WEIGHTING POLICIES FOR ROBUST UNSUPERVISED ENSEMBLE LEARNING

by

RAMAZAN UNLU
B.S. Istanbul University, 2010
M.S. University of Pittsburgh, 2014

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosphy
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando,Florida

Summer Term
2017

Major Professors: Petros Xanthopoulos
Qipeng Phil Zheng

ABSTRACT

The unsupervised ensemble learning, or consensus clustering, consists of finding the optimal combination strategy of individual partitions that is robust in comparison to the selection of an algorithmic clustering pool. Despite its strong properties, this approach assigns the same weight to the contribution of each clustering to the final solution. We propose a weighting policy for this problem that is based on internal clustering quality measures and compare against other modern approaches. Results on publicly available datasets show that weights can significantly improve the accuracy performance while retaining the robust properties. Since the issue of determining an appropriate number of clusters, which is a primary input for many clustering methods is one of the significant challenges, we have used the same methodology to predict correct or the most suitable number of clusters as well. Among various methods, using internal validity indexes in conjunction with a suitable algorithm is one of the most popular way to determine the appropriate number of cluster. Thus, we use weighted consensus clustering along with four different indexes which are Silhouette (SH), Calinski-Harabasz (CH), Davies-Bouldin (DB), and Consensus (CI) indexes. Our experiment indicates that weighted consensus clustering together with chosen indexes is a useful method to determine right or the most appropriate number of clusters in comparison to individual clustering methods (e.g., k-means) and consensus clustering. Lastly, to decrease the variance of proposed weighted consensus clustering, we borrow the idea of Markowitz portfolio theory and implement its core idea to clustering domain. We aim to optimize the combination of individual clustering methods to minimize the variance of clustering accuracy. This is a new weighting policy to produce partition with a lower variance which might be crucial for a decision maker. Our study shows that using the idea of Markowitz portfolio theory will create a partition with a less variation in comparison to traditional consensus clustering and proposed weighted consensus clustering.

To my parents, sister, lovely wife and my baby being with us soon.

ACKNOWLEDGMENTS

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of my doctoral study. I am thankful for their aspiring guidance, invaluably constructive criticism and friendy advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

First and foremost, I am most grateful to Dr. Petros Xanthopoulos for his guidance and support throughout the time of my dissertation research. Even the half of this work would not have been possible without him. His positive thinking in any case encouraged me to attempt again and again whenever I feel I stuck.

And last but not least, my special thanks to my parents, sister, and lovely wife. I always find their priceless support and encouragement during the stressful times. I am very lucky to have all of you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

A Brief Overview of Data Mining

Data mining (DM) is one of the most notable research areas in the last decades. DM is an interdisciplinary area of an intersection of AI, machine learning, and statistics. One of the earliest studies of the DM, which highlights some of its distinctive characteristics, is proposed by [Fayyad et al., 1996], who define it as ”the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.”. In general, the process of extraction implicit, hidden , and potentially useful knowledge from data is a well-accepted definition of DM.

With the growing use of computers and data storage technology, there exist a great amount of data being produced by different systems [Kantardzic, 2011]. Data can be defined as a set of qualitative or quantitative variables such as facts, numbers, or text that descript the things. For DM, the standard structure of a data is a collection of samples in which measurements named features are specified, and these features are obtained in many cases. If we consider that a sample is represented by a multidimensional vector, each dimension can be considered as one feature of the sample. In other words, we can say that features are some values that represent the specific characteristic of a sample [Kantardzic, 2011]. In the tabular form of data, columns represent features of samples and rows are values of these features for a specific sample as shown in Table 1.1.

In this example, age, work class, education and so on are the features of each sample, each row is one sample (i.e., there are 11 samples and each sample represent a person), and the number or string in the table is the values of a particular feature of a specified sample. Original data of this example can be found in *http://archive.ics.uci.edu/ml/datasets/Adult*, here we just give some samples and feature for illustration purpose.

As we see in Table 1.1, there are different types of features which can be categorized as follows.

Table 1.1: Tabular form of Data.

Age	Workclass	Education	Occupation	Sex	Capital-gain	Hours	Country
39	State-gov	Bachelors	Adm-clerical	Male	2174	40	USA
50	Self	Bachelors	Exec-managerial	Male	0	13	USA
38	Private	HS-grad	Handlers-cleaners	Male	0	40	USA
53	Private	11th	Handlers-cleaners	Male	0	40	USA
28	Private	Bachelors	Prof-specialty	Female	0	40	Cuba
37	Private	Masters	Exec-managerial	Female	0	40	USA
49	Private	9th	Other-service	Female	0	16	Jamaica
52	Self	HS-grad	Exec-managerial	Male	0	45	USA
31	Private	Masters	Prof-specialty	Female	14084	50	USA
42	Private	Bachelors	Exec-managerial	Male	5178	40	USA
37	Private	Some-college	Exec-managerial	Male	0	80	USA

1. Quantitive features
   (a) Continues values (e.g., real numbers)
   (b) Discrete values (e.g., binary numbers)
   (c) Interval values (e.g., $0 \leq x \leq 100$)
2. Qualitative features
   (a) Nominal or unordered values (e.g., gender is male or female)
   (b) Ordinal values (e.g., risk levels are high, medium, and low)

On the other hand, the data can be categorized as labeled and unlabeled data from DM perspective. Labeled data refers a set of samples or cases with known true classes, and unlabeled data is a set of samples or cases without known true classes. For example, in the given example in Table 1.1, we are not given true outputs. The true outputs can be, for example, people those have the annual

income more or less than $100.000. In general, we need to select an appropriate DM method to apply based on labeled or unlabeled data we have. It might be crucial to pick a suitable algorithm because it might not be effective to use a method developed for labeled data to mine unlabeled data.

In practice, DM tasks can be categorized as predictive and descriptive tasks [Nisbet et al., 2009]. Predictive models allow one to predict the value of a sample based on other existing information (e.g., values of features) [Hand et al., 2001]. For example, fraud detection to predict whether a transaction is a fraud or not [Fawcett and Provost, 1997]. Descriptive models, on the other hand, attempt to find some specific patterns describing the data and can be interpreted by humans [Kantardzic, 2011]. Customer segmentation can be given as an example of descriptive tasks. It works based on distinguishing customers based on their similarities and differences [Chen et al., 2006]. The goal of predictive and descriptive methods can vary across users and needs. And, it is achieved by using data mining techniques. There are various data mining techniques have been proposed and can be seen in different data mining studies such as [Ngai et al., 2009, Kantardzic, 2011, Bhojani and Bhatt, 2016] . We explain some of them as follows:

- **Classification** : It is one of the most commonly used models in DM that assigns each sample in the dataset into target categories or classes. The goal of a classification model is to maximize the number of samples that are accurately assigned. For example, a classification model could be used to predict future customer behaviors by classifying recorded data samples into a number of predefined classes based on certain features [Ahmed, 2004].

- **Clustering**: A common descriptive task that partitions a heterogeneous population into a number of more homogenous groups [Barlow, 1989, Jain et al., 1999]. By contrast with supervised learning, there is no explicit known true output. Moreover, since there are no

predefined clusters, the number of clusters should be determined.

- **Association Rules**: Finding a local model identify relationships/dependencies among a set of samples in a database [Agrawal et al., 1993]. Market basket analysis and cross-selling programs can be given as typical examples for which association rules is usually used [Ngai et al., 2009].

- **Regression**: One of the widely used predictive learning methods. It can be described as a kind of statistical estimation technique learning a predictive function that maps each data sample to a real value [Giraud-Carrier and Povel, 2003].

- **Summarization** : An additional descriptive task for finding a reliable description of a dataset. Tabulating the mean and standard deviations is an example of simple summarization methods are often used for data analysis, data visualization and automated report generation [Chandola and Kumar, 2007].

- **Sequence Discovery**: It is one of the DM techniques used to identify associations or patterns over time in a sequence database [Mabroukeh and Ezeife, 2010].

DM is not merely to apply a method, but it is a collection of a set of iterative processes in practice. Through DM process, one can collect data, examine it using different methods, decides to look at it from a different perspective, and then goes back to the beginning. Several studies such as [Jun Lee and Siau, 2001, Kantardzic, 2011, Fayyad et al., 1996, Weiss, 2005, Tomar and Agarwal, 2013] provide general entire process of DM. We provide one as show in Figure 1.1 inspired by the one suggested in [Kantardzic, 2011].

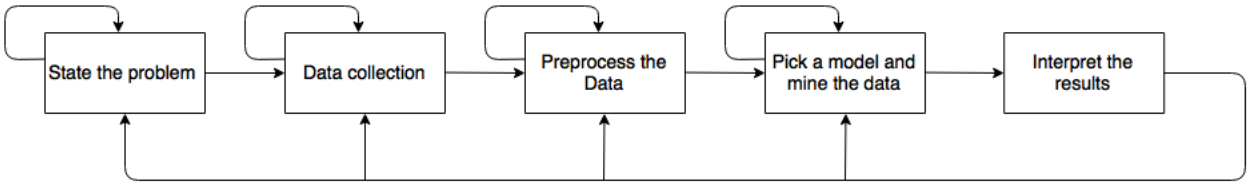

Figure 1.1: The data mining process

Today, across a wide variety of fields, extensive data are being gathered and stored at a breakneck pace. Having a real data without actual output is computationally much cheaper than data with the known output. Therefore, unsupervised learning -also called clustering- has become one of the important methods used to deal with unlabeled data. Through this study, we interchangeably use both unsupervised learning and clustering terms. This work will help to produce more robust performance than existing clustering methods. In particular, we study a novel unsupervised ensemble learning-also called consensus clustering- to deal with the deficiency of traditional unsupervised ensemble learning. As it will be discussed later, we also propose an application of proposed method to determine a suitable number of clusters and a study of the extension of proposed method to improve its performance concerning the variance of accuracy.

A Brief Overview of Unsupervised Learning

Clustering is one of the most widely used DM methods in different domains such as information retrieval and text mining [Jain et al., 1999], spatial database applications [Sander et al., 1998], sequence and heterogeneous data analysis [Cades et al., 2001], web data analysis [Srivastava et al., 2000], bioinformatics [de Hoon et al., 2004] and many others. In clustering, there are no labeled

data available. Therefore, the goal of clustering is a division of unlabeled data into groups of similar objects [Berkhin, 2006]. Objects in the same group are considered as similar to each other and dissimilar to objects in other groups. An example of clustering is illustrated in Figure 1.2, here points belonging to the same cluster are shown with the same symbol.

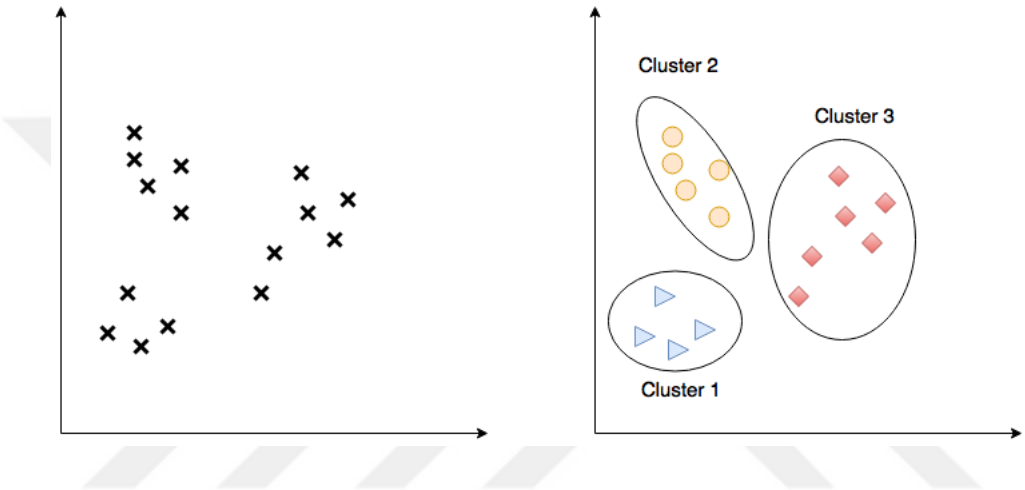

Figure 1.2: An example of clustering

Furthermore, for a given data set $X = \{(x_i)\}_{i=1}^{N}$ where $x_i \in \mathbb{R}^n$, N and n number of samples and features, respectively, clustering methods try to find k-clusters of X, $p = \{p_1, p_2, \cdots, p_k\}$ where $k < N$, such that:

1. $p_i \neq \emptyset$ for $i = 1, \cdots, k$
2. $\bigcup_{i=1}^{k} p_i = X$
3. $p_i \cap p_j = \emptyset$ for $i, j = 1, \cdots, k$

Through this clustering process, clusters are created based on dissimilarities and similarities between samples. Those dissimilarities and similarities are assessed based on the feature values

describing the objects and are relevant to the purpose of the study, to domain-specific assumptions and prior knowledge of the problem [Grira et al., 2004]. Since the similarity is an essential part of a cluster, a measure of the similarity between two objects is very crucial in clustering algorithms. This action must be chosen very carefully because the quality of a clustering model depends on this decision. Instead of using similarity measure, the dissimilarity between two samples are commonly used as well. For the dissimilarity metrics, a distance measure defined on the feature space such as Euclidean distance, Minkowski distance, and City-block distance can be given as examples [Kantardzic, 2011].

The standard process of clustering can be divided into the several steps. A brief overview of those necessary steps of a clustering model is given as follows and are depicted in Figure 1.3 [Xu and Wunsch, 2005].

- **Feature selection or extraction**: Extract and select the most useful and representative features from the raw data. While selection can be defined as to choose distinguishing features, extraction is to transform original features to create more useful features. Both of them might be critical for generating efficient clustering applications.
- **Clustering method selection or design**: Clustering algorithm should be chosen and designed according to the problem. Due to the fact that each clustering algorithms have pros and cons, one need to consider different parameter such as problem definition, data structure, and feature type to apply the suitable algorithm.
- **Cluster evaluation**: Clustering solution and goodness of algorithm should be evaluated. As different from classification problem, there is no true class information. Therefore, one need to use some other methods for evaluation purpose (e.g., cluster validity measures.)
- **Results interpretation**: After validating the result of the clustering algorithm, the solution

of the problem should be clearly interpreted and be given a practical explanation.

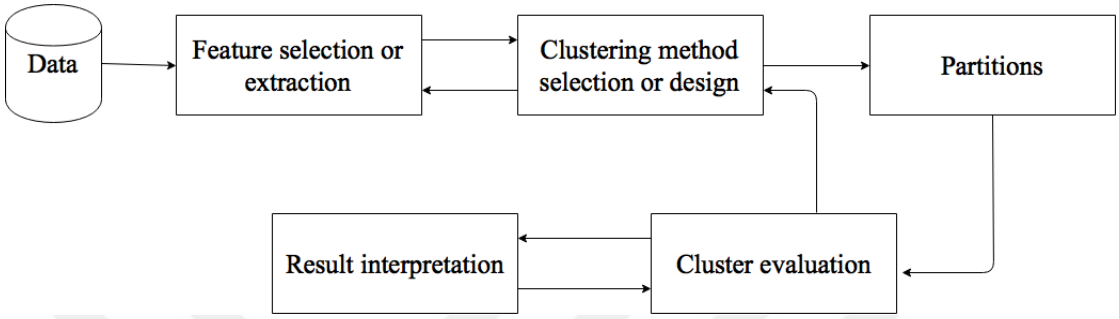

Figure 1.3: Clustering process

On the other hand, several taxonomies of clustering methods were proposed by [Xu and Wunsch, 2005, Xu and Tian, 2015, Nayak et al., 2015]. It is not easy to give the strong diversity of clustering methods because of different starting point and criteria. A rough but widely agreed categorization of clustering methods is to classify them as hierarchical clustering and partitional clustering, based on the properties of clusters generated [Xu and Wunsch, 2005]. However, we put forward the detailed taxonomy listed below in Table 1.2 inspired by the one suggested in [Xu and Tian, 2015]

In this study, we do not give the details of algorithms categorized in Table 1.2. We can refer the reader to [Xu and Tian, 2015] for a detailed explanation of these clustering algorithms. However, we give a brief introduction about ensemble based clustering algorithms which is the core algorithm of our proposed methods in the following section. Detailed discussion will be introduced in Chapter 3.

Table 1.2: Tradiational and Modern algorithms

Tradiational Algorithms		Modern Algorithms	
Based on	Typical Algorithms	Based on	Typical Algorithms
Partition	K-means,K-medoids PAM, CLARA CLARANS	Kernel	kernel K-means kernel SOM kernel FCM, SVC MMC, MKC
Hierarchy	BIRCH, CURE ROCK, Chameleon	Ensemble	CSPA, HGPA, MCLA VM, HCE LAC, WPCK, sCSPA sMCLA, sHBGPA
Fuzzy Theory	FCM,FCS,MM	Swarm Intelligence	ACO_based(LF) PSO_based SFLA_based, ABC_based
Distribution	DBCLASD, GMM	Qantum Theory	QC, DQC
Density	DBSCAN, OPTICS Mean-shift	Spectral graph theory	SM, NJW
Graph Theory	CLICK, MST	Affinity propagation	AP
Grid	STING, CLIQUE	Density and distance	DD
Fractal Theory	FC	Spatial data	DBSCAN, STING Wavecluster CLARANS
Model	COBWEB, GMM SOM, ART	Data Stream	STREAM, CluStream HPStream, DenStream
		Large-scale data	K-means, BIRCH, CLARA CUREDBSCAN DENCLUE, Wavecluster, FC

Clustering Algorithms Based on Ensemble

Clustering algorithms based on ensemble called unsupervised ensemble learning or consensus clustering can be considered as a modern clustering algorithm. Clustering results are prone to being

diverse across the algorithm, and each algorithm might work better for a particular dataset. We hypothetically illustrate this diversity by a toy example in Figure 1.4. In this figure, samples are in the same group represented by the same symbol. As shown, different clustering methods might give us different partitions of the data, and they can even produce the different number of clusters because of given the diverse objectives and methodological foundations [Haghtalab et al., 2015].

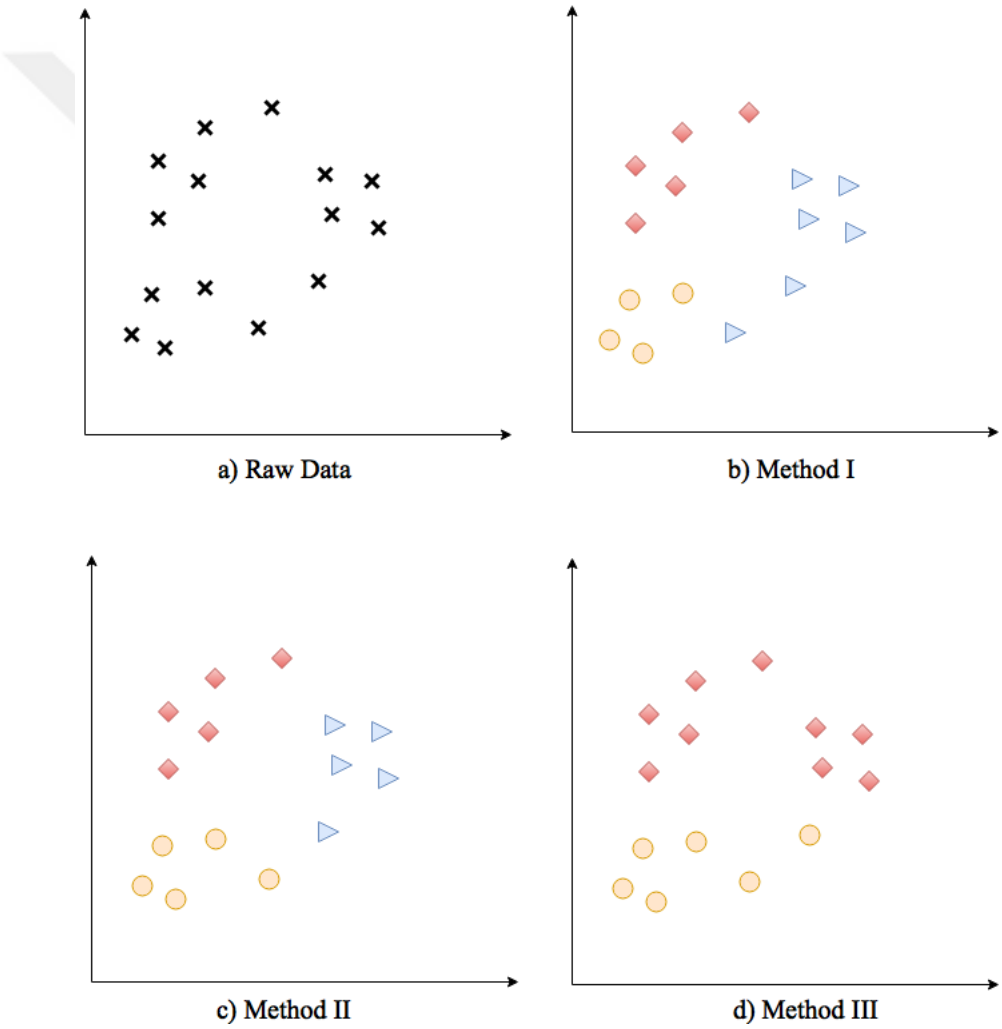

Figure 1.4: Schema of consensus clustering. *a* represents the raw data without knowing true classes. *b,c*, and *d* illustrate various partition of the data produced by different methods.

