system-1	manual	N/A	2 - 4	10	-0.04	0.14	0.05
Central nervous system-2	k-means	25% - 50%	2 - 6	10	0.23	0.50	0.38
Endometrial cancer	manual, random	N/A	4 - 5	10	0.0	0.31	0.12
Glioblastoma multiforme	k-means	75% - 85%	2 - 6	10	-0.03	0.46	0.18
Gliomagenesis	k-means	25% - 50%	2 - 7	10	0.11	0.49	0.28
Gliomas-1	manual	N/A	4 - 6	10	-0.02	0.11	0.06
Gliomas-2	manual, random	N/A	2 - 5	10	-0.04	0.02	-0.02
Gliomas-3	manual	N/A	2 - 3	10	-0.05	0.17	0.04
Hepatocellular carcinoma	k-means	75% - 85%	2 - 13	10	0.10	0.70	0.40
Leukemia-1	k-means	75% - 85%	2 - 15	10	0.10	0.32	0.18
Leukemia-2	k-means	25% - 50%	2 - 15	10	0.14	0.23	0.20
Leukemia-3	manual	N/A	2 - 5	10	0.10	0.46	0.27
Leukemia-4	k-means	75% - 85%	3 - 8	10	0.42	0.92	0.59
Leukemia-5	k-means	25% - 50%	2 - 8	10	0.15	0.89	0.45
Leukemia-6	k-means	25% - 50%	2 - 8	10	0.18	0.84	0.47
Lung tumor-1	k-means	25% - 50%	3 - 14	10	0.10	0.24	0.18
Lung tumor-2	k-means	25% - 50%	2 - 8	10	0.08	0.32	0.19
Lymphoma-1	k-means	25% - 50%	2 - 6	10	0.02	0.43	0.17
Lymphoma-2	k-means	25% - 50%	3 - 7	10	0.20	0.52	0.33
Lymphoma-3	k-means	25% - 50%	2 - 8	10	-0.01	0.32	0.11
Melanoma	manual, random	N/A	2	10	-0.02	0.28	0.11
Mesothelioma	k-means	25% - 50%	2 - 13	10	0.07	0.75	0.25
Multi-tissue	k-means	25% - 50%	2 - 10	10	0.15	0.41	0.31
Prostate cancer-1	manual	N/A	5 - 7	10	0.14	0.37	0.26
Prostate cancer-2	manual	N/A	4 - 6	10	0.15	0.34	0.23
Prostate cancer-3	manual	N/A	4 - 7	10	0.02	0.22	0.08
Prostate cancer-4	manual	N/A	5 - 6	10	0.08	0.39	0.20
Prostate cancer-5	k-means	25% - 50%	2 - 10	10	0.02	0.23	0.10
Round blue-cell tumor	k-means	25% - 50%	2 - 9	10	0.10	0.90	0.49
Serrated carcinomas	manual	N/A	2 - 6	10	-0.03	0.09	0.02

Table B.4: COMUSA on 1-spiral data set

INPUT	relaxation	ARI
1-**spiral, hand clustered**	0%	**1.0**
1-**spiral,** k-**means clustered**	34%	**1.0**

Table B.5: Cluster validity results on 2-spiral, 2-half rings, 2-Curve, and Syn5K data sets

INPUT	**COMUSA**			**PMETIS**	**KMETIS**	**HMETIS**	**MCLA**	**EAC**
	relaxation	ARI	k	ARI	ARI	ARI	ARI	ARI
2-**spiral**	%34	**1.0**	2	**1.0**	**1.0**	-0.005	**1.0**	0.0
2-**half rings**	%34	**1.0**	2	0.966	0.966	-0.008	**1.0**	**1.0**
2-**curve**	%50	**1.0**	2	0.057	0.057	-0.004	0.086	**1.0**
Syn5K	0%	0.301	3	0.488	0.480	0.317	0.611	0.482
	15%	**1.0**	5	**1.0**	**1.0**	0.426	**1.0**	**1.0**
	25%	0.999	7	0.578	0.556	0.607	0.953	0.999

Table B.6: Cluster validity results on 2D2K and 8D5K

INPUT	**COMUSA**	
	relaxation	ECS+ICS
2D2K	%0	**54.27**
8D5K	%0	**238.68**

INPUT		**PMETIS**	**KMETIS**	**HMETIS**	**MCLA**	**EAC**
	k	ECS+ICS	ECS+ICS	ECS+ICS	ECS+ICS	ECS+ICS
2D2K	2	26.80	26.93	12.22	26.93	26.93
	3	33.06	32.39	36.40	41.07	39.38
8D5K	5	214.18	214.18	57.98	219.87	192.31
	6	211.64	213.42	157.59	219.87	**238.68**