As we will discuss later, to deal with the potential variation of clustering methods, one can use consensus clustering. The core idea of consensus clustering is to combine good characteristics of different partitions to create a better clustering model. As the simple logic of process is shown in Figure 1.5 , different partitions $(P_1, P_2, \cdots, P_q)$ need to be somehow produced and combined to create optimum partition (P^*).

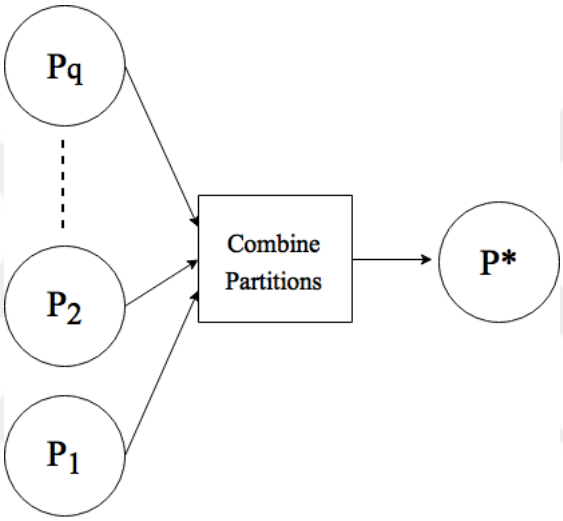

Figure 1.5: Schema of consensus clustering

The analysis of consensus clustering is summarized under the title of modern clustering methods in [Xu and Tian, 2015] as follows:

- Time complexity of this kind of algorithms depends on the algorithm chosen to combine its results.
- Consensus clustering can produce robust, scalable, consistent partition and can take the advantages of individual algorithms used.
- They have existing deficiencies of the design of the function which is used to combine results of individual algorithms.

Through this study, we work to enhance this type of algorithm and develop some useful extensions of existing methods. In Chapter 3, we give a detailed analysis of popular approaches of consensus clustering.

Dissertation Goal and Structure

Despite the fact that consensus clustering gives more robust and consistent results than individual clustering methods [Deodhar and Ghosh, 2006, Kuncheva et al., 2006, Vega-Pons and Ruiz-Shulcloper, 2011, Lancichinetti and Fortunato, 2012, Liu et al., 2015a], the prior assumption that all clustering methods should have the same contribution to the model has no basis. Essentially, a ”bad” clustering that contributes equally with a ”good” clustering could bias result. The main objective of this study is to handle with this problem and to improve existing traditional consensus model. Also, we extend our study to use proposed core idea to develop new applications.

This dissertation composed of 6 chapters, the first chapter is the introduction to DM and explanation of the primary idea of unsupervised learning and consensus clustering. In the second chapter, we give a brief literature review concerning the development of consensus clustering and recent studies. Since this is the general review, we do not provide additional subsections for the literature review purpose in the Chapters 3 and 5. In Chapter 4, we give another short review concerning the particular problem. All chapters are self-standing sections; each has an introduction, methodology of proposed method, results, and conclusion.

In Chapter 3, we propose a weighted consensus clustering based on internal validity measures. The primary objective of this research is to deal with this traditional combination procedure by using internal validity measurements which can be used as weights and they can reflect the goodness of individual clusterings while combining of different partitions. Here we aim to produce better

results than consensus clustering regarding robustness and consistency.

On the other hand, determining the number of a cluster which is an unknown parameter of any clustering algorithm is a crucial process, and there is no universal agreement on the best way of finding the correct or the most suitable number of clusters. Therefore, in Chapter 4 we propose our additional contribution which is to accurately predict the number of a cluster by using proposed weighted consensus clustering algorithm.

In Chapter 5, we develop a better weighting policy for unsupervised ensemble learning based on Markowitz portfolio theory. Here, instead of using only internal validity indexes as weight, we also use the variation of them to produce an optimum weight for each algorithm. Our key objective here is to reduce the variance of accuracy performance of proposed weighted consensus clustering.

Chapter sixth summarizes the results of proposed methods. We discuss the contribution of methods. Finally, we conclude our study and give direction for future research.

CHAPTER 2: LITERATURE REVIEW

This chapter composed of two sections. In the first section, we provide a brief methodological background of consensus clustering, various development, and some applications. Through the second chapter, we focus on studies in the area of consensus clustering introduced from 2010 to today.

Background of Consensus Clustering

Clustering consists in identifying groups of samples with similar properties, and it is one of the most common preliminary exploratory analysis for revealing "hidden" patterns, in particular for datasets where label information is unknown [Ester et al., 1996]. With the rise of big data efficient and robust algorithms able to handle massive amounts of data in a considerable amount of time are necessary [Abello et al., 2013, Rajaraman et al., 2012]. Clustering finds applications in numerous domains including information retrieval and text mining [Jain et al., 1999], spatial database applications [Sander et al., 1998], sequence and heterogeneous data analysis [Cades et al., 2001], web data analysis [Srivastava et al., 2000], bioinformatics [de Hoon et al., 2004], text mining [Jain et al., 1999] and many others. Some of the most common clustering schemes include, but are not limited to k-means [MacQueen et al., 1967], hierarchical clustering [McQuitty, 1957, Sneath, 1957], spectral clustering [Shi and Malik, 2000], and density-based clustering approaches [Ester et al., 1996]. The detailed taxonomy of clustering methods is given in Figure 1.2 in Section 1. Given the diverse objectives and methodological foundations of these methods, it is possible to yield clustering solutions that differ significantly across algorithms [Haghtalab et al., 2015]. Even for multiple runs of the same algorithm, on the same dataset, one is not guaranteed the same solution. This is a well-known phenomenon that is attributed to the local optimality of clustering algorithms such

as k-means [Xanthopoulos, 2014]. In addition to local optimality, algorithmic choice or even the dataset itself might be responsible for utterly unreliable and unusable results. Therefore, once we apply two different clustering algorithm to the same dataset and obtain entirely different results, it is not easy to say the correct one. To handle with this problem, consensus clustering can help to minimize this variability through an ensemble procedure that combines the "good" characteristics from a diverse pool of clusterings [Fred and Jain, 2005, Vega-Pons and Ruiz-Shulcloper, 2011, Liu et al., 2015a]. It has emerged as a powerful technique to produce an optimum and useful partition of a dataset. Some studies such as [Fred and Jain, 2005, Topchy et al., 2004, Strehl and Ghosh, 2003] defined various properties that endorses the use of consensus clustering. Some of them are described as follows:

- Robustness: The consensus clustering might have better overall performance than majority of individual clustering methods.
- Consistency: The combination of individual clustering methods is similar to all combined ones.
- Stability: The consensus clustering shows less variability across iterations than all combined algorithms.

In terms of properties like these, the better partitions can be produced in comparison to the majority of individual clustering methods. However, it cannot be expected the result of consensus clustering as the best result. It can only be ensured that consensus clustering outperforms the majority of all single algorithms combined concerning some properties by assuming as fact that combination of good characteristics of various partition is more reliable than any single algorithm.

Over the past years, many different algorithms have been proposed for consensus clustering [Al-Razgan and Domeniconi, 2006, Ana and Jain, 2003, d Souto et al., 2006, Azimi and Fern, 2009,

Hadjitodorov et al., 2006, Hu et al., 2005, Li and Ding, 2008, Li et al., 2007, Naldi et al., 2013, Ren et al., 2016, Huang et al., 2016a]. As we mentioned earlier, it can be seen in the literature that the consensus clustering framework is able to enhance the robustness and stability of clustering analysis [Fred and Jain, 2002]. Thus, consensus clustering has gained a lot of real-world applications such as gene classification, image segmentation [Hong et al., 2008], video retrieval and so on [Jain et al., 1999, Fischer and Buhmann, 2003, Azimi et al., 2006]. From a combinatorial optimization point of view, the task of combining different partitions has been formulated as a *median partitioning problem* which is known to be N-P complete [Křivánek and Morávek, 1986]. Even with the use of recent breakthroughs this approach cannot handle datasets of size greater than several hundreds of samples [Sukegawa et al., 2013]. For a comprehensive literature of formulation of 0-1 linear program for the consensus clustering problem, we refer the reader to [Xanthopoulos, 2014].

The problem of consensus clustering can be verbally defined such that by using given multiple partitions of the dataset, find a combined clustering model- or final partition- that somehow gives better quality regarding some aspects as pointed out above. Therefore, every consensus clustering method is made up of two steps in general: (1) generation of multiple partition and (2) consensus function as shown in Figure 2.1 [Topchy et al., 2004, Topchy et al., 2003, Xu and Tian, 2015].

Generation of multiple partitions is the first step of consensus clustering. This action aims to create multiple partitions that will be combined. It might be imperative for some in particular problems because final partition will depend on partitions produced in this step. Several methods are proposed to create multiple partitions in literature as follows:

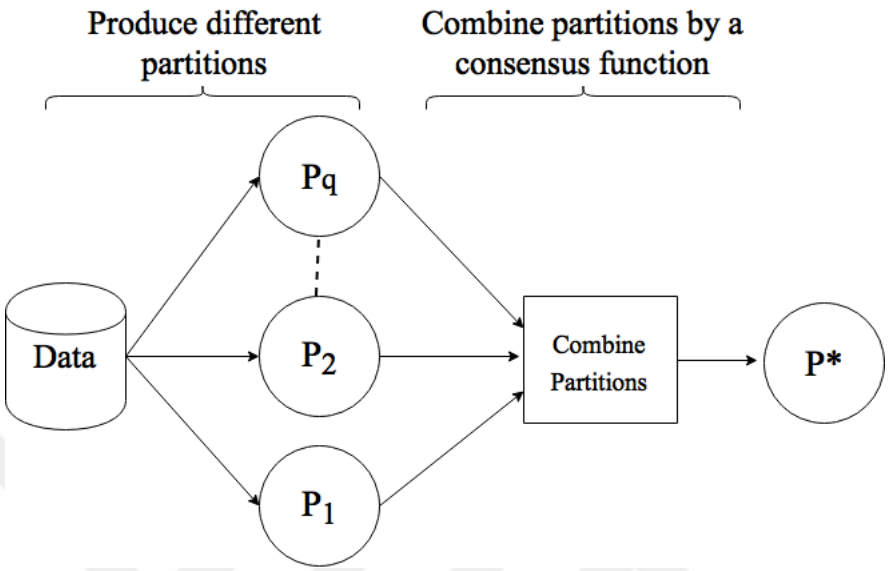

Figure 2.1: The process of consensus clustering

- **For the same dataset, employ different traditional clustering methods**: Using different clustering algorithms might be the most commonly used method to create multiple partitions for a given dataset. Despite the fact that there is no particular rule to choose the conventional algorithms to apply, it is advisable to use those methods that can have more information about the data in general. However, it is not easy to know in advance which methods will be suitable for a particular problem. Therefore, an expert experience could be very useful [Vega-Pons and Ruiz-Shulcloper, 2011, Strehl and Ghosh, 2003, Xu and Tian, 2015].

- **For the same dataset, employ different traditional clustering methods with different initializations or parameters**: Using different algorithms with a different parameter or initialization is an another efficient methods [Ailon et al., 2008, Fred and Jain, 2002]. A simple algorithm can produce different informative partition about the data, and it can yield an effective consensus in conjunction with a suitable consensus function. For example, using the

k-means algorithm with different random initial centers and a various number of the cluster to generate different partitions introduced by [Fred and Jain, 2005].

- **Using weak clustering algorithms**: In generation step, the weak clustering algorithms are also used. These methods produce a set of partition for data using very straightforward methodology. Despite the simplicity of this kind of methods, it is showed that weak clustering algorithms could provide high-quality consensus clustering along with a proper consensus function [Topchy et al., 2005, Luo et al., 2006, Topchy et al., 2003]

- **Data resampling**: Data resampling such as bagging and boosting is an another useful method to create multiple partitions [Hong et al., 2008, Dudoit and Fridlyand, 2003]. Dudoit S. and Jane Fridlyand J. [Dudoit and Fridlyand, 2003] applied a partitioning clustering method (e.g., Partitioning Around Medoids) to a set of bootstrap learning data to produce multiple partitions. They aimed to reduce variability in the partitioning based algorithm result by averaging. And, they successfully produced more accurate clusters than an application of a single algorithm.

The consensus function is the crucial and leading step of any consensus clustering algorithm. These functions are used to combine a set of labels produced by individual clustering algorithms in the previous step. The combined labels - or final partition- can be considered as a result of another clustering algorithm. Foundation or definition of a consensus function can profoundly impact the goodness of final partition which is the product of any consensus clustering. However, the way of the combination of multiple partitions is not the same in all cases. A sharp -but well-accepted- division of consensus functions are (1) objects co-occurrence and (2) median partition approaches.

The idea of objects co-occurrence methods works based on similar and dissimilar objects. If two data points are in the same cluster, those can be considered as similar, otherwise dissimilar. Therefore, in objects co-occurrence methods it should be analyzed how many times data samples belongs

to one cluster. In median partition approach, the final partition is obtained by solving an optimization problem which is the problem of finding the median partition concerning cluster ensemble. Now we can define the formal version of the *median partition problem*. Given a set of q partitions and a similarity measure such as distance $\omega(,)$ between two partitions, we want to find a set of partition P^* such that:

$$P^* = \underset{P}{\operatorname{argmin}} \sum_{i=1}^{q} \omega(P_i, P) \tag{2.1}$$

We can find the detailed review of consensus functions, and taxonomy of principal consensus functions in different studies by [Vega-Pons and Ruiz-Shulcloper, 2011, Xu and Tian, 2015, Topchy et al., 2004, Ghaemi et al., 2009]. Also, relations among different consensus functions can be found in [Li et al., 2010]. We summarized some of the main functions as follows:

- **Based on relabeling and voting**: These methods are based on two important steps. At the first step, the *labeling correspondence problem* needs to solved. The label of each sample is symbolic; a set of the label given by an algorithm might be different than labels given by another algorithm while, however, both sets of labels correspond to the same partition. This problem what makes the combination process is involved. If the labeling correspondence problem is solved, then at the second step voting procedure can be applied. The voting process finds how many times a sample is labeled with the same label. To apply these methods, each produced partition should have the same number of the cluster with final partition [Topchy et al., 2005, Vega-Pons and Ruiz-Shulcloper, 2011]. On the other hand, the strength of this method is easy to understand and employ. Plurality Voting (PV) [Fischer and Buhmann, 2003], Voting-Mergin (VM) [Weingessel et al., 2003], Voting for fuzzy clusterings [Dimitriadou et al., 2002b], Voting Active Cluster (VAC) [Tumer and Agogino, 2008]. and

Cumulative Voting (CV) [Ayad and Kamel, 2008] can be given as examples.

- **Based on co-association matrix**: Algorithms based on the co-association matrix is used to avoid the *labeling correspondence* problem. The main idea of this approach is to create a co-association matrix whose each element is computed based on how many times two particular samples are in the same clusters. A clustering algorithm is necessary to produce the final partition. One of the deficiency of this kind of algorithm is that the complexity of method quadratic in the number of samples. Therefore it is not suitable for large datasets. On the other hand, they are very easy to understand and employ. Evidence accumulation in conjunction with Single Link (EA-CL) or Complete Link algorithms (EA-CL) [Fred, 2001] can be given as examples.

- **Based on graph partition**: This kind of methods transform the combination of multiple partitions into graph or hypergraph partitioning problem [Vega-Pons and Ruiz-Shulcloper, 2011]. All partitions in ensemble procedure can be represented by a hyperedge, and final partition is obtained by implementing a graph-based clustering algorithm. Three graph partitioning algorithms, Cluster-based Similarity Partitioning Algorithm (CSPA), Hypergraph Partitioning Algorithm (HGPA), and Meta-CLustering Algorithm (MCLA), are proposed by [Strehl and Ghosh, 2003]. In CSPA, a similarity matrix is created from a hypergraph. Each element of this matrix shows how many times two points are assigned to the same cluster. Final partition can be obtained by applying a graph similarity-based algorithm such as spectral clustering or METIS. In HGPA, the hypergraph is directly clustered by removing the minimum number of hyperedges. To cluster the hypergraph, HMETIS [Karypis et al., 1999] algorithm is used. In MCLA, the similarity between two clusters is defined based on the number of common samples by using Jaccard index. The similarity matrix between the clusters is the adjacency matrix of the graph whose nodes are the clusters and edge is the similarity between the clusters. METIS algorithm used to recluster that graph. While com-

putational and storage complexity of CSPA is quadratic in the number of sample n, HGPA and MCLA are linear in n.

Another graph based method is Hybrid Bipartite Graph Formulation (HBGF) is proposed by [Fern and Brodley, 2004]. As different from the previous methods, they showed both samples and clusters of the ensemble simultaneously as vertices in the bipartite graph. In this graph, edges are only between clusters and samples (edges with zero weights are no exist). The final partition is obtained by using a graph similarity-based algorithm.

- **Based on information theory**: Information theory based algorithms define the ensembling problem as the finding median partition by a heuristic solution. In these methods, the category utility function is used to determine the similarity measures between clusters. Within the context of clustering, the category utility function [Gluck, 1985] can be defined as the partition quality scoring function. It is proved that this function is same as within cluster variance minimization problem and it can be maximized by using k-means algorithm [Mirkin, 2001]. Using k-means algorithms, on the other hand, bring a deficiency which is the necessity of determining the number of cluster as an initial parameter. Besides, the method should be run multiple times to avoid bad local minima. For the methodological details and implementation of the method, we can refer the reader to [Topchy et al., 2005, Gluck, 1985].

- **Based on local adaptation**: Local adoption based algorithm combines multiple partition generated by using locally adaptive clustering algorithm (LAC) which is proposed by [Domeniconi et al., 2007] with different parameters initialization. Weighty similarity partition algorithm (WSPA), weighty bipartite partition algorithm (WBPA) [Domeniconi and Al-Razgan, 2009], and weighted subspace bipartite partitioning algorithm (WSPA). To obtain final partition, each method uses a graph partitioning algorithm such as METIS. The strong restriction of these kinds of methods is that LAC algorithms can be applied to only numerical data.

- **Based on kernel method**: Weighted partition consensus via Kernels (WPCK) is proposed by [Vega-Pons et al., 2010]. This method uses an intermediate step called Partition Relevance Analysis to assign weights to represent the significance of the partition in the ensemble. Also, this approach defines the consensus clustering via the median partition problem by using a kernel function as the similarity measure between the clusters [Vega-Pons and Ruiz-Shulcloper, 2011]. Other proposed methods using the same idea can be found in [Vega-Pons et al., 2008, Vega-Pons and Ruiz-Shulcloper, 2009]

- **Based on fuzzy theory**: So far, we have explained ensemble clustering methods whose methodology is developed based on hard partitioning. However, we can also work with the soft partitioning. There are clustering methods like EM and fuzzy-c-means that produce soft partition -or called fuzzy partition- of the data. Thus, to combine fuzzy partition instead of hard ones as an internal step of the process is the main logic of these kinds of methods. sCSPA, sMCLA, and sHBGF [Punera and Ghosh, 2008] can be found as examples in literature.

Recent Studies in Consensus Clustering

In the literature, we can find various studies which focus on the development of the consensus clustering or application of the existing methods. To the best of our knowledge, clustering internal validity measures are not combined with graph based consensus clustering. In this section, we summarized some relatively recent and related works. We search those studies by looking at terms listed below.

- Consensus clustering
- Ensemble clustering

- Unsupervised ensemble learning

Ayad and Kamel proposed the cumulative voting-based aggregation algorithm (CVAA) as multi-response regression problem [Ayad and Kamel, 2010]. The CVAA is enhanced by assigning weights to the individual clustering methods that are used to generate the consensus based on the mutual information associated with each method, which is measured by the entropy [Saeed et al., 2014]. Weighted partition consensus via Kernels (WPCK) is proposed by [Vega-Pons et al., 2010]. This method uses an intermediate step called Partition Relevance Analysis to assign weights to represent the significance of the partition in the ensemble. Also, this method defines the consensus clustering via the median partition problem by using a kernel function as the similarity measure between the clusters. Different from partitional clustering methods whose results can be represented by vectors hierarchical clustering methods produce a more complex solution which is shown by dendrograms or trees. This makes using hierarchical clustering in consensus framework more challenging. A hierarchical ensemble clustering is proposed by [Zheng et al., 2010] to handle with this difficult problem. This algorithm combines both partitional and hierarchical clustering and yield the output as hierarchical consensus clustering.

Link-based clustering ensemble (LCE) is proposed as an extension of hybrid bipartite graph (HBGF) technique [Iam-on et al., 2010, Iam-On et al., 2012]. They applied a graph based consensus function to an improved similarity matrix instead of conventional one. The main difference between the proposed method and HBGF is the similarity matrix. While the association between samples is represented by the binary values [0,1] in traditional similarity matrix, the approximate value of unknown relationships (0) is used in the improved one. This is accomplished through the link-based similarity measure called Weighted Connected Triple (WCT). Mainly, after they have created some base partitions, an improved similarity matrix is created to get an optimal partition by using spectral clustering. An improved version of LCE is proposed by [Iam-On and Boongoen, 2012] with

the goal of using additional information by implementing 'Weighted Triple Uniqueness (WTU)'. An iterative consensus clustering is applied to a complex network [Lancichinetti and Fortunato, 2012]. Lancichinetti and Fortunat stress there might be a noisy connection in consensus graph should be removed. Thus, they refined consensus graph by removing some edges whose value is lower than some threshold value and reconnect it to closest neighbor until obtaining a block diagonal matrix. At the end, a graph-based algorithm is applied to consensus graph to get final partition. To efficiently find the similarity between two data points, which can be interpreted as the probability of being in the same cluster, a new index, called the Probabilistic Rand Index (PRI) is developed by [Carpineto and Romano, 2012]. According to the author, they gain better results than existing related methods. One of the possible problem in consensus framework is an inability to handle with uncertain data points which are assigned the same cluster in about the half of the partitions and assigned to different clusters in rest of the partitions. This can yield a final partition with the poor quality. To overcome this limitation, [Yi et al., 2012] propose an ensemble clustering method based on the technique of matrix completion. The proposed algorithm constructs a partially observed similarity matrix based on the pair of samples which are assigned to the same cluster by most of the clustering algorithms. Therefore, the similarity matrix consists of three elements 0,1, and unobserved. It then used the matrix completion algorithm to complete unobserved elements. The final data partition is obtained by applying a spectral clustering algorithm to final matrix [Yi et al., 2012].

A boosting theory based hierarchical clustering ensemble algorithm called *Bob-Hic* is proposed by [Rashedi and Mirzaei, 2013] as an improved version of the method suggested by [Rashedi and Mirzaei, 2011]. *Bob-Hic* includes several boosting steps, and in each step, first a weighted random sampling is implied on the data, and then a single hierarchical clustering is created on the selected samples. At the end, the results of individual hierarchical clustering are combined to obtain final partition. The diversity and the quality of combined partitions are critical properties for a strong

ensemble. Validity Indexes are used to select high-quality partition among the produced ones by [Naldi et al., 2013]. In this study, the quality of a partition is measured by using a single index or combination of some indexes. APMM is another criterion to use determining the quality of partition proposed by [Alizadeh et al., 2014]. This criterion is also used to select some partition among the all produced. A consensus particle swarm clustering algorithm based on the particle swarm optimization (PSO) [Kennedy, 2011] is proposed by [Esmin and Coelho, 2013]. According to the results of this study, the PSO algorithm produces results as good as or better than other well-known consensus clustering algorithms.

A novel consensus clustering called Gravitational Ensemble Clustering (GEC) is proposed by [Sadeghian and Nezamabadi-pour, 2014] based on gravitational clustering [Wright, 1977]. This method combines ”weak” clustering algorithms such as k-means, and according to the authors, it has the ability to determine underlying clusters with arbitrary shapes, sizes, and densities. A weighted voting based consensus clustering [Saeed et al., 2014] is proposed to overcome the limitations of the traditional voting-based methods and improve the performance of combining multiple clusterings of chemical structures.

To reduce the time and space complexity of the suggested ensemble clustering methods, Liu et al. [Liu et al., 2015b] developed a spectral ensemble clustering approach, where Spectral clustering is applied on the obtained co-association matrix to compute the final partition. A stratified sampling method for generating a subspace of data sets with the goal of producing the better representation of big data in consensus clustering framework was proposed by [Jing et al., 2015]. Another approach based on (EAC) is proposed by [Lourenço et al., 2015]. This method is not limited to hard partition and fully use the intuition of the co-association matrix. They determined the probability of the assignment of the points to particular cluster by developed methodology.

Another method based on the refinement of the co-association matrix is proposed by [Zhong et al.,

2015]. From the data sample level, even if a pair of samples is in the same cluster, their probability of assignment might vary. This also affects the contribution of the whole partition. From this perspective, they have developed a refined co-association matrix by using a probability density estimation function.

A method based on giving the weights to each sample is proposed by [Ren et al., 2016]. The idea is coming from boosting method commonly used supervised classification problems. They distinguished points as hard-to-cluster (receive larger weight) and easy-to- cluster (receive smaller weight) based on agreement between partition for a pair of samples. To handle with neglecting diversity of the partition in the combination process, a method based on ensemble-driven cluster uncertainty estimation and local weighting strategy is proposed by [Huang et al., 2016b]. The difference of each partition is estimated via entropic criterion in conjunction with a new novel ensemble-driven cluster validity measure.

According to the [Huang et al., 2016a] introduced the concept of super-object which is the high qualify representation of the data to reduce the complexity of the ensemble problem. They cast consensus problem into a binary linear programming problem, and they proposed an efficient solver based on factor graph to solve it.

Researches on consensus clustering are not limited to those studies summarized above, other contributions can be seen in [Wang et al., 2011b, Wang et al., 2011a, Wu et al., 2013, Lock and Dunson, 2013, Parvin et al., 2013, Berikov, 2014, Gupta and Verma, 2014, Su et al., 2015, Kang et al., 2016]

Here we introduced a modified weighted consensus graph-based clustering method by adding weights that are determined by internal clustering validity measures. The intuition for this framework comes from the fact that internal clustering measures can be used for a preliminary assessment of the quality of each clustering which in turns can be utilized for providing a better clustering

result. By internal quality measures, we refer to the real-valued quality metrics that are computed directly from a clustering and do not include calculations that involve data sample class information as opposed to external quality measures.

CHAPTER 3: A WEIGHTED UNSUPERVISED ENSEMBLE LEARNING BASED ON INTERNAL VALIDITY MEASURES

This section provides the methodology and experimental results of our proposed method along with discussion and future research directions. This method is the base model for the future application and development will be introduced through Chapters 4 and 5, respectively.

Methodology

Consensus Clustering Based on Consensus Graph

The idea of consensus clustering emerges from the combination of the different clustering results obtained for a dataset might help to find a single clustering which fits better to data and emphasizes differences between individual clusters. For this, consensus clustering methods have two crucial components: producing a set of partitions and consensus function that creates a single partition from produced different partitions [Topchy et al., 2005, Strehl and Ghosh, 2003, Lancichinetti and Fortunato, 2012]. As given in details in section 2, in literature, there are various methods to produce different partitions like running single algorithm many times with different parameters or running different clustering algorithms [Fred, 2001, Lancichinetti and Fortunato, 2012, Strehl and Ghosh, 2003] and similar to methods of producing different partitions, one can find various consensus functions in literature such as voting based, co-association matrix based, and graph based [Strehl and Ghosh, 2003, Kuncheva et al., 2006, Xanthopoulos, 2014, Goder and Filkov, 2008, Vega-Pons and Ruiz-Shulcloper, 2011]. In this study, we choose Cluster-Based Similarity Algorithm (CSPA) that builds similarity matrix from hypergraph in which each group of samples represented by an hyperedge proposed by [Strehl and Ghosh, 2003]. Although its computational and storage

complexity are quadratic in the number of samples, it is very easy to use and obvious heuristic [Strehl and Ghosh, 2003]. Basically, CSPA creates $n \times n$ similarity matrix based on similarity partitioning. If two samples are in the same group means represented as the same hyperedge, they can be considered as similar, and otherwise, they are dissimilar. Similarity matrix can be interpreted as a fraction of clustering in which two samples are in the same cluster.

For a given dataset $X = \{(x_i)\}_{i=1}^{N}$ where $x_i \in \mathbb{R}^n$, N and n number of samples and features, respectively. $P = \{P_q \quad | \quad q \in \{1, \cdots, C\}$ is a set of label vectors generated. For given each $P_q \in N^n$, we can construct the binary membership matrix H^q for each partition with a column for each cluster which is now represented by a hyperedge as shown in Table 3.1.

Table 3.1: Representation of orginal partitions (on left) by hyperedges $(h_1, h_2, \cdots, h_k)$. $k = 2$ for P_1 and P_3, and $k = 3$ for P_2 and P_4.

						H^1		H^2			H^3		H^4		
	P_1	P_2	P_3	P_4		h_1	h_2	h_1	h_2	h_3	h_1	h_2	h_1	h_2	h_3
x_1	0	0	0	0		1	0	1	0	0	1	0	1	0	0
x_2	0	0	0	0		1	0	1	0	0	1	0	1	0	0
x_3	0	1	0	0	$\Rightarrow$	1	0	0	1	0	1	0	1	0	0
x_4	1	2	0	2		0	1	0	0	1	1	0	0	0	1
x_5	1	1	1	1		0	1	0	1	0	0	1	0	1	0
x_6	1	2	1	2		0	1	0	0	1	0	1	0	0	1
x_7	1	1	1	1		0	1	0	1	0	0	1	0	1	0

Each $H^q \quad | \quad q \in \{1, \cdots, C\}$ is a $n \times k$ matrix where k is the number of cluster Each entry of the matrix takes a binary value representing if the sample assigned to the corresponding cluster. $H = (H^1 \cdots H^C)$ is the concatenated block matrix that represents adjacency matrix of the hypergraph. We can form $n \times n$ similarity matrix S as in Equation 3.1.

$$S = \frac{1}{C} H H^T \tag{3.1}$$

This similarity matrix can be rewritten as the consensus graph or adjacency matrix of a network. According to the [Strehl and Ghosh, 2003], a graph-based similarity method can be applied to similarity matrix to obtain final partition. In our study, we use spectral clustering due to its robustness compared to other methods [Lancichinetti and Fortunato, 2012], empirically high-performance [Ng et al., 2002], and computationally efficiency due to using only matrix Eigen decomposition [Xanthopoulos, 2014].

In equation 3.1, $\frac{1}{C}$ is basically the weight for algorithms -or partitions- in pool. Thus, each algorithm has the same weight. In other words, each algorithm has same importance effect on consensus graph. In the following section, we explain how these equal weights can be transformed in order to consider the quality of individual algorithms as weights.

Weighted Consensus Clustering Based on Consensus Graph

Although consensus clustering gives more robust and consistent results than individual clustering methods [Deodhar and Ghosh, 2006, Kuncheva et al., 2006, Vega-Pons and Ruiz-Shulcloper, 2011, Lancichinetti and Fortunato, 2012, Liu et al., 2015a], it still might be unstable due to the prior assumption that all clusterings should have the same contribution has no basis. Essentially, a bad clustering that contributes equally with a good clustering could bias result. Our proposed method looks at this weighting policy from a different angle. We believe that internal validity measures can be used as weights and they can reflect the goodness of individual clusterings in the combination of multiple partitions. The main idea is that to give either more or less importance to one cluster based on validity measure while constructing consensus graph. Loosely speaking, weights can make edges in consensus graph more or less visible than regular consensus function. We first define matrix H^* which is the weighted adjacency matrix of the hypergraph, then revise the formulation that constructs $n \times n$ weighted similarity matrix $\hat{S}$ as in equations 3.2 and 3.3.

$$H^* = (W_1 H^1 \cdots W_q H^q) \tag{3.2}$$

$$\hat{S} = H^* H^T \tag{3.3}$$

where $W = \{W_q \quad | \quad q \in 1, \cdots, C\}$ that is normalized as $\sum_{q=1}^{C} W_q = 1$ represents weight of each individual clusterings computed based on internal validity measures. Finally, we can follow same procedure which is using graph-based similarity algorithm to obtain final partition.

Internal Validity Measures

The majority of clustering algorithms might give different results based on attributes of data and some initial assumptions [Halkidi et al., 2002]. So, evaluating clustering results become an important task for reliable results in most applications. In that point, the internal measure can help to give better insight into the performance of clustering methods from different aspects. Since internal measures use inherent information of data alone and in practice, pre-defined information such as class label does not exist in most application, we prefer to use them in our weighting policy for weighted consensus clustering framework.

There are some internal validity measures in literature including RMSSTD (root mean square standard deviation) index [Sharma, 1996], SD validity index [Halkidi et al., 2000], S_Dbw index [Halkidi and Vazirgiannis, 2001], dunn index [Dunn, 1973], silhouette index [Rousseeuw, 1987], calinski-harabasz index [Caliński and Harabasz, 1974], Davies-Bouldin index [Davies and

Bouldin, 1979] and so on. There might be some limitations accross internal validity measures and they can be affected by different data chracteristics. For instance, noise in data can significantly affect performance of internal validity measure, if minumum or maximum pairwise distance is used [Liu et al., 2010].A good comparison of them from different aspects can be found in [Rendón et al., 2011, Liu et al., 2010, Kovács et al., 2005]. Among all those indexes, we choose silhouette index [Rousseeuw, 1987], calinski-harabasz index [Caliński and Harabasz, 1974], and davies bouldin index [Davies and Bouldin, 1979]. According to [Rendón et al., 2011], these indexes show respectively better performance to predict correct number of clusters. In addition, they perform well enough in some aspects such as monotonicity, noise, density, skewed distributions, and subclusters [Liu et al., 2010].

Silhouette Validation Index (SH):

Silhouette validation index proposed by [Rousseeuw, 1987] validates the performance of clustering based on the pairwise distance between and within clusters. Also, the optimum number of clusters can be determined by maximizing index value [Liu et al., 2010].Silhouette value is formalized as in Equation 3.4.

$$s(i) = \frac{b(i) - a(i)}{Max\{a(i), b(i)\}} \tag{3.4}$$

where $s(i)$ is called silhouette width of point. $a(i)$ is the mean distance between ith sample and all the points in given cluster $p_i \quad (i = 1,2,3,...,k)$. And, $b(i)$ is the smallest of these distance. Thus, it can be seen that silhouette value will be between 1 and -1.

Calinski-Harabasz Validation Index (CH):

Calinski-Harabasz validation index proposed by [Caliński and Harabasz, 1974] evaluates cluster quality based on the mean between and within cluster sum of squares. It is defined as:

$$CH = \frac{SS_B}{SS_W} \times \frac{(n-k)}{(k-1)} \tag{3.5}$$

where SS_B is average between-cluster sum of squares, SS_W is the average within-cluster sum of squares, k is the number of clusters, and n is the number of observations. The average between-cluster sum of squares is computed as:

$$SS_B = \sum_{i=1}^{k} n_i \|m_i - \mu\|^2 \tag{3.6}$$

where k is the number of clusters, m_i is the centroid of cluster k, μ is the mean of the all samples, and $\|m_i - \mu\|$ is the euclidean distance between centroid of cluster and mean of all samples. The average between-cluster sum of squares is computed as;

$$SS_W = \sum_{i=1}^{k} \sum_{x \in p_i} \|x - m_i\|^2 \tag{3.7}$$

where k is the number of clusters, x is a sample, p_i is the ith cluster, m_i is the centroid of cluster p_i, and $\|x - m_i\|$ euclidean distance between sample and centroid of cluster.

Large CH value shows better data partition. So, a well-defined clustering has a high SS_B and low

SS_W value.

Davies-Bouldin Validation Index (DB):

Davies bouldin index proposed by [Davies and Bouldin, 1979] try to identify clusters which are compact and well-separated. It is computed as:

$$DB = \frac{1}{k} \sum_{i,j=1}^{k} max_{i \neq j} \left\{ \frac{\hat{d}_i + \hat{d}_j}{d_{i,j}} \right\} \tag{3.8}$$

where $\hat{d}_i$ is the mean distance between each sample in the ith cluster and the centroid of the ith cluster. $\hat{d}_j$ is the mean distance between each sample in the jth cluster and the centroid of the jth cluster. $d_{i,j}$ is the euclidean distance between the centroid of the ith and jth clusters. Low DB index value refers to the well-defined data partition.

One needs to note that we use the inverse of Davies-Bouldin values because of the minimum value of it shows the better partition.

Illustrative Example

In this section, we propose a toy example to make the concept of weighted consensus clustering method more concrete. Let us consider that we are given a set of algorithm results $P = (P_1, P_2, P_3, P_4)$ for a dataset X as follows:

$$P_1 = \{0,0,0,1,1,1,1\}$$
$$P_2 = \{0,0,1,2,1,2,1\}$$

$$P_3 = \{0,0,0,0,1,1,1\}$$
$$P_4 = \{0,0,0,2,1,2,1\}$$

These individual partitions construct similarity matrix S based on CSPA consensus function given in equation 3.1.

$$S = \begin{Bmatrix} 1 & 1 & 0.75 & 0.25 & 0 & 0 & 0 \\ 1 & 1 & 0.75 & 0.25 & 0 & 0 & 0 \\ 0.75 & 0.75 & 1 & 0.25 & 0.25 & 0 & 0.25 \\ 0.25 & 0.25 & 0.25 & 1 & 0.25 & 0.75 & 0.25 \\ 0 & 0 & 0.25 & 0.25 & 1 & 0.5 & 1 \\ 0 & 0 & 0 & 0.75 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0.25 & 0.25 & 1 & 0.5 & 1 \end{Bmatrix}$$

Each element S_{ij} in similarity matrix S represent the weight between node i and node j in consensus graph. These weights are a fraction of clustering in which two samples are in the same cluster.

Now, besides given different clustering results P assume that we are also given corresponding weights $W_c = (0.45, 0.28, 0.18, 0.09)$ coming from one of internal validity measures. Based on equations 3.3 and 3.2, similarity matrix $\hat{S}$ is computed as follows:

$$\hat{S} = \begin{Bmatrix} 1 & 1 & 0.72 & 0.18 & 0 & 0 & 0 \\ 1 & 1 & 0.72 & 0.18 & 0 & 0 & 0 \\ 0.72 & 0.72 & 1 & 0.18 & 0.28 & 0 & 0.28 \\ 0.18 & 0.18 & 0.18 & 1 & 0.45 & 0.82 & 0.45 \\ 0 & 0 & 0.28 & 0.45 & 1 & 0.63 & 1 \\ 0 & 0 & 0 & 0.82 & 0.63 & 1 & 0.63 \\ 0 & 0 & 0.28 & 0.45 & 1 & 0.63 & 1 \end{Bmatrix}$$

Each element $\hat{S}_{i,j}$ in similarity matrix $\hat{S}$ represent the weight between node i and node j in consensus graph. These weights are a weighted fraction of clustering in which two samples are in the same cluster. Figure 3.1 represents how different partitions can be combined based on consensus clustering and weighted consensus clustering approach.