Table B.7: Cluster validity results on Iris, Glass, Breast Cancer, and Image Segmentation data sets

INPUT	**COMUSA**	
	relaxation	ARI
Iris	0%	0.676
	50%	0.698
Glass	0%	0.308
	25%	0.403
	34%	**0.964**
Breast Cancer	0%	0.156
	17%	0.301
	20%	0.604
	25%	**1.0**
Image Segmentation	0%	0.350
	12%	0.932
	13%	0.831

INPUT		**PMETIS**	**KMETIS**	**HMETIS**	**MCLA**	**EAC**
	k	ARI	ARI	ARI	ARI	ARI
Iris	3	0.691	0.688	0.096	**0.711**	**0.711**
	4	0.412	0.368	0.036	0.662	0.690
Glass	6	0.4740	0.4698	0.1568	0.9633	0.8041
	8	0.3835	0.4151	0.0063	0.6439	0.7862
	22	0.2026	0.1971	0.0654	0.2898	0.6404
	24	0.1892	0.1705	0.0615	0.2847	0.4292
Breast Cancer	2	0.3597	0.3942	0.0024	0.5778	0.8322
	6	0.2174	0.2039	0.1311	0.2967	0.8981
	11	0.1189	0.1122	0.0684	0.3268	0.5463
	16	0.0836	0.0828	0.0188	0.4547	0.5316
Image	7	0.987	**0.988**	0.535	0.985	0.840
Segmentation	9	0.633	0.639	0.352	0.938	0.837
	11	0.574	0.539	0.354	0.969	0.838

Table B.8: Cluster validity results on gene expression data sets

Input	COMUSA		LCE	
	relaxation	ARI	k	ARI
Bladder carcinoma	0%	0.155	3	0.410
	29%	0.619		
Breast Cancer	0%	0.072	2	0.560
	15%	0.560		
Breast-Colon tumors	0%	0.038	2	0.920
	43%	0.924		
Carcinomas	0%	0.234	10	0.570
	12%	0.501		
Central nervous system-1	0%	0.060	2	-0.110
	29%	0.151		
Central nervous system-2	0%	0.391	5	0.610
	25%	0.507		
Endometrial cancer	0%	0.062	4	0.238
	29%	0.262		
Glioblastoma multiforme	0%	0.110	3	0.160
	12%	0.264		
Gliomagenesis	0%	0.100	3	0.370
	25%	0.470		
Gliomas-1	0%	0.032	4	0.057
	15%	0.097		
Gliomas-2	0%	-0.012	2	-0.028
	34%	0.002		
Gliomas-3	0%	-0.012	2	0.170
	13%	0.043		
Hepatocellular carcinoma	0%	0.064	2	0.640
	13%	0.641		
Leukemia-1	0%	0.021	2	0.960
	29%	0.960		
Leukemia-2	0%	0.097	6	0.370
	15%	0.262		
Leukemia-3	0%	0.131	2	0.268
	25%	0.514		
Leukemia-4	0%	0.219	3	0.920
	29%	0.816		
Leukemia-5	0%	0.084	2	0.840
	12%	0.500		
Leukemia-6	0%	0.110	3	0.790
	13%	0.616		
Lung tumor-1	0%	0.036	5	0.320
	17%	0.504		
Lung tumor-2	0%	0.048	4	0.150
	12%	0.240		
Lymphoma-1	0%	0.032	2	0.370
	15%	0.213		
Lymphoma-2	0%	0.136	3	0.380
	34%	0.893		
Lymphoma-3	0%	0.065	2	0.250
	17%	0.134		

Continued on Next Page…

Table B.8 Cluster validity results on gene expression data sets – Continued

Input	COMUSA		LCE	
	relaxation	ARI	k	ARI
Melanoma	**0%** **12%**	**0.119** **0.134**	**2**	**-0.002**
Mesothelioma	**0%** **29%**	**0.021** **0.719**	**2**	**0.780**
Multi-tissue	**0%** **12%**	**0.299** **0.497**	**14**	**0.440**
Prostate cancer-1	**0%** **15%**	**0.141** **0.304**	**5**	**0.291**
Prostate cancer-2	**0%** **15%**	**0.141** **0.304**	**4**	**0.250**
Prostate cancer-3	**0%** **13%**	**0.059** **0.127**	**3**	**0.352**
Prostate cancer-4	**0%** **13%**	**0.082** **0.194**	**4**	**0.122**
Prostate cancer-5	**0%** **13%**	**0.029** **0.137**	**2**	**0.020**
Round blue-cell tumor	**0%** **50%**	**0.263** **0.836**	**4**	**0.890**
Serrated carcinomas	**0%** **25%**	**0.040** **0.055**	**2**	**-0.001**