That can be seen that weighting policy give more or less similarity values in comparison to traditional consensus clustering. For example, edge $\hat{S}_{4,6}$ has more weight in weighted consensus method than traditional consensus one, or edge $\hat{S}_{4,1}$ has less weight.

Computational Results and Discussions

In this section, we present experiment results of individual clusterings, consensus and proposed weighted consensus clusterings. We conduct experiments on 20 different datasets to evaluate the performance of weighted consensus clusterings in comparison to individual clustering techniques and consensus clustering. Table 3.2 gives the details of 20 datasets.

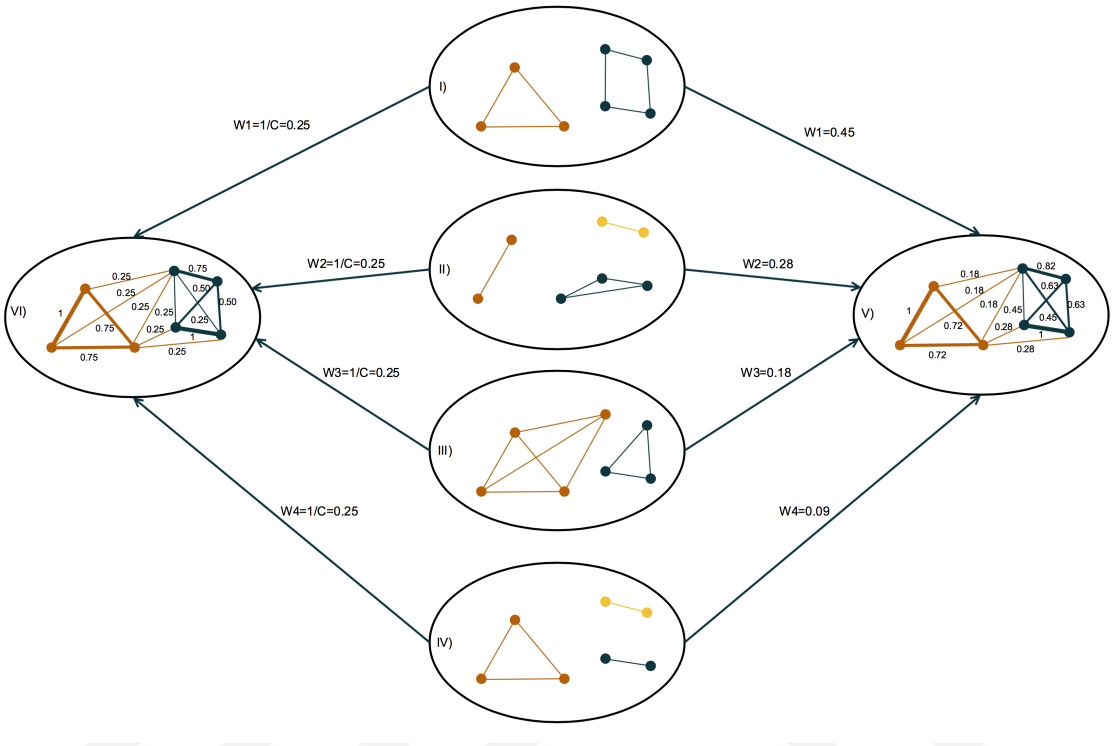

Figure 3.1: Illustrative example of consensus graph. I, II, III, and IV show results of individual clustering, and graph V and VI are weighted consensus and consensus methods, respectively. While each algorithm has equivalent effect on consensus graph, which is $1/C$, they have different effect on weighted consensus graph, which is W_c.

All datasets are used as found in the original repositories. Only exceptions are the dataset Letter_IJL that consists of capital English letters I, J, and L and MNIST_123 that consists of handwritten digits 1, 2, and 3 are randomly sampled from Letter and MNIST datasets. On the other hand, since spectral clustering does not work well for imbalance datasets, we ignore some group of samples in Balance and Yeast datasets to avoid having imbalance data.

Table 3.2: Description of datasets.

Datasets	# Samples	# Attributes	# Cluster	Source
Aggregation	788	2	7	[Gionis et al., 2007]
Appendicitis	106	7	2	[Weiss and Kulikowski, 1991]
Breast	679	9	2	[Ferris and Mangasarian, 1995]
Zoo	101	16	7	[Lichman, 2013]
WDBC	569	30	2	[Lichman, 2013]
Letter_IJL	400	16	3	[Lichman, 2013]
Liver	341	6	2	[Lichman, 2013]
Balance	576	4	2	[Lichman, 2013]
Banknote	1372	4	2	[Lichman, 2013]
Ecoli	272	7	3	[Lichman, 2013]
Glass	214	9	6	[Lichman, 2013]
Soybean	47	35	4	[Lichman, 2013]
Yeast	892	8	2	[Lichman, 2013]
Seeds	210	7	3	[Lichman, 2013]
Wine	178	12	3	[Lichman, 2013]
Iris	150	4	3	[Lichman, 2013]
Compound	399	2	6	[Zahn, 1971]
MNIST_123	500	400	3	[LeCun and Cortes, 2010]
Pathbased	300	2	3	[Chang and Yeung, 2008]
Flame	240	2	2	[Fu and Medico, 2007]

In machine learning community, the average accuracy is the most common external validation measure unless the majority of instances labeled as one class. If this is the case, average accuracy might give a misleading idea about performance classifier because of assigning instances to the dominant class [Kotsiantis et al., 2006, Brodersen et al., 2010, Weng and Poon, 2008]. All datasets selected in our experiment are balanced so that we report the clustering accuracy (Acc) that is a reliable measure regarding the performance of clustering methods in our case. It is calculated as in equation (3.9) [Li and Ding, 2008, Li et al., 2006]. Besides, three internal measures described earlier -silhouette (SH), Calinski-Harabasz (CH), and Davies-Bouldin (DB) - are also reported through our evaluation.

$$\text{Accuracy} = \text{Max}(\sum_{p_k, L_m} T(p_k, L_m))/n \tag{3.9}$$

where n is number of samples, p_k represents kth cluster, L_m is the mth class and $T(C_k, L_m)$ is the number of samples in class m assigned to cluster k.

Also, we use five different individual clustering algorithms -Fuzzy [Jang et al., 1997], Gaussian clustering used the Expectation-Maximization (EM) algorithm [McLachlan and Peel, 2000], Hierarchical [Johnson, 1967], K-means [MacQueen et al., 1967], and Spectral clustering [Ng et al., 2002]-, consensus clustering (CON) [Strehl and Ghosh, 2003], and 3 weighted consensus clusterings. WConSH, WConCH, and WConDB using silhouette, calinski-harabasz, and davies-bouldin index values as weight, respectively.

The selected algorithms have distinct algorithmic differences, and they can show different performance based on the data structure. For example, spectral clustering performs better for balanced data while k-means is more suitable for normally distributed data. Gaussian clustering has some advantages such as exist well-studied statistical inference techniques and flexibility regarding choosing a component distribution. Moreover, it can accommodate clusters that have different sizes and correlation structures within them. Hierarchical is commonly used a greedy iterative approach in various fields including medical. We aim to create an algorithm pool with some diversity to use advantages of different algorithms in different data structures.

All datasets features are initially normalized before clustering so that they have 0 mean and unitary standard deviation. We performed the experiment on Intel Core i5, 2.3 GHz with 8 Gb of RAM in a 64-bit platform. And all codes are developed in Matlab version 2014a and R version 3.2.3.

Results

Through our experiment, we compare our proposed method with individual clusterings and consensus clustering regarding given evaluation metrics. One needs to take into consideration that chosen individual algorithms might affect the performance of consensus and weighted consensus clusterings. Thus, results of methods for some datasets might be different than similar studies. In order to avoid bias outcomes, we run each individual algorithm 30 times, and results are the average of them. On the other hand, 30 different similarity matrixes are used in consensus clustering and weighted consensus clusterings. Then, results are computed by averaging them. Additionally, we shift negative weight up before normalizing them.

Our principal objective is to enhance the performance of consensus clustering regarding more robust accuracy and other given performance measures. Since clustering methods are sensitive to different data structures, getting more robust results regardless of the data structure is crucial. For instance, as shown in Table 3.3, clustering techniques show inconsistent results. While hierarchical clustering performs not bad regarding accuracy, the performance of fuzzy, Gaussian and spectral clustering are quite poor. In that case, consensus method gives a relatively good result, which is better than the majority of individual algorithms. Furthermore, we can improve the performance of consensus method by WConSH and WConDB using SH and DB indexes as weight.

In the problem above, it is not easy to improve the result of each individual method due to a quite high performance of them. However, we can get more robust and consistent result by using consensus clustering and proposed weighted consensus clusterings. And, we successfully improve the performance of traditional consensus clustering by using the weight that reflects the quality of clustering solutions. WConSH also gives better internal validity measure.

Table 3.3: Results of clustering methods for the Ecoli dataset. Italicized values show the best performance among all methods, and boldface entries show the best performance among consensus and weighted consensus methods.

Algorithms	Accuracy	Silhouette	Calinski-Harabasz	Davies-Bouldin
Fuzzy	91.180	0.420	*225.800*	0.900
Gaussian	*94.860*	0.430	215.950	0.780
Hierarchical	87.500	0.390	205.970	*0.760*
Spectral	93.330	*0.440*	222.990	0.870
K-means	87.810	0.410	209.020	0.920
Consensus	88.940	0.390	200.040	1.030
WConSH	**90.810**	**0.410**	**216.350**	**0.930**
WConCH	87.870	0.380	195.520	1.050
WConDB	90.420	**0.410**	213.080	0.970

Thus, we can say that WConSH gives more reliable partitions with less variability. In that point, it worths mentioning that having the high correlation between accuracies and weights in the majority of algorithms might help more to enhance the performance of consensus clustering. For this problem, the correlation between accuracies and weights is 0.88 in WConSH, which is pretty good, and reflect the goodness of clusterings' solutions, as they should be. The figure 3.2 illustrates accuracies of individual clusterings and corresponding weights for WConSH.

As we mentioned earlier, there is no guarantee that one clustering method achieves similar performance regardless of datasets. That is a typical situation that one clustering method poorly performs for one dataset while its performance quite well for another dataset. Table 3.4 shows an example of how same clustering method might vary across the datasets.

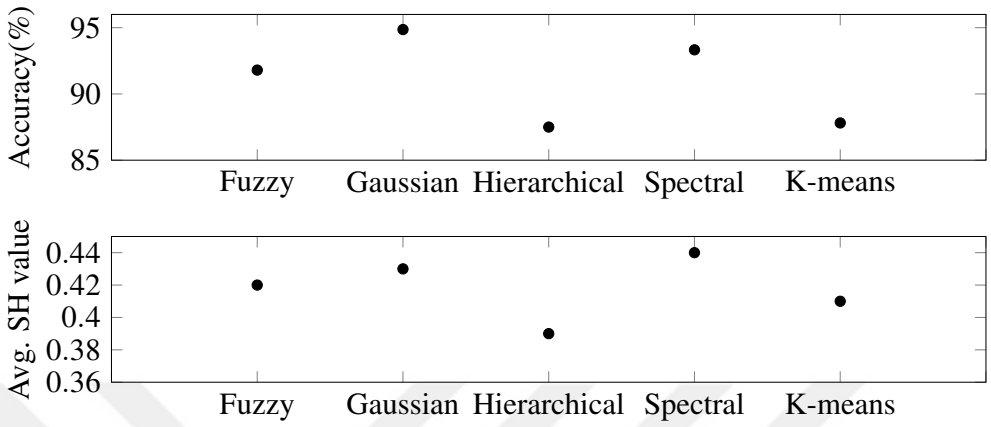

Figure 3.2: Accuracy performance of individual algorithms and corresponding mean SH values used as weights for Ecoli dataset.

As different from previous Ecoli data, the performance of Gaussian clustering dramatically dropped in MNIST_123 dataset. Moreover, solutions of clustering might vary not only across dataset but also they might show inconsistent performance across the algorithms for the same dataset. The problem is given in Table 3.4 illustrates that accuracy performance of clustering methods quite different from each other. To handle with inconsistent results of clustering methods, we use consensus framework to get more robust and consistent results. And, our proposed methods enhance the performance of traditional consensus clustering while saving consistency.

We mentioned above that having a high correlation between accuracies and weights in the majority of dataset make improving the performance of consensus clustering easier. In the MNIST_123 problem, correlation coefficient in WConSH is 0.25 out of 5 algorithms that respectively bad. However, weights and accuracies are inversely correlated in only fuzzy clustering. While its accuracy performance is not as good as other clustering methods, it's SH index value used as weight in WConSH is the best one among all other methods.

Table 3.4: Results of clustering methods for MNIST_123 dataset

Algorithms	Accuracy	Silhouette	Calinski-Harabasz	Davies-Bouldin
Fuzzy	61.400	*0.147*	56.910	2.308
Gaussian	69.200	0.081	35.628	5.804
Hierarchical	*90.400*	0.130	63.743	2.685
Spectral	68.600	0.064	45.381	3.443
K-means	80.160	0.136	*69.431*	*2.275*
Consensus	78.160	0.108	59.523	2.702
WConSH	**79.420**	**0.120**	**64.187**	**2.440**
WConCH	77.980	0.112	61.417	2.548
WConDB	78.060	**0.120**	63.264	2.493

If we consider other four algorithms, the correlation between accuracies and weights is 0.87. So that high correlation in other four algorithms will eliminate the adverse effect of inverse correlation in fuzzy clustering. Thanks to this, we still might get better results than traditional consensus clustering even if some weights of individual clustering methods are not correlated with their accuracy. The figure 3.3 shows the relation between accuracies of individual clustering methods and weights for WConSH.

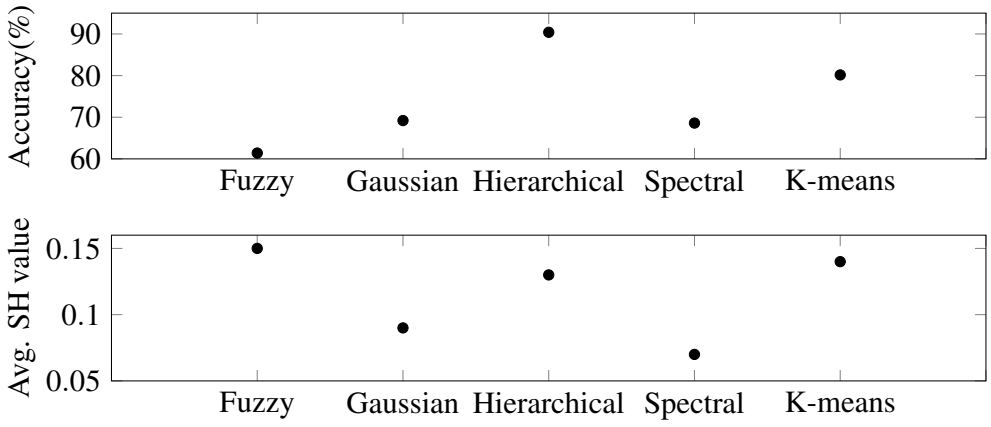

Figure 3.3: Accuracy performance of individual algorithms and corresponding mean SH weights used for MNIST_123 dataset.

The figure 3.4 is a good illustration to show the performance of proposed weighted consensus clustering methods in comparison to traditional consensus clustering in terms of accuracy. Using different weights based on validity measures help to create better consensus. We can see the pattern in figure 3.4 that shows those methods yield better overall accuracy results than consensus clustering methods. One needs to note that we normalized results as being between 0 and 1 to compare them in good scalability.

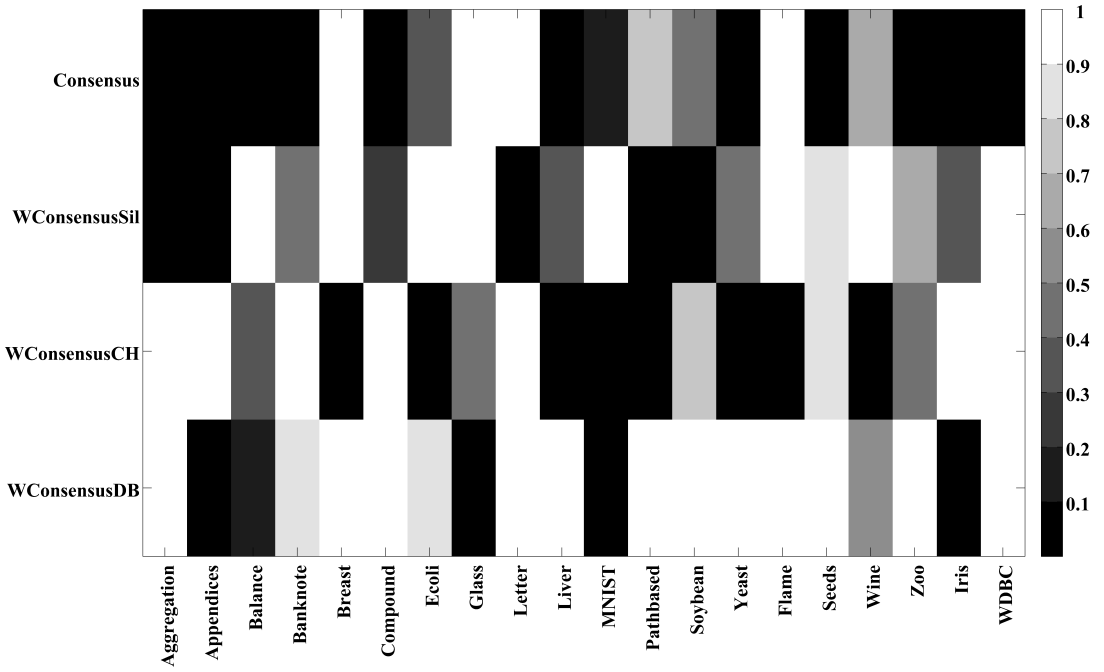

Figure 3.4: The figure denotes that comparison of weighted consensus clusterings and consensus clustering in terms of accuracy.

Also, Tables 3.5 and 3.6 show results of individual, consensus, and weighted consensus clusterings for all datasets in detail with respect to accuracy and three internal validity measures which are SH, CH, and DB as a sign of reliability of clustering results. Weighted consensus clusterings show better performance than traditional consensus clustering not only for accuracy but also three

selected internal validity measures in the majority of datasets. So that, we can see that proposed methods will give more reliable results than consensus method. In the following section, we provide summary tables to show how many times weighted consensus clustering provides better or same results regarding given evaluation metrics out of 20 datasets.

Conclusion

In this study, we propose an unsupervised weighted consensus framework for solving different types of problem. Specifically, we use internal validity measures as weights for individual clusterings in creating consensus matrix. The use of weighted consensus clustering helps us to give different importance to individual clustering based on the goodness of their results. Besides providing better accurate partition, weighted consensus frameworks enhance the quality of clusterings in the majority of datasets.

Table 3.7 shows the comparison of consensus clustering and proposed weighted consensus clustering methods with respect to accuracy and internal validity indexes. One needs to note that consensus and weighted consensus clusterings methods might give same results for some datasets. Generally, we can conclude that weighted consensus schemes outperform for all aspects. On the other hand, with respect to accuracy WConSH, WConCH, and WConDB give better results than 2.5, 2.2, and 2.55 number of individual clusterings on average out of 5 chosen. Giving better validity measures help us to conclude using weighted consensus approaches might provide more reliable partitions even if accuracy is not as good as traditional consensus framework.

Moreover, the last row of Table 3.7 shows how many times at least one weighted consensus clusterings give better or same results than traditional one with respect to given evaluation metrics. That shows there exist a real potential to vastly outperforms traditional consensus method in the case using single and better weighting policy. Our weighting policy might severely affect the performance

Table 3.5: The performance of individual, consensus, and weighted consensus clusterings for all datasets regarding evaluation metrics (EM); accuracy and three internal validity measures. While italicized values show the best performance among all methods, bolded ones shows the best performance among consensus and weighted consensus methods.

Datasets	EM	Fuzzy	Gaussian	Hierarchical	Spectral	K-means	Consensus	WConSH	WConCH	WConDB
Aggregation	Acc	73.60	78.69	81.22	*91.44*	77.64	74.68	74.91	77.77	**77.81**
	SH	0.45	0.50	*0.51*	*0.51*	0.49	0.38	0.41	**0.42**	0.39
	CH	1214.19	1228.62	*1358.37*	931.17	1310.44	730.80	832.94	**916.25**	789.04
	DB	0.69	*0.49*	0.51	0.67	0.67	1.71	**0.79**	0.99	1.13
Appendicities	Acc	79.25	58.50	*81.14*	77.90	80.85	79.78	79.78	**80.32**	79.82
	SH	0.38	0.18	*0.40*	0.36	0.39	**0.38**	**0.38**	**0.38**	**0.38**
	CH	71.61	7.82	56.56	30.40	71.55	70.80	70.80	***71.62***	70.79
	DB	1.04	3.08	1.04	1.13	1.02	1.05	1.05	***1.03***	1.05
Balance	Acc	49.290	54.880	45.760	*57.820*	52.230	51.960	**53.820**	52.550	52.320
	SH	0.170	0.090	0.140	0.170	*0.180*	**0.160**	**0.160**	**0.160**	**0.160**
	CH	126.380	90.300	101.030	124.900	*134.810*	117.130	115.380	**119.730**	114.360
	DB	*1.750*	3.190	1.970	1.780	1.720	1.830	1.840	**1.800**	1.850
Banknote	Acc	55.500	58.750	52.500	*69.880*	56.090	53.500	53.770	**54.100**	53.980
	SH	0.400	0.350	0.390	*0.540*	0.410	**0.390**	**0.390**	**0.390**	**0.390**
	CH	372.840	297.810	339.150	235.300	*375.380*	346.160	356.970	358.810	**359.230**
	DB	0.950	1.070	1.000	*0.680*	0.940	1.000	**0.980**	**0.980**	**0.980**
Breast	Acc	91.760	57.740	88.810	64.400	90.630	***92.550***	92.550	92.380	***92.550***
	SH	0.300	0.300	0.280	*0.450*	0.300	**0.300**	**0.300**	**0.300**	**0.300**
	CH	299.370	98.750	265.920	5.110	293.530	***300.480***	***300.480***	300.020	***300.480***
	DB	1.440	1.570	1.520	*0.980*	1.430	**1.430**	**1.430**	**1.430**	**1.430**
Compound	Acc	59.120	56.900	*69.930*	66.750	62.560	61.230	61.780	**63.080**	62.900
	SH	0.420	0.360	*0.440*	0.410	*0.440*	**0.340**	0.320	**0.340**	**0.340**
	CH	788.400	559.300	*826.490*	576.020	728.760	**525.340**	370.580	462.240	474.690
	DB	0.720	0.680	*0.600*	1.360	0.680	**0.990**	1.390	1.090	1.010
Ecoli	Acc	91.180	*94.860*	87.500	93.330	87.810	88.940	**90.810**	87.870	90.420
	SH	0.420	0.430	0.390	*0.440*	0.410	0.390	**0.410**	0.380	**0.410**
	CH	*225.800*	215.950	205.970	222.990	209.020	200.040	**216.350**	195.520	213.080
	DB	0.900	0.780	*0.760*	0.870	0.920	1.030	**0.930**	1.050	0.970
Glass	Acc	49.340	47.670	*51.410*	48.950	50.600	**47.050**	47.030	46.360	45.860
	SH	0.260	0.160	*0.370*	0.220	0.360	**0.130**	**0.130**	**0.130**	0.110
	CH	86.270	31.330	109.310	62.000	*111.780*	37.920	43.600	**44.890**	35.260
	DB	1.140	*0.720*	0.930	1.300	0.810	1.600	1.650	**1.480**	1.580
Letter_IJL	Acc	*56.110*	50.000	46.750	46.150	50.100	49.590	48.030	**49.600**	49.500
	SH	0.190	0.280	0.290	*0.320*	0.260	0.260	**0.270**	0.260	0.240
	CH	92.920	104.840	112.500	77.480	*117.000*	98.130	**101.700**	99.720	90.160
	DB	1.670	1.010	*0.980*	1.420	1.530	1.530	**1.510**	1.540	1.600
Liver	Acc	53.080	50.440	51.620	*57.070*	55.720	51.910	52.810	52.110	**54.530**
	SH	0.430	0.330	0.390	*0.620*	0.480	0.390	0.420	0.390	**0.510**
	CH	307.750	221.100	277.830	205.280	*324.070*	277.750	**300.750**	282.750	245.840
	DB	0.850	1.070	0.930	*0.530*	0.770	0.930	0.870	0.920	**0.720**

Table 3.6: The performance of individual, consensus, and weighted consensus clusterings for all datasets regarding evaluation metrics (EM); accuracy and three internal validity measures. While italicized values show the best performance among all methods, bolded ones shows the best performance among consensus and weighted consensus methods.

Datasets	EM	Fuzzy	Gaussian	Hierarchical	Spectral	K-means	Consensus	WConSH	WConCH	WConDB
MNIST_123	Acc	61.4	69.2	*90.4*	68.6	80.16	78.16	**79.42**	77.98	78.06
	SH	0.15	0.09	0.13	0.07	*0.14*	0.11	**0.12**	**0.12**	**0.12**
	CH	56.92	35.63	63.75	45.39	*69.44*	59.53	**64.19**	61.42	63.27
	DB	2.31	5.81	2.69	3.45	*2.28*	2.71	**2.44**	2.55	2.5
Pathbased	Acc	74.34	71	70	*85.26*	74.34	74.02	73.32	73.29	**74.3**
	SH	0.51	0.56	*0.57*	0.34	0.52	**0.51**	**0.51**	**0.51**	**0.51**
	CH	358.02	332.7	315.66	163.16	*359.08*	351.87	341.65	341.62	**352.46**
	DB	0.69	*0.63*	0.65	1.53	0.67	**0.68**	**0.68**	0.7	0.69
Soybean	Acc	72.35	*89.37*	76.6	73.55	68.59	66.46	63.41	68.3	**70.08**
	SH	0.35	*0.4*	0.35	0.33	0.32	0.24	0.23	0.28	**0.3**
	CH	*33.7*	30.79	33.2	30.57	28.62	22.79	19.05	24.84	**27.29**
	DB	1.15	*1.13*	1.3	1.15	1.2	**1.24**	1.27	1.29	1.34
Yeast	Acc	53.34	52.67	*55.67*	54.98	53.99	52.18	52.53	52.18	**52.89**
	SH	0.19	0.18	0.21	*0.46*	0.2	0.19	0.2	0.19	**0.23**
	CH	*63.86*	58.43	45.78	17.25	62.78	**63.73**	61.72	**63.73**	57.07
	DB	1.9	1.97	1.79	*0.83*	1.86	1.9	1.87	1.9	**1.76**
Flame	Acc	85	71.67	80.42	69.19	84.64	***89.46***	***89.46***	84.53	89.24
	SH	0.37	0.36	0.37	*0.56*	0.38	**0.38**	**0.38**	**0.38**	**0.38**
	CH	148.63	123.7	142.14	20.88	*155.42*	144.07	144.07	**154.74**	144.94
	DB	1.12	1.19	1.17	*0.66*	1.12	**1.13**	**1.13**	**1.13**	**1.13**
Seeds	Acc	*89.53*	85.24	89.05	72.15	89.24	83	85	84.96	**85.43**
	SH	*0.48*	0.47	0.46	*0.48*	*0.48*	**0.45**	**0.45**	**0.45**	0.44
	CH	*375.81*	353.87	352.84	149.2	375.29	330.89	**331.41**	331.38	319.02
	DB	0.72	0.84	*0.7*	0.84	*0.7*	0.89	0.94	**0.84**	0.92
Wine	Acc	69.11	68.54	67.98	58.23	*69.67*	68.28	**69.33**	66.3	68.06
	SH	0.52	0.39	*0.55*	0.48	0.52	0.47	**0.5**	0.44	0.47
	CH	407.25	203.82	347.48	245.68	*408.01*	334.63	**378.09**	304.34	324.86
	DB	*0.57*	0.96	0.63	1.01	0.59	1.16	**0.59**	2.37	1.26
Zoo	Acc	58.22	65.35	*79.21*	74.11	69.31	58.97	62.48	61.74	**64.81**
	SH	0.31	0.34	*0.39*	0.34	0.36	0.28	**0.34**	0.32	0.33
	CH	60.45	61.67	*70.49*	54.88	62.88	37.41	40.9	41.37	**46.44**
	DB	1.26	*0.9*	0.99	1.13	0.96	**1.14**	1.35	1.52	1.4
WDBC	Acc	*88.93*	69.4	62.92	88.05	88.76	88.89	***88.93***	***88.93***	***88.93***
	SH	0.71	0.29	0.73	0.5	0.71	0.84	0.95	***0.97***	0.94
	CH	743.28	188.68	6.8	393.97	743.93	2315.57	5669.16	***9571.95***	4212.91
	DB	0.73	1.39	0.33	1.03	0.73	0.42	0.24	***0.18***	0.28
Iris	Acc	89.34	*96.67*	90.67	68.74	85.16	84.34	85.63	**87.96**	84.36
	SH	0.55	0.51	0.56	*0.67*	0.54	**0.55**	0.5	**0.55**	0.51
	CH	*560.23*	481.79	556.88	287.38	523.73	498.91	501.13	**540.24**	490.52
	DB	0.65	0.72	0.62	*0.41*	0.67	**0.63**	1.03	0.66	0.79

of the method if validity measures and accuracy is not proportional. In some cases, one algorithm can take much lower weight although its accuracy performance is much better than others. Thus, it might worth for future research to find a better weighting policy.

Table 3.7: Comparison of consensus and weighted consensus clusterings. The first three columns denote comparison of one of proposed weighted consensus clustering and traditional consensus clustering. The values represent how many times a weighted consensus clustering gives better results than consensus one regarding given evaluation measurements. For example, WConSH gives better results than consensus one in 16,18,16 and 14 datasets with respect to reported performance measure Acc, SH, CH, and DB. The last column of the table shows how many times at least one of the weighted consensus clustering shows better performance than consensus clustering(e.g.in 19 datasets at least one weighted consensus clustering out of three gives better accuracy than consensus clustering).

Algorithms			Acc	SH	CH	DB
WConSH	vs.	Consensus	16	18	16	14
WConCH	vs.	Consensus	14	19	15	13
WConDB	vs.	Consensus	16	16	10	11
WCon	vs.	Consensus	19	20	19	16

Finally, in this study, we focus on only the values of internal validity measures of the algorithms to build weights. However, in the rest of this study, we give more attention to the variability of them to minimize the potential risk of the current weighting policy that might vary across the iterations, and also we use proposed the method to determine the number of clusters. We discuss details in the next chapters.

CHAPTER 4: DETERMINING NUMBER OF CLUSTER VIA WEIGHTED CONSENSUS CLUSTERING BASED ON INTERNAL VALIDITY MEASURES

In this chapter, we use the proposed weighted consensus clustering method to determine the number of correct or the most suitable number of clusters. We organize this chapter as follows. First, we give a brief introduction about the determining number of cluster and related works. Then, we introduce our methodology. Finally, we summarized the results and discussed possible implementation in future.

Introduction

One of the major challenges in clustering analysis is to determine the number of clusters for a given data set when the only information available belongs to the data set itself. Although, there exist many studies which propose methods to find correct or the most suitable number of clusters in a given dataset, some studies argued there is no optimal procedure to find correct number of clusters [Everitt et al., 2001, Hartigan, 1975, Bock, 1985, Hardy, 1996, Gordon, 1999]. Steinley and Brusco divided methods for determining number of clusters into four groups which are traditional formulaic procedures used in conjunction with classical clustering procedures, likelihood (e.g., BIC, Akaike information criterion (AIC)), replication analysis, and lower bound of the sum-of-squares error in K-means clustering [Steinley and Brusco, 2011]. We briefly explain some of these methods through the section.

Clustering validity indexes in conjunction with a proper clustering algorithm is a commonly used procedure to determine a correct number of clusters. Based on chosen validity index, either max-

imum or minimum index value might help to find the number of clusters. In a survey paper, [Milligan and Cooper, 1985] performed 30 different criterions including heuristic, ad hoc procedures and well-known validity indexes to estimate the correct number of clusters. Dimitriadou et al proposed another comparison of fifteen validity indices for binary data sets [Dimitriadou et al., 2002a]. Maximum clustering similarity method using indices of Rand, Fowlkes and Mallows, and Kulczynski proposed by [Albatineh and Niewiadomska-Bugaj, 2011] to determine the number of clusters based on the similarity between partitions. More studies using clustering validity indexes can be found in [Milligan and Cooper, 1986, Jain and Dubes, 1988, Kryszczuk and Hurley, 2010, Žalik, 2010, Wang and Zhang, 2007, Chae et al., 2006]

A nonparametric method based on distortion, which measures the average distance between each data point and its closest cluster center is proposed by [Sugar and James, 2011]. A new clustering validity evaluation based on risk computed by loss function and possibilities along with a new hierarchical clustering algorithm is proposed by [Yu et al., 2014]. The idea is coming from extension of the decision-theoretic rough set model to clustering, and it automatically estimates the number of clusters with a much smaller time cost. Several studies, for example [Tibshirani and Walther, 2005, Levine and Domany, 2001, Ben-Hur et al., 2001, Mufti et al., 2005] and [Bertrand and Mufti, 2006], propose that cluster stability is a good way to estimate number of clusters of any partitioning of the data. Fang and Wang develop a new estimation method for clustering instability based on the bootstrap, and the number of clusters is selected so that the corresponding estimated clustering instability is minimized [Fang and Wang, 2012]. A novel method based on cross-validation proposed by [Wang, 2010]. The key idea is to estimate the number of clusters that reduces the algorithm's instability. Also, this approach applies to both distance based and non-distance based algorithms.

Some studies in the literature use Bayesian Information Criterion (BIC) and Akaike's information criterion (AIC) in the context of likelihood function to estimate the correct number of cluster.

The AIC criterion is a measure of the relative quality of statistical models for a given data set was introduced by Bozdogan and Slove [Bozdogan and Sclove, 1984]. Some studies, for example, [Bondarenko et al., 1994, Koziol, 1990], use AIC criterion to estimate the correct number of cluster (e.g., determining the right number of clusters of tumor types with similar profiles of cell surface antigens). BIC is a criterion for model selection among a finite set of models was introduced by [Schwarz et al., 1978] is one of the commonly used criteria for determining the number of clusters. More studies using BIC criterion can be found in [Ishioka, 2005, Zhao et al., 2008, Cheong and Lee, 2008].

As we mentioned above, clustering validity indexes in conjunction with a proper clustering algorithm is a commonly used technique to estimate the correct number of cluster. However, the solution of clustering method is not stable across algorithms. Combining solution of individual clustering method, which is called consensus clustering or ensemble learning, might give more robust and consistent partition regardless of data structure [Topchy et al., 2005, Strehl and Ghosh, 2003, Lancichinetti and Fortunato, 2012]. Even though consensus clustering provides a better partition in terms of robustness and consistency, prior assumption that each individual clustering techniques have equal contribution has no basis. Xanthopulos and Unlu proposed weighted consensus clustering based on internal validity measure to handle with this problem and they successfully improve traditional consensus clustering. Since weighted consensus clustering might give better partition by remaining consistency regardless of data set, it might also yield the number of clusters tends to be less diverse.

From this perspective, we use three well-known indexes described in Chapter 3 which are SH, CH, and DB in conjunction with weighted consensus clustering to estimate the number of clusters and compare results with the k-means algorithm using same indexes. Additionally, we also use Consensus Index that is proposed by [Vinh and Epps, 2009] to determine the number of the cluster by consensus clustering and compare results with traditional consensus clustering.

For a given dataset $X = \{(x_i)\}_{i=1}^{N}$ where $x_i \in \mathbb{R}^n$, N and n number of samples and features, respectively. For a given particular number of clusters $(k \quad | \quad k = 2, \cdots, k_{max})$, suppose we have created a set of clustering solutions P^k for each k by a chosen method based on Equation 4.1a. Then, we can use following procedure given in equations 4.1b to determine correct or the most suitable number of clusters.

$$P^k = CM(X,k) \tag{4.1a}$$

$$k^* = \begin{cases} \arg\max_{k=2,\cdots,k_{max}}(Ind(X,P^k)), & \text{if max better index value refer better partition.} \\ \arg\min_{k=2,\cdots,k_{max}}(Ind(X,P^k)), & \text{if min better index value refer better partition.} \end{cases} \tag{4.1b}$$

where CM is chosen clustering method that returns P^k for a given dataset X with parameter k, Ind shows selected validity index, and k^* is the optimum number of the cluster which is determined based on optimum index value computed for given k.

Through our study we use four different index values, which are Consensus Index (CI) [Vinh and Epps, 2009],SH, CH, and DB, and five different clustering methods, which are K-means [MacQueen et al., 1967], consensus clustering based on Cluster-Based Similarity Algorithm (CSPA), [Strehl and Ghosh, 2003], and three weighted consensus clustering methods based on internal validity measures which are WConSH, WConCH, and WConDB. Those three indexes SH, CH, and DB indexes are also used to estimate the number of clusters. In the chapter 3, we have already explained details and the methodology of chosen indexes and methods. Thus, here we give only the methodology of CI.

Consensus index (CI)

The idea of consensus index has emerged from the consensus clustering method. It is proposed by [Vinh and Epps, 2009] aims to compute the similarity between different partitions. These partitions for a given number of cluster k can be obtained by running either single algorithm or different algorithms n times. Suppose we are generated multiple clustering solutions $P = (P_q^k \quad | \quad q = 1, \cdots, C)$ each with k clusters. Then, we can compute the similarity between different partitions based on the following equation.

$$CI(P^k) = \sum_{i<j} AM(P_i^k, P_j^k) \tag{4.2}$$

where CI is the consensus index and AM is a suitable similarity index. Thus, the CI computes the average similarity between all pairs of clustering solutions in a clustering set P^k. Then, optimum number of cluster k^* is the one that maximize CI:

$$k^* = \underset{k=2\cdots k_{max}}{\arg\max}\, CI(P^k) \tag{4.3}$$

Finally, we need to choose any index for CI that compute the similarity between partitions. Like original study, we use Adjusted Rand Index (ARI) [Hubert and Arabie, 1985], which is a similarity index based on pairs counting.

Results and Discussion

In this section, we present a comparison of k-means clustering, consensus clustering and three weighted consensus clusterings, which are WConSH, WConCH, and WConDB which are de-

scribed in chapter 3. Since CI is used within the context of consensus clustering, we only compare consensus clustering and weighted consensus clusterings as using CI. On the other hand, we add the k-means algorithm to our comparison when we use other three indexes SH, CH, and DB. We conduct the experiment on 20 different data sets which are given in Table 3.2 to evaluate performance given methods with respect to determining correct or the most suitable number of clusters. We refer the closest number of cluster to the correct number of the cluster by 1 as the most appropriate number of clusters. Again among those datasets Letter_IJL consisting of letters I, J, and L and MNIST_123 composed of digits 1, 2, and 3 are randomly sampled from Letter and MNIST data sets.

Results

Table 4.1 shows how many times given clustering methods estimated the best or second best number of clustering by using CI index. The performance of consensus clustering is quite weak in comparison to weighted consensus clusterings. We observe that CI index does not work well in a high number of clusters (e.g. greater than 4) for chosen data sets. On the other hand, one drawback of CI index is that it still returns high index value while two partitions are quite similar to each other despite the fact that they are dissimilar to original partition. This might cause to predict a wrong number of clusters.

Table 4.1: Comparison of consensus and weighted consensus clusterings as using CI index to determine correct or the most suitable number of cluster.

Methods	Best	Second Best	Total
Consensus	2	7	9
WConSH	9	5	14
WConCH	8	6	14
WConDB	7	8	15

Following Tables 4.2 and 4.3 illustrate the number of times given clustering methods returns correct or the most suitable number of clusters. It can be seen that in Table 4.2, in terms of predicting correct number of cluster weighted consensus frameworks give better results than k-means and consensus clustering regardless of the chosen index. Among all methods, WConCH shows the best performance by using CH index. It successfully predict correct number of clusters in 11 datasets out of 20.

Table 4.2: Comparison of k-means, consensus and weighted consensus clusterings along with SH, CH, and DB indexes to determine the correct number of clusters.