Table B.9: Execution time results (ms)

INPUT	COMUSA	PMETIS	KMETIS	HMETIS	MCLA	EAC
2-**spiral**	1.1	4.0	4.0	1.0	201.0	148.0
2-**half rings**	2.3	2.0	2.0	1.0	196.0	114.0
2-**curve**	1.1	2.0	3.0	3.0	203.0	150.0
2D2K	25.1	48.5	46.0	2.0	197.5	1203.5
8D5K	21.7	43.0	35.5	2.5	195.5	1207
Iris	3.6	3.0	2.5	2.5	199.5	117.5
Glass	1.3	5.5	7.3	34.3	198.5	149.5
Breast Cancer	2.1	9.5	12	12	198.8	601
Image Segmentation	7083.6	251.0	206.0	2	922.3	15466.3
Syn5K	71769.0	1371.0	1159.0	6	372.3	30774.6

Table B.10: Number of clusters

1-**spiral, hand clustered**	**1**	**1**
1-**spiral,** k-**means clustered**	**1**	**1**
2-**spiral**	**2**	**2**
2-**half rings**	**2**	**2**
2-**curve**	**2**	**2**
2D2K	**2**	**2**
8D5K	**6**	**5**
Iris	**3**	**3**
Glass	**6**	**6**
Breast Cancer	**2**	**2**
Image Segmentation	**9**	**7**
Syn5K	**5**	**5**
Bladder carcinoma	**3**	**3**
Breast Cancer	**2**	**2**
Breast-Colon tumors	**2**	**2**
Carcinomas	**9**	**10**
Central nervous system-1	**3**	**2**
Central nervous system-2	**6**	**5**
Endometrial cancer	**4**	**4**
Glioblastoma multiforme	**3**	**3**
Gliomagenesis	**4**	**3**
Gliomas-1	**4**	**4**
Gliomas-2	**2**	**2**
Gliomas-3	**2**	**2**
Hepatocellular carcinoma	**2**	**2**
Leukemia-1	**2**	**2**
Leukemia-2	**8**	**6**
Leukemia-3	**3**	**2**
Leukemia-4	**3**	**3**
Leukemia-5	**2**	**2**
Leukemia-6	**3**	**3**
Lung tumor-1	**5**	**5**
Lung tumor-2	**5**	**4**
Lymphoma-1	**2**	**2**
Lymphoma-2	**3**	**3**
Lymphoma-3	**2**	**2**
Melanoma	**2**	**2**
Mesothelioma	**3**	**2**
Multi-tissue	**13**	**14**
Prostate cancer-1	**4**	**5**
Prostate cancer-2	**5**	**4**
Prostate cancer-3	**4**	**3**
Prostate cancer-4	**3**	**4**
Prostate cancer-5	**2**	**2**
Round blue-cell tumor	**4**	**4**
Serrated carcinomas	**2**	**2**

CV

Name Surname	:	Ertunç ERDİL
Address	:	Bahçeşehir Üniversitesi Mühendislik Fakültesi Çırağan Caddesi 34353 Beşiktaş/İSTANBUL
Date and Place of Birth	:	18.10.1987 İSTANBUL
Languages	:	Turkish (native), English (fluent)
B.S.	:	Ege University
M.S.	:	Bahçeşehir University
Institute	:	The Graduate School of Natural and Applied Sciences
Program	:	Computer Engineering
Publications	:	Selim Mimaroglu, Ertunc Erdil, ASOD: Arbitrary Shape Object Detection, in Engineering Applications of Artificial Intelligence, Elsevier 2011 Selim Mimaroglu, Ertunc Erdil, Combining Multiple Clusterings Using Similarity Graph, in Pattern Recognition, Elsevier 2011 Selim Mimaroglu, Ertunc Erdil, Obtaining Better Quality Final Clustering by Merging a Collection of Clusterings, in Bioinformatics, Oxford University Press 2010
Work Experience	:	Bahcesehir University Computer Engineering Department *Research and Teaching Assistant* (Istanbul, 2010 - today)