Methods	Silhouette	Calinski-Harabasz	Davies-Bouldin
k-means	9	7	7
Consensus	7	7	6
WConSH	9	10	8
WConCH	**10**	**11**	**9**
WConDB	**10**	10	**9**

Concerning second best prediction, regular consensus clustering gives slightly better results than other algorithms.

Table 4.3: Comparison of k-means, consensus and weighted consensus clusterings along with SH, CH, and DB indexes to determine the most suitable number of clusters.

Methods	Silhouette	Calinski-Harabasz	Davies-Bouldin
k-means	4	7	7
Consensus	**7**	**8**	**8**
WConSH	6	6	7
WConCH	6	5	7
WConDB	6	7	7

For better comparison, we provide total results in Table 4.4. Regardless of validity index weighted consensus clusterings outperform k-means and consensus clustering. Among weighted consensus

frameworks, we can say that WConDB shows the best performance in conjunction with CH index. More specifically, WConDB provide the correct or the most suitable number of the cluster in 17 datasets out of 20 which is quite good performance.

Table 4.4: Comparison of k-means, consensus and weighted consensus clusterings when using SH, CH, and DB indexes to determine a correct or the most suitable number of clusters.

Methods	Silhouette	Calinski-Harabasz	Davies-Bouldin
k-means	13	14	14
Consensus	14	15	14
WConSH	15	16	15
WConCH	**16**	16	**16**
WConDB	**16**	**17**	**16**

Conclusion

Determining the number of clusters is an important and necessary step in cluster analysis. Weighted consensus clustering in conjunction with internal validity index is proposed in this study to estimate correct or the most suitable number of cluster. Through our experiment, we compare the performance of weighted framework with k-means and consensus clusterings as using different types of indexes. Based on the experiment in 20 datasets, weighted consensus clusterings gives better results than other methods regardless of chosen indexes. The capability of working with any index is a profound advantage of weighted consensus clustering scheme. So that, it can be used for any data structure without spending the effort to find a proper index. For future research, it is worth to give attention to applying some other methods along with weighted consensus frameworks. By doing this, we might receive the greater number of best prediction and less second best prediction than proposed study.

CHAPTER 5: A NOVEL WEIGHTING POLICY FOR UNSUPERVISED ENSEMBLE LEARNING BASED ON MARKOWITZ PORTFOLIO THEORY

Introduction

In the Chapter 3, we have used internal validity index values itself to combine partitions with different weights. However, while we are doing this process, the variance of index values which can be an important performance measure is neglected. Assign the high weight to a noise partition might increase the variance of the overall results. Table 5.1 show a hypothetical example in which each row correspond particular index value of a partition given by a method. As it can be seen, among the index values of the first partitions, partition produced by Method-1 is the highest one. In other words, in the combination process, it will take the highest weight. However, if we look at the overall results, we can see that this is just an exemption. In general, the index value of the partitions produced by the Method-1 one is lower than others. This might cause a high variability in accuracy performance of regular weighted consensus clustering proposed in Chapter 3.Therefore, we proposed a new method to take variability of calculated index values across the iterations into consideration.

Methodology

In the weighting policy described in the Chapter 3, weights are the particular index value of the partition. Intrinsically, we can conclude that number of iteration times index values are calculated. In other words, each partition of a particular method has its own weight in the combination process.

Table 5.1: Hypothetically produced index values for each partition by different methods.

Method-1	Method-2	Method-3	Method-4
1.000	0.474	0.585	0.637
0.102	0.563	0.571	0.645
0.102	0.545	0.512	0.660
0.105	0.534	0.600	0.712
0.107	0.544	0.571	0.641
0.102	0.590	0.501	0.650
0.103	0.663	0.602	0.687
⋮	⋮	⋮	⋮

Here, we change this policy and calculated a single weight for the single method instead of the partition by using all assessed index values. To do this, we apply Markowitz portfolio theory to produce optimum weight. In the following section, we respectively introduce Markowitz portfolio theory and its implementation into our study.

Markowitz Portfolio Theory

An asset can be defined as a resource with an economic value which can be sold and bought. From an investor point of view, the key goal is to make a profit from an asset as much as possible. This expected profit is considered as the return of the asset. As shown in Figure 5.1, if one invest the amount of money (M_0) at the time of t_0, it expected to become the amount of money (M_1) at the time of t_1. Clearly, the expected total return (R) and expected rate of return (r) can be calculated as in Equations 5.1 and 5.2, respectively.

$$\text{Total Return} = R = \frac{M_1}{M_0} \tag{5.1}$$

Figure 5.1: Illustration of the expected return.

$$\text{Rate of return} = r = \frac{M_1 - M_0}{M_0} \tag{5.2}$$

Consider that as an investor, multiple assets are available and you would like to invest each of them by apportioning the money you have. In this case, a master asset -or portfolio- can be formed. The $(M_{0i} \quad | \quad i = 1, 2, \ldots, n)$ now represent the amount of money invested in asset i such that

$$\sum_{i=1}^{n} M_{0i} = M_0 \tag{5.3}$$

Here, the amount invested M_0 can be written as the fraction of total investment such that

$$M_{0i} = w_i M_0, \quad i = 1, 2, \ldots, n \tag{5.4}$$

And clearly,

$$\sum_{i=1}^{n} w_i = 1 \tag{5.5}$$

Another important property of the portfolio is the total expected return. Suppose that we have an n assets with rates of returns $r_1, r_2, \cdots, r_n$. These have expected returns $E(r_1) = \bar{r_1}, E(r_2) = \bar{r_2}, \cdots, E(r_n) = \bar{r_n}$. Suppose now we have create a portfolio of these n assets using the weights

$(w_i|i = 1,2,\cdots,n)$. The rate of the return of the portfolio can be calculated as

$$r = w_1 r_1 + w_2 r_2 + \cdots + w_n r_n \tag{5.6}$$

If we take the expected values of both sides and using linearity propoerty, we obtain:

$$E(r) = w_1 E(r_1) + w_2 E(r_2), \cdots, w_n E(r_n) \tag{5.7}$$

Here we use the term "expected" since an investor should face off some risk. Unless there is a riskless investment, it is not a realistic situation to profit -or lose- money every time from an investment. Therefore, one need to take the risk of portfolio into consideration for an investment decision. The risk of a portfolio is considered as the variance of the portfolio(σ^2) and it is calculated as in Equation 5.8.

$$\sigma^2 = \sum_{i,j=1}^{n} w_i w_j \sigma_{ij} \tag{5.8}$$

where σ_{ij} is the covariance of the return of asset i with asset j

Now we have fundamental terms concerning portfolio. However, the question is how a source needs to apportioned to optimize expected return based on a given level of market risk, defined as variance. In investment theory, we know the fact that higher risk is associated with greater probability of higher return and lower risk with a greater probability of smaller return. This trade-off which an investor faces between risk and return while considering investment decisions. In other words, one need to consider the risk and the expected return to maximize profit simultaneously. Markowitz -or Modern- portfolio theory (MPT) was proposed by Harry Markowitz in 1952

[Markowitz, 1952] to deal with this risk and return trade-off. The main objective of MPT is to form an optimum portfolio to maximize return with respect to given market risk.

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\sum_{i,j=1}^{n} w_i w_j \sigma_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{n} w_i \bar{r}_i = \bar{r} \\ & \sum_{i=1}^{n} w_i = 1 \end{aligned} \tag{5.9}$$

The solution of this formulation yields optimum w_i values. Since there is a no non-negativity constraint for w_i, it can be either negative or positive. This corresponds another fact in the economy which is called *short selling*. Short selling is the sale of security that is not owned by the seller, or that the seller has borrowed. In other words, if the short selling is allowed, a negative weight can be given to an asset. Here we do not go into detail of this concept, but we will explain in the following section how and why we did not allow the short selling.

Produce Weights Based on Markowitz Portfolio Theory

In this section, we explain how the weights are produced by using portfolio theory from the clustering perspective. These weights will be utilized later in the combination process within consensus clustering process.

The Table 5.2 illustrates hypothetically calculated index values of different partitions produced by different methods. We can consider methods as the assets in our ”portfolio” and index values as the returns of the assets. Therefore, we have all key inputs to create portfolio model.

Table 5.2: Interpreting algorithms and results of them based on portfolio theory.

	Method-1	Method-2	Method-3	Method-4
Expected Returns	1	0.474	0.585	0.637
	0.102	0.563	0.571	0.645
	0.102	0.545	0.512	0.660
	0.105	0.534	0.600	0.712
	0.107	0.544	0.571	0.641
	0.102	0.590	0.501	0.650
	0.103	0.663	0.602	0.687
	⋮	⋮	⋮	⋮
	Assests			

As we mention in the previous chapter, allowing short selling will create flexibility to assign negative weights to an asset. Here, since the asset corresponds to a partition produced by a clustering method, if we allow short selling, the partition will be able to given a negative weight which dramatically drops consensus performance down as pointed out in Chapter 3. Therefore, we need to add an extra constraint ($w_i \geq 0 \quad \text{for} \quad i = 1,2,\ldots,n$) to avoid negative weight as shown below.

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\sum_{i,j=1}^{n} w_i w_j \sigma_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{n} w_i \bar{r}_i = \bar{r} \\ & \sum_{i=1}^{n} w_i = 1 \\ & w_i \geq 0 \quad \text{for} \quad i = 1,2,\ldots,n \end{aligned} \tag{5.10}$$

At the moment we are ready to construct our process. The whole process can be divided into two stages. In the first stage, we first need to create multiple partitions to combine. To do this, we use

five different clustering methods which are Fuzzy, Hierarchical, Gaussian, k-means, and spectral as we did in Chapter 1. However, we apply these methods to randomly sampled %70 of the data instead of all data. As we pointed out before, high risk is associated with high return and vice versa. This fact from an investment point of view might not be valid in clustering framework. Although the result of a method is poor, its index value can be better in comparison to other methods. In another case, no matter how many times the algorithm is run, we might obtain an exactly same result which can be described as a riskless method. This kind of results might not be informative about existing data and also not useful in case adding new data samples. To avoid all these limitations and increase the usage validity of the weights in different cases, we randomly choose %70 of the data in each iteration.

Afterward, chosen indexes which are SH, CH, and DB index values as we choose again in Chapter 3 are calculated for the produced partitions. These index values will form the expected returns of the algorithms which are used as the input of the portfolio theory to obtain optimum weights for each algorithm. The general concept of the first part of the whole process is illustrated in Figure 5.2.

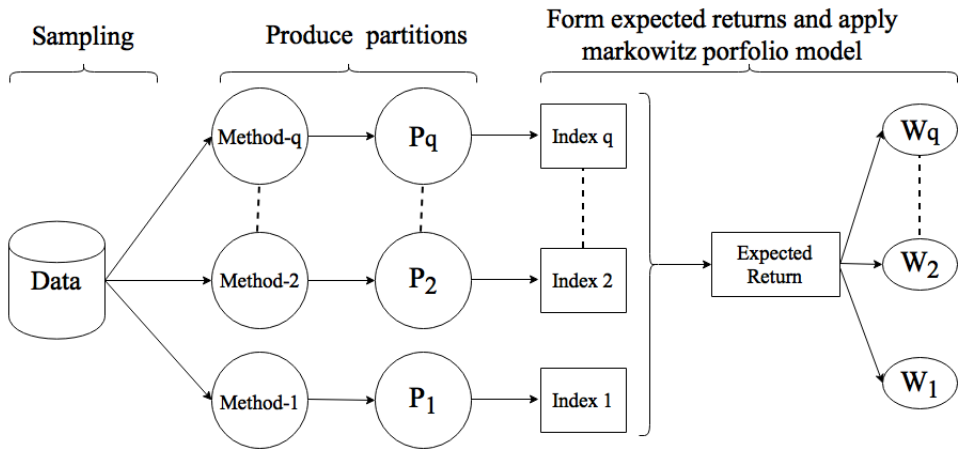

Figure 5.2: The first stage of Markowitz portfolio theory based weighted consensus clustering

In the second stage, we will follow the similar procedure we apply in weighted consensus clustering proposed in Chapter 3 except that the weights now come from the first stage as the result of the implementation of the Markowitz portfolio theory. Another difference is that in our case the weights will be used globally that means every single partition produced over iterations will be combined by using same weight. We illustrate the second stage in Figure 5.3. One needs to note that, we do not take samples from data set, but we cluster all data to obtain final partition as it should be. In the following section, we give results and possible future directions.

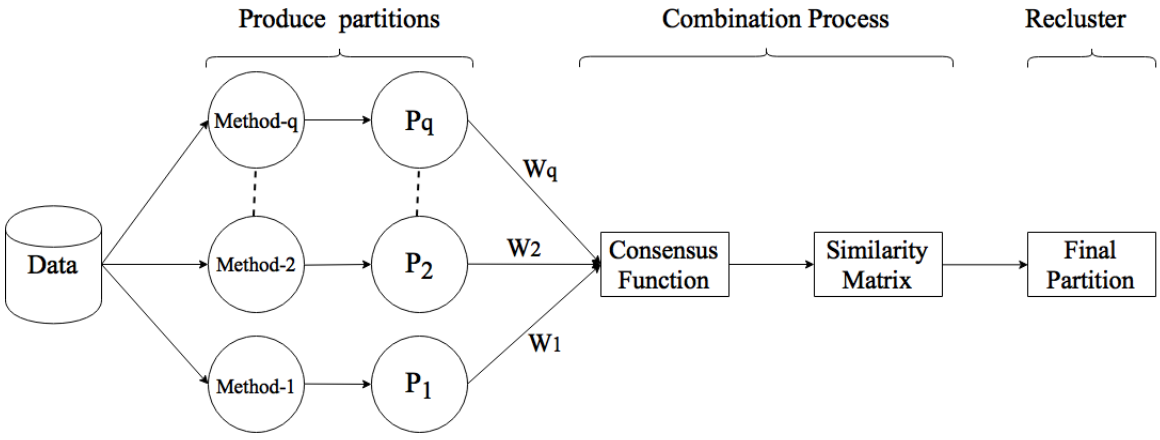

Figure 5.3: The second stage of Markowitz portfolio theory based weighted consensus clustering

Results and Discussion

In this section, we present experiment results of consensus, weighted consensus based on internal validity measures consensus and proposed Markowitz portfolio based weighted consensus clusterings as shown in Table 5.3. We conduct experiments on 20 different datasets to evaluate the performance of compared methods. In section 3, Table 3.2 gives the details of those 20 datasets. All datasets are used as found in the original repositories. Only exceptions are the dataset Letter_IJL

that consists of capital English letters I, J, and L and MNIST_123 that consists of handwritten digits 1, 2, and 3 are randomly sampled from Letter and MNIST datasets. If needs, datasets features are initially normalized prior to clustering so that they have 0 mean and unitary standard deviation. We performed the experiment on Intel Core i5, 2.3 GHz with 8 Gb of RAM on a 64-bit platform. Also, all codes are developed in Matlab version 2014a.

Table 5.3: Compared methods and corresponding indexes used as weight

Methods	Weights
Consensus	-
WConSH	Using Silhouette index
WConCH	Using Calinski-Harabasz index
WConDB	Using Davies-Bouldin index
MWConSH	Using Silhouette index
MWConCH	Using Calinski-Harabasz index
MWConDB	Using Davies-Bouldin index

Results

The Table 5.4 shows the results of algorithms for Iris dataset. As shown, seven different algorithms are compared regarding 5 different performance measures which are accuracy, variance, SH, CH, and DB indexes. Our core objective is to reduce the variance of the previously proposed methods WConSH, WConCH, and WConDB. We compare only each pair of methods which both are using the same index as weight (e.g., WConSH vs. MWConSH). Also, we compare proposed methods with traditional consensus clustering .

As pointed out, our priority is to reduce performance variance of previously proposed methods. While doing this, improving accuracy performance and other index values is considered as a further improvement.

Table 5.4: Results of algorithsm for Iris dataset.

Algorithms	Accuracy	Variance	Silhouette	Calinski-Harabasz	Davies-Bouldin
Consensus	84.71	199.48	0.62	373.74	0.98
WConSH	87.93	120.41	0.83	392.30	0.96
WConCH	91.31	50.90	0.72	448.41	0.73
WConDB	86.80	143.43	0.62	382.96	0.97
MWConSH	90.04	76.23	0.66	417.18	0.90
MWConCH	70.76	28.99	0.32	120.83	1.16
MWConDB	90.27	99.91	0.68	428.79	0.82

In given results (see Table 5.4) , for example, MWConSH and MWConDB not only reduce the variance of WConSH and MWConCH, but improve in terms of the accuracy, SH, and CH. On the other hand, MWConCH reduces the variance of the WConCH, but it reduces the accuracy of WConCH about %20 as well. In this point, the question is how we can evaluate this performance of method? We believe that it depends on sacrifice limit of the decision maker from the accuracy. Therefore, we need to assess the performance of these methods based on some threshold values(%5, %3, and %1) which show our sacrifice limit from accuracy. As the result of using these threshold values and since producing more accurate partition is not our priority, we have different conditions regarding accuracy when we compare proposed methods with traditional consensus clustering and weighted consensus clustering as shown in the very left column of Tables 5.5, 5.6, and 5.7.

The overall performance of proposed methods versus the traditional consensus clustering and previously proposed weighted consensus clustering methods based on threshold values is summarized in Tables 5.5 and 5.6.

The values in Table 5.5 show that in how many datasets Markowitz based proposed methods can reduce the variance of the accuracy of the regular weighted consensus methods with respect to

specified threshold values and given condition which is the accuracy of Markowitz based methods might be less than or equal to regular consensus and regular weighted consensus methods. For example, MWConSH reduced the variance of the accuracy of WConSH in 16 datasets out of 20 if at most %5 sacrifice from both accuracies of consensus and the regular weighted consensus is fine, in 14 datasets if at most %1 sacrifice from accuracy is fine, and so on. In general, we can say that while SH and DB work well, the performance of CH is poor when we use them in portfolio theory to create weights.

Table 5.5: Comparison of Markowitz based methods with regular weighted consensus methods.

Condition	Methods	%5	%3	%1
Accuracy of MWConsensus	MWConSH vs. WConSH	16	16	14
might be $\leq$ accuracy of	MWConCH vs. WConCH	8	8	5
Consensus and WConsensus	MWConDB vs. WConDB	15	14	12

In Table 5.6, we change the condition. Now, the accuracy of Markowitz based methods must be greater than or equal to regular consensus but might be less than or equal to regular weighted consensus methods. In this case, threshold values are useless. In other words, to reduce the variance of accuracy, we need to sacrifice less than just %1 from the accuracy of regular weighted consensus clustering.

Table 5.6: Comparison of Markowitz based methods with regular weighted consensus methods.

Condition	Methods	%5	%3	%1
Accuracy of MWConsensus must	MWConSH vs. WConSH	13	13	13
be $\geq$ accuracy of Consensus, but	MWConCH vs. WConCH	3	3	3
might be $\leq$ WConsensus	MWConDB vs. WConDB	12	12	12

The Table 5.7 shows overall results under another condition. This time our condition is that the accuracy of Markowitz based methods must be greater than or equal to regular consensus and

regular weighted consensus methods. This means that threshold value is 0. Again SH and DB work well to be treated as the expected returns from portfolio theory point of view, but CH is not a suitable index based on the results.

Table 5.7: Comparison of Markowitz based methods with regular weighted consensus methods.

Condition	Methods	%0
Accuracy of MWConsensus	MWConSH vs. WConSH	11
must be $\geq$ accuracy of	MWConCH vs. WConCH	2
Consensus and WConsensus	MWConDB vs. WConDB	10

On the other hand, we compare methods in term of index performance. Overall results are shown in Table 5.8. We again compare each pair of methods which use the same index. The values show that how many times Markowitz based consensus clustering method gives better performance than regular weighted consensus clustering on the performance of chosen index out of 20 datasets.

Besides all these conclusion detailed results for each dataset are given in Tables 5.9 and 5.10.

Table 5.8: Comparison of Markowitz based methods with regular weighted consensus methods regarding chosen index performance.

Methods	Silhouette	Calinski-Harabasz	Davies-Bouldin
MWConSH vs. WConSH	14	13	12
MWConCH vs. WConCH	8	8	9
MWConDB vs. WConDB	10	11	10

Table 5.9: Performance of regular consensus, weighted consensus methods (WCconSH, WConCH, and WConDB), and Markowitz based consensus methods(MWCconSH, MWConCH, and MW-ConDB) for given data sets in terms of particular evaluation metrics (EM).

Datasets	EM	Consensus	WConSH	WConCH	WConDB	MWConSH	MWConsCH	MWConDB
	Acc	75.50	75.40	73.90	77.20	75.60	74.90	76.50
	Var	123.79	50.71	54.30	83.63	29.61	53.90	32.01
Aggregation	SH	0.52	0.48	0.45	0.49	0.46	0.48	0.52
	CH	976.86	822.14	835.58	860.93	846.87	884.14	1032.07
	DB	0.94	1.13	1.70	1.03	1.24	1.00	0.96
	Acc	78.10	78.10	78.90	78.10	75.50	75.70	75.50
	Var	0.78	0.78	0.51	0.78	0.00	0.92	0.00
Appendicities	SH	0.58	0.58	0.59	0.58	0.52	0.52	0.52
	CH	70.79	70.79	71.39	70.79	64.51	60.44	64.51
	DB	1.05	1.05	1.04	1.05	1.13	1.09	1.13
	Acc	87.20	87.10	87.90	87.00	95.10	95.10	94.10
	Var	38.85	45.91	52.58	44.70	0.00	0.00	6.81
Balance	SH	0.28	0.28	0.29	0.28	0.30	0.30	0.30
	CH	125.46	126.21	128.91	126.04	133.76	133.76	131.82
	DB	2.05	2.05	2.03	2.05	2.00	2.00	2.01
	Acc	59.15	59.10	59.30	59.23	58.75	58.75	59.50
	Var	0.15	0.15	0.12	0.14	0.00	0.00	0.00
Banknote	SH	0.60	0.60	0.60	0.60	0.60	0.60	0.60
	CH	418.52	418.58	418.34	418.43	419.06	419.06	418.18
	DB	0.89	0.89	0.89	0.89	0.89	0.89	0.89
	Acc	91.50	92.50	92.60	91.50	91.80	91.80	91.80
	Var	36.04	0.12	0.22	36.14	0.00	0.00	0.00
Breast	SH	0.47	0.48	0.48	0.47	0.47	0.47	0.47
	CH	295.12	302.01	302.00	296.35	299.36	299.36	299.36
	DB	1.43	1.42	1.42	1.43	1.43	1.43	1.43
	Acc	56.00	56.70	57.30	58.00	51.40	54.70	53.50
	Var	47.14	37.61	42.35	40.33	16.43	32.49	32.21
Compound	SH	0.40	0.39	0.36	0.43	0.40	0.55	0.39
	CH	522.75	558.37	520.37	609.23	532.53	701.24	513.83
	DB	1.11	1.27	1.42	1.09	1.06	0.89	1.14
	Acc	86.00	89.90	88.70	87.30	91.20	60.50	87.90
	Var	108.34	24.98	58.33	79.93	0.00	16.66	73.94
Ecoli	SH	0.53	0.59	0.54	0.55	0.61	0.24	0.56
	CH	191.67	216.01	206.09	202.57	225.79	30.09	206.17
	DB	1.22	0.98	1.13	1.12	0.92	2.02	1.00
	Acc	46.80	49.90	51.90	50.80	50.10	35.70	50.00
	Var	26.30	17.30	10.00	22.10	12.10	5.10	14.40
Glass	SH	0.11	0.17	0.22	0.20	0.17	-0.31	0.14
	CH	60.74	75.77	79.72	70.87	73.07	12.18	74.63
	DB	2.13	1.63	1.51	1.69	1.23	3.69	1.49
	Acc	54.70	55.00	54.00	55.20	56.10	44.00	56.10
	Var	23.61	18.78	30.64	9.84	4.41	32.18	4.41
Letter_IJL	SH	0.26	0.28	0.30	0.29	0.29	0.16	0.29
	CH	86.28	90.55	97.92	95.23	91.91	58.14	93.53
	DB	1.77	1.70	1.63	1.64	1.69	3.60	1.67
	Acc	51.40	50.59	53.13	50.20	55.72	56.22	53.13
	Var	9.79	1.00	0.01	0.01	0.00	14.33	0.01
Liver	SH	0.61	0.70	0.78	0.69	0.83	0.33	0.78
	CH	203.72	242.72	309.75	232.74	326.24	76.42	309.75
	DB	1.04	1.00	0.84	1.03	0.76	1.21	0.84

Table 5.10: Performance of regular consensus, weighted consensus methods (WCconSH, WConCH, and WConDB), and Markowitz based consensus methods(MWCconSH, MWConCH, and MWConDB) for given data sets in terms of particular evaluation metrics (EM).

Datasets	EM	Consensus	WConSH	WConCH	WConDB	MWConSH	MWConsCH	MWConDB
	Acc	58.30	62.30	77.00	76.10	76.90	46.00	76.10
	Var	159.71	214.06	151.40	161.90	144.41	103.98	149.70
MNIST_IJL	SH	0.09	0.14	0.23	0.24	0.23	0.02	0.23
	CH	36.01	43.28	64.62	63.92	66.93	13.43	64.71
	DB	6.81	6.81	2.46	2.77	2.31	9.86	2.33
	Acc	82.90	77.40	73.00	74.20	72.00	73.00	73.70
	Var	25.76	75.85	46.23	1.75	48.81	12.39	9.03
Pathbased	SH	0.57	0.64	0.65	0.71	0.67	0.69	0.67
	CH	221.99	286.19	308.78	345.06	320.79	330.11	318.58
	DB	1.55	1.09	0.82	0.74	0.85	0.72	0.77
	Acc	67.90	69.20	66.50	67.50	69.20	67.90	68.40
	Var	76.91	62.36	87.94	87.08	52.05	74.91	55.91
Soybean	SH	0.42	0.44	0.41	0.41	0.45	0.41	0.43
	CH	27.52	29.82	27.64	26.86	29.79	28.55	28.12
	DB	1.41	1.34	1.38	1.39	1.40	1.54	1.32
	Acc	58.03	58.07	58.07	58.08	57.17	55.38	57.17
	Var	0.23	0.16	0.18	0.17	0.66	0.00	0.38
Yeast	SH	0.28	0.28	0.28	0.28	0.28	0.28	0.28
	CH	186.41	186.46	186.57	186.48	183.81	180.35	183.82
	DB	1.94	1.94	1.94	1.94	1.95	1.96	1.94
	Acc	87.60	87.20	86.20	87.90	84.60	84.60	85.10
	Var	14.91	9.98	6.18	13.27	0.50	0.50	0.25
Flame	SH	0.52	0.52	0.53	0.52	0.53	0.53	0.53
	CH	148.31	150.65	152.31	148.18	154.76	154.76	154.00
	DB	1.12	1.12	1.12	1.12	1.12	1.12	1.12
	Acc	84.70	86.00	82.70	83.60	86.60	53.90	87.30
	Var	152.58	98.39	152.97	163.68	74.82	168.63	61.10
Seeds	SH	0.51	0.58	0.48	0.51	0.62	0.30	0.62
	CH	310.91	333.45	293.47	305.16	347.98	88.09	356.11
	DB	1.16	1.03	1.27	1.15	0.82	1.08	0.83
	Acc	67.50	64.90	64.10	68.90	69.00	59.20	67.70
	Var	31.64	55.28	66.47	0.47	11.46	82.70	31.90
Wine	SH	0.67	0.44	0.43	0.64	0.68	0.48	0.63
	CH	380.57	292.48	283.33	377.23	398.61	202.61	369.76
	DB	0.61	2.56	2.49	0.62	0.61	0.95	0.62
	Acc	59.40	60.20	59.00	64.10	60.00	49.90	48.20
	Var	48.95	50.35	31.75	30.47	43.03	49.66	58.30
Zoo	SH	0.32	0.39	0.33	0.46	0.37	-0.16	-0.12
	CH	44.69	49.81	47.61	59.10	52.73	8.73	10.77
	DB	1.73	1.49	1.48	1.30	1.53	4.44	4.14
	Acc	92.84	85.41	85.41	87.07	85.41	85.41	85.41
	Var	3.38	0.00	0.00	11.30	0.00	0.00	0.00
WDBC	SH	0.68	0.83	0.83	0.80	0.83	0.83	0.83
	CH	768.15	1300.21	1300.21	1199.46	1300.21	1300.21	1300.21
	DB	0.67	0.50	0.50	0.53	0.50	0.50	0.50
	Acc	84.71	87.93	91.31	86.80	90.04	70.76	90.27
	Var	199.48	120.41	50.90	143.43	76.23	28.99	99.91
Iris	SH	0.62	0.63	0.72	0.62	0.66	0.32	0.68
	CH	373.74	392.30	448.41	382.96	417.18	120.83	428.79
	DB	0.98	0.96	0.73	0.97	0.90	1.16	0.82

Conclusion

In this study, we propose a novel weighting policy for unsupervised ensemble learning. We borrow the idea of Markowitz portfolio theory and implement it to our proposed weighted consensus clustering. Our key objective is to reduce the variance of the accuracy of traditional consensus clustering and regular weighted consensus clustering. We compare the results based on some threshold values that represent the sacrifice limit of the decision maker from the accuracy. According to the experimental results, proposed weighted consensus clustering outperforms traditional consensus clustering and regular weighted consensus clustering in the majority of the dataset in any threshold values.

Now, the question is that how a decision maker can know how much accuracy needs to be sacrificed to reduce the desired amount of variance. Therefore, for future research, an automated tool can be developed to help decision maker to tune up the correlation between accuracy and variance. Moreover, we use a straightforward and useful method to create optimum weights with expected index values, a more advanced method such as Conditional Value at Risk (CVAR) might be proposed in future to enhance our proposed methodology.

CHAPTER 6: CONCLUSION AND RECOMMENDATIONS

In this study, we first propose a weighting policy for unsupervised ensemble learning named consensus clustering based on internal validity measures. Our primary goal is to avoid treating each partition equally while we combine them. Due to the methodological foundation and different objective of clustering methods, they tend to produce different partitions with different qualities for a given dataset. From this perspective, we use internal validity measures as weight through combination process. According to the experimental result, our proposed method yields overall better performance than traditional consensus clustering regarding accuracy and chosen index values.

However, the base algorithm CSPA is not suitable for big datasets since its computational complexity quadratic in the number of samples n. While keeping the main idea of weighting policy same, we can use other graph methods such as HPGA its computational complexity linear in n. Besides implementing weighting policy into graph partitioning method, we can also choose another approach its time and computational complexity is lower than CSPA.

Next, since the weighted consensus clustering produces more consistent partition, we thought that it might have the ability to predict a better number of cluster. Therefore, we use it to determine the correct number of clusters. Based on our results using weighted consensus clustering with a proper validity index might show much better performance than a single algorithm and regular consensus clustering on finding the number of clusters. As we mentioned before, nevertheless, we need to take into consideration that applied method is the simplest -but commonly used- one to determine the number of clusters and it is not realistic to expect same high performance in various conditions. So that the performance of proposed method for the datasets with the high number of clusters is not as good as the performance for the datasets with the small number of clusters. Clustering practitioners can focus on to develop more advanced methods to determine the number

of clusters and we believe strong sides of ensemble learning approach can be used in conjunction with any suitable method.

On the other hand, we extend proposed weighted consensus clustering by implying Markowitz portfolio theory. We aim to reduce the variance of accuracy subject to the variation in the assessed validity index values. The results of the experiment show that the optimum weights can be produced by portfolio theory to reduce the variance of the regular weighted consensus clustering in the majority of the data sets. In addition to current proposed method, more advanced portfolio optimization methods such as Condition Value at Risk (CVAR) might be used to increase the better performance of proposed method.

Finally, CSPA has no explicit objective function so that it is not possible to consider it as an optimization problem. One can focus on to develop weighted ensemble learning as an optimization task for the future research. Also, we consider only hard partitions, but a fuzzy version of these methods can be developed by transforming a proper method such as sCSPA. The main difference will be the assignment strategy of data samples. Instead of assigning a data sample into exactly one group, we can calculate the degree of membership of each data sample to a cluster. Then, similarity matrix might represent the total degree of membership of two sample for a particular cluster.

LIST OF REFERENCES

[Abello et al., 2013] Abello, J., Pardalos, P. M., and Resende, M. G. (2013). *Handbook of massive data sets*, volume 4. Springer.

[Agrawal et al., 1993] Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM.

[Ahmed, 2004] Ahmed, S. R. (2004). Applications of data mining in retail business. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 2, pages 455–459. IEEE.

[Ailon et al., 2008] Ailon, N., Charikar, M., and Newman, A. (2008). Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23.

[Al-Razgan and Domeniconi, 2006] Al-Razgan, M. and Domeniconi, C. (2006). Weighted clustering ensembles. In *SDM*, pages 258–269. SIAM.

[Albatineh and Niewiadomska-Bugaj, 2011] Albatineh, A. N. and Niewiadomska-Bugaj, M. (2011). Mcs: A method for finding the number of clusters. *Journal of classification*, 28(2):184–209.

[Alizadeh et al., 2014] Alizadeh, H., Minaei-Bidgoli, B., and Parvin, H. (2014). Cluster ensemble selection based on a new cluster stability measure. *Intelligent Data Analysis*, 18(3):389–408.

[Ana and Jain, 2003] Ana, L. and Jain, A. K. (2003). Robust data clustering. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–128. IEEE.

[Ayad and Kamel, 2008] Ayad, H. G. and Kamel, M. S. (2008). Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):160–173.

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

[Azimi and Fern, 2009] Azimi, J. and Fern, X. (2009). Adaptive cluster ensemble selection. In *IJCAI*, volume 9, pages 992–997.

[Azimi et al., 2006] Azimi, J., Mohammadi, M., Analoui, M., et al. (2006). Clustering ensembles using genetic algorithm. In *Computer Architecture for Machine Perception and Sensing, 2006. CAMP 2006. International Workshop on*, pages 119–123. IEEE.

[Barlow, 1989] Barlow, H. B. (1989). Unsupervised learning. *Neural computation*, 1(3):295–311.

[Ben-Hur et al., 2001] Ben-Hur, A., Elisseeff, A., and Guyon, I. (2001). A stability based method for discovering structure in clustered data. In *Pacific symposium on biocomputing*, volume 7, pages 6–17.

[Berikov, 2014] Berikov, V. (2014). Weighted ensemble of algorithms for complex data clustering. *Pattern Recognition Letters*, 38:99–106.

[Berkhin, 2006] Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer.

[Bertrand and Mufti, 2006] Bertrand, P. and Mufti, G. B. (2006). Loevinger's measures of rule quality for assessing cluster stability. *Computational statistics & data analysis*, 50(4):992–1015.

[Bhojani and Bhatt, 2016] Bhojani, S. H. and Bhatt, N. (2016). Data mining techniques and trends–a review. *Global Journal For Research Analysis*, 5(5).

[Bock, 1985] Bock, H.-H. (1985). On some significance tests in cluster analysis. *Journal of classification*, 2(1):77–108.

[Bondarenko et al., 1994] Bondarenko, I., Van Malderen, H., Treiger, B., Van Espen, P., and Van Grieken, R. (1994). Hierarchical cluster analysis with stopping rules built on akaike's information criterion for aerosol particle classification based on electron probe x-ray micro-analysis. *Chemometrics and intelligent laboratory systems*, 22(1):87–95.

[Bozdogan and Sclove, 1984] Bozdogan, H. and Sclove, S. L. (1984). Multi-sample cluster analysis using akaike's information criterion. *Annals of the Institute of Statistical Mathematics*, 36(1):163–180.

[Brodersen et al., 2010] Brodersen, K. H., Ong, C. S., Stephan, K. E., and Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. In *Pattern recognition (ICPR), 2010 20th international conference on*, pages 3121–3124. IEEE.

[Cades et al., 2001] Cades, I., Smyth, P., and Mannila, H. (2001). Probabilistic modeling of transactional data with applications to profiling, visualization and prediction, sigmod. *Proc. of the 7th ACM SIGKDD. San Francisco: ACM Press*, pages 37–46.

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

[Carpineto and Romano, 2012] Carpineto, C. and Romano, G. (2012). Consensus clustering based on a new probabilistic rand index with application to subtopic retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 34(12):2315–2326.

[Chae et al., 2006] Chae, S. S., DuBien, J. L., and Warde, W. D. (2006). A method of predicting the number of clusters using rand's statistic. *Computational statistics & data analysis*, 50(12):3531–3546.

[Chandola and Kumar, 2007] Chandola, V. and Kumar, V. (2007). Summarization–compressing data into an informative representation. *Knowledge and Information Systems*, 12(3):355–378.

[Chang and Yeung, 2008] Chang, H. and Yeung, D.-Y. (2008). Robust path-based spectral clustering. *Pattern Recognition*, 41(1):191–203.

[Chen et al., 2006] Chen, Y., Zhang, G., Hu, D., and Wang, S. (2006). Customer segmentation in customer relationship management based on data mining. In *Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management*, pages 288–293. Springer.

[Cheong and Lee, 2008] Cheong, M.-Y. and Lee, H. (2008). Determining the number of clusters in cluster analysis. *Journal of the Korean Statistical Society*, 37(2):135–143.

[d Souto et al., 2006] d Souto, M., de Araujo, D. S., and da Silva, B. L. (2006). Cluster ensemble for gene expression microarray data: accuracy and diversity. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 2174–2180. IEEE.

[Davies and Bouldin, 1979] Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):224–227.

[de Hoon et al., 2004] de Hoon, M. J., Imoto, S., Nolan, J., and Miyano, S. (2004). Open source clustering software. *Bioinformatics*, 20(9):1453–1454.

[Deodhar and Ghosh, 2006] Deodhar, M. and Ghosh, J. (2006). Consensus clustering for detection of overlapping clusters in microarray data. In *ICDM Workshops*, pages 104–108.

[Dimitriadou et al., 2002a] Dimitriadou, E., Dolničar, S., and Weingessel, A. (2002a). An examination of indexes for determining the number of clusters in binary data sets. *Psychometrika*, 67(1):137–159.

[Dimitriadou et al., 2002b] Dimitriadou, E., Weingessel, A., and Hornik, K. (2002b). A combination scheme for fuzzy clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(07):901–912.

[Domeniconi and Al-Razgan, 2009] Domeniconi, C. and Al-Razgan, M. (2009). Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(4):17.

[Domeniconi et al., 2007] Domeniconi, C., Gunopulos, D., Ma, S., Yan, B., Al-Razgan, M., and Papadopoulos, D. (2007). Locally adaptive metrics for clustering high dimensional data. *Data Mining and Knowledge Discovery*, 14(1):63–97.

[Dudoit and Fridlyand, 2003] Dudoit, S. and Fridlyand, J. (2003). Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099.

[Dunn, 1973] Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.

[Esmin and Coelho, 2013] Esmin, A. A. and Coelho, R. A. (2013). Consensus clustering based on particle swarm optimization algorithm. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2280–2285. IEEE.

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

[Everitt et al., 2001] Everitt, B. S., Landau, S., and Leese, M. (2001). Cluster analysis arnold. *A member of the Hodder Headline Group, London*.

[Fang and Wang, 2012] Fang, Y. and Wang, J. (2012). Selection of the number of clusters via the bootstrap method. *Computational Statistics & Data Analysis*, 56(3):468–477.

[Fawcett and Provost, 1997] Fawcett, T. and Provost, F. (1997). Adaptive fraud detection. *Data mining and knowledge discovery*, 1(3):291–316.

[Fayyad et al., 1996] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37.

[Fern and Brodley, 2004] Fern, X. Z. and Brodley, C. E. (2004). Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the twenty-first international conference on Machine learning*, page 36. ACM.

[Ferris and Mangasarian, 1995] Ferris, M. and Mangasarian, O. (1995). Breast-cancer diagnosis via linear-programming.

[Fischer and Buhmann, 2003] Fischer, B. and Buhmann, J. M. (2003). Bagging for path-based clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1411–1415.

[Fred, 2001] Fred, A. (2001). Finding consistent clusters in data partitions. In *Multiple classifier systems*, pages 309–318. Springer.

[Fred and Jain, 2002] Fred, A. L. and Jain, A. K. (2002). Data clustering using evidence accumulation. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 4, pages 276–280. IEEE.

[Fred and Jain, 2005] Fred, A. L. and Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):835–850.

[Fu and Medico, 2007] Fu, L. and Medico, E. (2007). Flame, a novel fuzzy clustering method for the analysis of dna microarray data. *BMC bioinformatics*, 8(1):3.

[Ghaemi et al., 2009] Ghaemi, R., Sulaiman, M. N., Ibrahim, H., Mustapha, N., et al. (2009). A survey: clustering ensembles techniques. *World Academy of Science, Engineering and Technology*, 50:636–645.

[Gionis et al., 2007] Gionis, A., Mannila, H., and Tsaparas, P. (2007). Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):4.

[Giraud-Carrier and Povel, 2003] Giraud-Carrier, C. and Povel, O. (2003). Characterising data mining software. *Intelligent Data Analysis*, 7(3):181–192.

[Gluck, 1985] Gluck, M. (1985). Information, uncertainty and the utility of categories. In *Proc. of the Seventh Annual Conf. on Cognitive Science Society*, pages 283–287. Lawrence Erlbaum.

[Goder and Filkov, 2008] Goder, A. and Filkov, V. (2008). Consensus clustering algorithms: Comparison and refinement. In *Alenex*, volume 8, pages 109–117. SIAM.

[Gordon, 1999] Gordon, A. D. (1999). Classification, (chapman & hall/crc monographs on statistics & applied probability).

[Grira et al., 2004] Grira, N., Crucianu, M., and Boujemaa, N. (2004). Unsupervised and semi-supervised clustering: a brief survey. *A review of machine learning techniques for processing multimedia content*, 1:9–16.

[Gupta and Verma, 2014] Gupta, M. and Verma, D. (2014). A novel ensemble based cluster analysis using similarity matrices & clustering algorithm (smca). *International Journal of Computer Application*, 100(10):1–6.

[Hadjitodorov et al., 2006] Hadjitodorov, S. T., Kuncheva, L. I., and Todorova, L. P. (2006). Moderate diversity for better cluster ensembles. *Information Fusion*, 7(3):264–275.

[Haghtalab et al., 2015] Haghtalab, S., Xanthopoulos, P., and Madani, K. (2015). A robust unsupervised consensus control chart pattern recognition framework. *Expert Systems with Applications*.

[Halkidi et al., 2002] Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2002). Cluster validity methods: part i. *ACM Sigmod Record*, 31(2):40–45.

[Halkidi and Vazirgiannis, 2001] Halkidi, M. and Vazirgiannis, M. (2001). Clustering validity assessment: Finding the optimal partitioning of a data set. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 187–194. IEEE.

[Halkidi et al., 2000] Halkidi, M., Vazirgiannis, M., and Batistakis, Y. (2000). Quality scheme assessment in the clustering process. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '00, pages 265–276, London, UK, UK. Springer-Verlag.

[Hand et al., 2001] Hand, D. J., Mannila, H., and Smyth, P. (2001). *Principles of data mining*. MIT press.

[Hardy, 1996] Hardy, A. (1996). On the number of clusters. *Computational Statistics & Data Analysis*, 23(1):83–96.

[Hartigan, 1975] Hartigan, J. A. (1975). Clustering algorithms.

[Hong et al., 2008] Hong, Y., Kwong, S., Chang, Y., and Ren, Q. (2008). Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. *Pattern Recognition*, 41(9):2742–2756.

[Hu et al., 2005] Hu, X., Yoo, I., Zhang, X., Nanavati, P., and Das, D. (2005). Wavelet transformation and cluster ensemble for gene expression analysis. *International journal of bioinformatics research and applications*, 1(4):447–460.

[Huang et al., 2016a] Huang, D., Lai, J., and Wang, C.-D. (2016a). Ensemble clustering using factor graph. *Pattern Recognition*, 50:131–142.

[Huang et al., 2016b] Huang, D., Wang, C.-D., and Lai, J.-H. (2016b). Locally weighted ensemble clustering. *arXiv preprint arXiv:1605.05011*.

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

[Iam-On et al., 2012] Iam-On, N., Boongeon, T., Garrett, S., and Price, C. (2012). A link-based cluster ensemble approach for categorical data clustering. *IEEE Transactions on knowledge and data engineering*, 24(3):413–425.

[Iam-On and Boongoen, 2012] Iam-On, N. and Boongoen, T. (2012). Improved link-based cluster ensembles. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[Iam-on et al., 2010] 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):1513–1519.

[Ishioka, 2005] Ishioka, T. (2005). An expansion of x-means for automatically determining the optimal number of clusters. In *Proceedings of International Conference on Computational Intelligence*, pages 91–96.

[Jain and Dubes, 1988] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

[Jain et al., 1999] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

[Jang et al., 1997] Jang, J.-S. R., Sun, C.-T., and Mizutani, E. (1997). Neuro-fuzzy and soft computing; a computational approach to learning and machine intelligence.

[Jing et al., 2015] Jing, L., Tian, K., and Huang, J. Z. (2015). Stratified feature sampling method for ensemble clustering of high dimensional data. *Pattern Recognition*, 48(11):3688–3702.

[Johnson, 1967] Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.

[Jun Lee and Siau, 2001] Jun Lee, S. and Siau, K. (2001). A review of data mining techniques. *Industrial Management & Data Systems*, 101(1):41–46.

[Kang et al., 2016] Kang, Q., Liu, S., Zhou, M., and Li, S. (2016). A weight-incorporated similarity-based clustering ensemble method based on swarm intelligence. *Knowledge-Based Systems*, 104:156–164.

[Kantardzic, 2011] Kantardzic, M. (2011). *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons.

[Karypis et al., 1999] Karypis, G., Aggarwal, R., Kumar, V., and Shekhar, S. (1999). Multilevel hypergraph partitioning: applications in vlsi domain. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 7(1):69–79.

[Kennedy, 2011] Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer.

[Kotsiantis et al., 2006] Kotsiantis, S., Kanellopoulos, D., Pintelas, P., et al. (2006). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36.

[Kovács et al., 2005] Kovács, F., Legány, C., and Babos, A. (2005). Cluster validity measurement techniques. In *6th International symposium of hungarian researchers on computational intelligence*.

[Koziol, 1990] Koziol, J. (1990). Cluster analysis of antigenic profiles of tumors: selection of number of clusters using akaike's information criterion. *Methods of information in medicine*, 29(3):200–204.

[Křivánek and Morávek, 1986] Křivánek, M. and Morávek, J. (1986). Np-hard problems in hierarchical-tree clustering. *Acta Informatica*, 23(3):311–323.

[Kryszczuk and Hurley, 2010] Kryszczuk, K. and Hurley, P. (2010). Estimation of the number of clusters using multiple clustering validity indices. In *Multiple Classifier Systems*, pages 114–123. Springer.

[Kuncheva et al., 2006] Kuncheva, L. I., Hadjitodorov, S. T., and Todorova, L. P. (2006). Experimental comparison of cluster ensemble methods. In *Information Fusion, 2006 9th International Conference on*, pages 1–7. IEEE.

[Lancichinetti and Fortunato, 2012] Lancichinetti, A. and Fortunato, S. (2012). Consensus clustering in complex networks. *Scientific reports*, 2.

[LeCun and Cortes, 2010] LeCun, Y. and Cortes, C. (2010). Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*.

[Levine and Domany, 2001] Levine, E. and Domany, E. (2001). Resampling method for unsupervised estimation of cluster validity. *Neural computation*, 13(11):2573–2593.

[Li and Ding, 2008] Li, T. and Ding, C. (2008). Weighted consensus clustering. *Mij*, 1(2).

[Li et al., 2007] Li, T., Ding, C., and Jordan, M. I. (2007). Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 577–582. IEEE.

[Li et al., 2010] Li, T., Ogihara, M., and Ma, S. (2010). On combining multiple clusterings: an overview and a new perspective. *Applied Intelligence*, 33(2):207–219.

[Li et al., 2006] Li, T., Ogihara, M., and Zhu, S. (2006). Integrating features from different sources for music information retrieval. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 372–381. IEEE.

[Lichman, 2013] Lichman, M. (2013). UCI machine learning repository.

[Liu et al., 2015a] Liu, H., Cheng, G., and Wu, J. (2015a). Consensus clustering on big data. In *Service Systems and Service Management (ICSSSM), 2015 12th International Conference on*, pages 1–6. IEEE.

[Liu et al., 2015b] Liu, H., Liu, T., Wu, J., Tao, D., and Fu, Y. (2015b). Spectral ensemble clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 715–724. ACM.

[Liu et al., 2010] Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of internal clustering validation measures. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 911–916. IEEE.

[Lock and Dunson, 2013] Lock, E. F. and Dunson, D. B. (2013). Bayesian consensus clustering. *Bioinformatics*, page btt425.

[Lourenço et al., 2015] Lourenço, A., Bulò, S. R., Rebagliati, N., Fred, A. L., Figueiredo, M. A., and Pelillo, M. (2015). Probabilistic consensus clustering using evidence accumulation. *Machine Learning*, 98(1-2):331–357.

[Luo et al., 2006] Luo, H., Jing, F., and Xie, X. (2006). Combining multiple clusterings using information theory based genetic algorithm. In *2006 International Conference on Computational Intelligence and Security*, volume 1, pages 84–89. IEEE.

[Mabroukeh and Ezeife, 2010] Mabroukeh, N. R. and Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys (CSUR)*, 43(1):3.

[MacQueen et al., 1967] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[Markowitz, 1952] Markowitz, H. (1952). Portfolio selection. *The journal of finance*, 7(1):77–91.

[McLachlan and Peel, 2000] McLachlan, G. and Peel, D. (2000). Multivariate normal mixtures. *Finite Mixture Models*, pages 81–116.

[McQuitty, 1957] McQuitty, L. L. (1957). Elementary linkage analysis for isolating orthogonal and oblique types and typal relevancies. *Educational and Psychological Measurement*.

[Milligan and Cooper, 1985] Milligan, G. W. and Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179.

[Milligan and Cooper, 1986] Milligan, G. W. and Cooper, M. C. (1986). A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 21(4):441–458.

[Mirkin, 2001] Mirkin, B. (2001). Reinterpreting the category utility function. *Machine Learning*, 45(2):219–228.

[Mufti et al., 2005] Mufti, G. B., Bertrand, P., and Moubarki, E. (2005). Determining the number of groups from measures of cluster stability. In *Proceedings of international symposium on applied stochastic models and data analysis*, pages 17–20.

[Naldi et al., 2013] Naldi, M. C., Carvalho, A., and Campello, R. J. (2013). Cluster ensemble selection based on relative validity indexes. *Data Mining and Knowledge Discovery*, 27(2):259–289.

[Nayak et al., 2015] Nayak, J., Naik, B., and Behera, H. (2015). Fuzzy c-means (fcm) clustering algorithm: a decade review from 2000 to 2014. In *Computational Intelligence in Data Mining-Volume 2*, pages 133–149. Springer.

[Ng et al., 2002] Ng, A. Y., Jordan, M. I., Weiss, Y., et al. (2002). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856.

[Ngai et al., 2009] Ngai, E. W., Xiu, L., and Chau, D. C. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert systems with applications*, 36(2):2592–2602.

[Nisbet et al., 2009] Nisbet, R., Miner, G., and Elder IV, J. (2009). *Handbook of statistical analysis and data mining applications*. Academic Press.

[Parvin et al., 2013] Parvin, H., Minaei-Bidgoli, B., Alinejad-Rokny, H., and Punch, W. F. (2013). Data weighing mechanisms for clustering ensembles. *Computers & Electrical Engineering*, 39(5):1433–1450.

[Punera and Ghosh, 2008] Punera, K. and Ghosh, J. (2008). Consensus-based ensembles of soft clusterings. *Applied Artificial Intelligence*, 22(7-8):780–810.

[Rajaraman et al., 2012] Rajaraman, A., Ullman, J. D., Ullman, J. D., and Ullman, J. D. (2012). *Mining of massive datasets*, volume 77. Cambridge University Press Cambridge.

[Rashedi and Mirzaei, 2011] Rashedi, E. and Mirzaei, A. (2011). A novel multi-clustering method for hierarchical clusterings based on boosting. In *2011 19th Iranian Conference on Electrical Engineering*, pages 1–4. IEEE.

[Rashedi and Mirzaei, 2013] Rashedi, E. and Mirzaei, A. (2013). A hierarchical clusterer ensemble method based on boosting theory. *Knowledge-Based Systems*, 45:83–93.

[Ren et al., 2016] Ren, Y., Domeniconi, C., Zhang, G., and Yu, G. (2016). Weighted-object ensemble clustering: methods and analysis. *Knowledge and Information Systems*, pages 1–29.

[Rendón et al., 2011] Rendón, E., Abundez, I., Arizmendi, A., and Quiroz, E. (2011). Internal versus external cluster validation indexes. *International Journal of computers and communications*, 5(1):27–34.

[Rousseeuw, 1987] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

[Sadeghian and Nezamabadi-pour, 2014] Sadeghian, A. H. and Nezamabadi-pour, H. (2014). Gravitational ensemble clustering. In *Intelligent Systems (ICIS), 2014 Iranian Conference on*, pages 1–6. IEEE.

[Saeed et al., 2014] Saeed, F., Ahmed, A., Shamsir, M. S., and Salim, N. (2014). Weighted voting-based consensus clustering for chemical structure databases. *Journal of computer-aided molecular design*, 28(6):675–684.

[Sander et al., 1998] Sander, J., Ester, M., Kriegel, H.-P., and Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery*, 2(2):169–194.

[Schwarz et al., 1978] Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

[Sharma, 1996] Sharma, S. (1996). *Applied Multivariate Techniques*. John Wiley & Sons, Inc., New York, NY, USA.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905.

[Sneath, 1957] Sneath, P. H. (1957). The application of computers to taxonomy. *Journal of general microbiology*, 17(1):201–226.

[Srivastava et al., 2000] Srivastava, J., Cooley, R., Deshpande, M., and Tan, P.-N. (2000). Web usage mining: Discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, 1(2):12–23.

[Steinley and Brusco, 2011] Steinley, D. and Brusco, M. J. (2011). Choosing the number of clusters in k-means clustering. *Psychological methods*, 16(3):285.

[Strehl and Ghosh, 2003] 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 et al., 2015] Su, P., Shang, C., and Shen, Q. (2015). A hierarchical fuzzy cluster ensemble approach and its application to big data clustering. *Journal of Intelligent & Fuzzy Systems*, 28(6):2409–2421.

[Sugar and James, 2011] Sugar, C. A. and James, G. M. (2011). Finding the number of clusters in a dataset. *Journal of the American Statistical Association*.

[Sukegawa et al., 2013] Sukegawa, N., Yamamoto, Y., and Zhang, L. (2013). Lagrangian relaxation and pegging test for the clique partitioning problem. *Advances in Data Analysis and Classification*, 7(4):363–391.

[Tibshirani and Walther, 2005] Tibshirani, R. and Walther, G. (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528.

[Tomar and Agarwal, 2013] Tomar, D. and Agarwal, S. (2013). A survey on data mining approaches for healthcare. *International Journal of Bio-Science and Bio-Technology*, 5(5):241–266.

[Topchy et al., 2003] Topchy, A., Jain, A. K., and Punch, W. (2003). Combining multiple weak clusterings. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 331–338. IEEE.

[Topchy et al., 2004] Topchy, A., Jain, A. K., and Punch, W. (2004). A mixture model for clustering ensembles. In *Society for Industrial and Applied Mathematics. Proceedings of the SIAM International Conference on Data Mining*, page 379. Society for Industrial and Applied Mathematics.

[Topchy et al., 2005] Topchy, A., Jain, A. K., and Punch, W. (2005). Clustering ensembles: Models of consensus and weak partitions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1866–1881.

[Tumer and Agogino, 2008] Tumer, K. and Agogino, A. K. (2008). Ensemble clustering with voting active clusters. *Pattern Recognition Letters*, 29(14):1947–1953.

[Vega-Pons et al., 2008] Vega-Pons, S., Correa-Morris, J., and Ruiz-Shulcloper, J. (2008). Weighted cluster ensemble using a kernel consensus function. In *Iberoamerican Congress on Pattern Recognition*, pages 195–202. Springer.

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

[Vega-Pons and Ruiz-Shulcloper, 2009] Vega-Pons, S. and Ruiz-Shulcloper, J. (2009). Clustering ensemble method for heterogeneous partitions. In *Iberoamerican Congress on Pattern Recognition*, pages 481–488. Springer.

[Vega-Pons and Ruiz-Shulcloper, 2011] Vega-Pons, S. and Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372.

[Vinh and Epps, 2009] Vinh, N. X. and Epps, J. (2009). A novel approach for automatic number of clusters detection in microarray data based on consensus clustering. In *Bioinformatics and BioEngineering, 2009. BIBE'09. Ninth IEEE International Conference on*, pages 84–91. IEEE.

[Wang et al., 2011a] Wang, H., Shan, H., and Banerjee, A. (2011a). Bayesian cluster ensembles. *Statistical Analysis and Data Mining*, 4(1):54–70.

[Wang, 2010] Wang, J. (2010). Consistent selection of the number of clusters via crossvalidation. *Biometrika*, 97(4):893–904.

[Wang et al., 2011b] Wang, P., Laskey, K. B., Domeniconi, C., and Jordan, M. I. (2011b). Non-parametric bayesian co-clustering ensembles. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 331–342. SIAM.

[Wang and Zhang, 2007] Wang, W. and Zhang, Y. (2007). On fuzzy cluster validity indices. *Fuzzy sets and systems*, 158(19):2095–2117.

[Weingessel et al., 2003] Weingessel, A., Dimitriadou, E., and Hornik, K. (2003). An ensemble method for clustering. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*.

[Weiss, 2005] Weiss, G. M. (2005). Data mining in telecommunications. In *Data Mining and Knowledge Discovery Handbook*, pages 1189–1201. Springer.

[Weiss and Kulikowski, 1991] Weiss, S. and Kulikowski, C. (1991). Computer systems that learn.

[Weng and Poon, 2008] Weng, C. G. and Poon, J. (2008). A new evaluation measure for imbalanced datasets. In *Proceedings of the 7th Australasian Data Mining Conference-Volume 87*, pages 27–32. Australian Computer Society, Inc.

[Wright, 1977] Wright, W. E. (1977). Gravitational clustering. *Pattern recognition*, 9(3):151–166.

[Wu et al., 2013] Wu, J., Liu, H., Xiong, H., and Cao, J. (2013). A theoretic framework of k-means-based consensus clustering. In *IJCAI*.

[Xanthopoulos, 2014] Xanthopoulos, P. (2014). A review on consensus clustering methods. In *Optimization in Science and Engineering*, pages 553–566. Springer.

[Xu and Tian, 2015] Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193.

[Xu and Wunsch, 2005] Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678.

[Yi et al., 2012] Yi, J., Yang, T., Jin, R., Jain, A. K., and Mahdavi, M. (2012). Robust ensemble clustering by matrix completion. In *2012 IEEE 12th International Conference on Data Mining*, pages 1176–1181. IEEE.

[Yu et al., 2014] Yu, H., Liu, Z., and Wang, G. (2014). An automatic method to determine the number of clusters using decision-theoretic rough set. *International Journal of Approximate Reasoning*, 55(1):101–115.

[Zahn, 1971] Zahn, C. T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *Computers, IEEE Transactions on*, 100(1):68–86.

[Žalik, 2010] Žalik, K. R. (2010). Cluster validity index for estimation of fuzzy clusters of different sizes and densities. *Pattern Recognition*, 43(10):3374–3390.

[Zhao et al., 2008] Zhao, Q., Hautamaki, V., and Fränti, P. (2008). Knee point detection in bic for detecting the number of clusters. In *Advanced Concepts for Intelligent Vision Systems*, pages 664–673. Springer.

[Zheng et al., 2010] Zheng, L., Li, T., and Ding, C. (2010). Hierarchical ensemble clustering. In *2010 IEEE International Conference on Data Mining*, pages 1199–1204. IEEE.

[Zhong et al., 2015] Zhong, C., Yue, X., Zhang, Z., and Lei, J. (2015). A clustering ensemble: Two-level-refined co-association matrix with path-based transformation. *Pattern Recognition*, 48(8):2699–2709.