A MULTI-OBJECTIVE APPROACH TO CLUSTER ENSEMBLE SELECTION PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DILAY AKTAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
OPERATIONAL RESEARCH

JULY 2019

Approval of the thesis:

A MULTI-OBJECTIVE APPROACH TO CLUSTER ENSEMBLE SELECTION PROBLEM

submitted by **DILAY AKTAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Operational Research Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Assoc. Prof. Dr. Cem İyigün
Head of Department, **Operational Research** _____

Assist. Prof. Dr. Banu Lokman
Supervisor, **Industrial Engineering, METU** _____

Assoc. Prof. Dr. Tülin İnkaya
Co-supervisor, **Industrial Engineering, Uludag University** _____

Examining Committee Members:

Assist. Prof. Dr. Sakine Batun
Industrial Engineering, METU _____

Assist. Prof. Dr. Banu Lokman
Industrial Engineering, METU _____

Assoc. Prof. Dr. Tülin İnkaya
Industrial Engineering, Uludag University _____

Assist. Prof. Dr. Gülşah Karakaya
Business Administration, METU _____

Assist. Prof. Dr. Fatma Yerlikaya Özkurt
Industrial Engineering, Atılım University _____

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Dilay Aktaş

Signature :

ABSTRACT

A MULTI-OBJECTIVE APPROACH TO CLUSTER ENSEMBLE SELECTION PROBLEM

Aktaş, Dilay
M.S., Department of Operational Research
Supervisor: Assist. Prof. Dr. Banu Lokman
Co-Supervisor: Assoc. Prof. Dr. Tülin İnkaya

July 2019, 71 pages

Clustering is an unsupervised learning method that partitions a data set into groups. The aim is to assign similar points to the same cluster and dissimilar points to different clusters with respect to some notion of similarity. It is applicable to a wide range of areas such as recommender systems, anomaly detection, market research, and customer segmentation. With the advances in the computational power, a diverse set of clustering solutions can be obtained from a dataset using different clustering algorithms, different parameter settings and different features. Clustering ensemble has emerged as a powerful tool for combining the strengths of these multiple clustering solutions and generating a consensus solution. It improves the quality of clustering in terms of accuracy and robustness. In this study, we address the cluster ensemble selection problem, and propose a multi-objective approach to generate a consensus clustering solution. Our proposed algorithm selects a representative subset of clustering solutions, and produces a consensus clustering solution by combining these representatives. Different from the existing approaches, we design the representative selection approach based on three criteria: quality, diversity, and size of the represen-

tative set. Before the representative selection, we apply a preprocessing procedure to analyze the characteristics of the clustering solutions in the library and eliminate the ones that may mislead the consensus function. We test the performance of the proposed approach on the benchmark datasets. The results show that the proposed approach works well, and the resulting consensus solution is better than the clustering solutions in the library.

Keywords: Cluster ensembles, Consensus clustering, Multi-objective clustering

ÖZ

KÜMELEME TOPLULUĞU SEÇİMİ PROBLEMİNE ÇOK AMAÇLI YAKLAŞIM

Aktaş, Dilay
Yüksek Lisans, Yöneylem Araştırması Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Banu Lokman
Ortak Tez Yöneticisi: Doç. Dr. Tülin İnkaya

Temmuz 2019 , 71 sayfa

Kümeleme, verideki gizli örüntüleri ön bilgi olmadan ortaya çıkarmayı hedefleyen gözetimsiz bir öğrenme biçimidir. Kümelemede benzer olan nesneler aynı kümede, benzer olmayan nesneler farklı kümelerde olacak şekilde verinin gruplandırılması amaçlanmaktadır. Öneri sistemleri, dolandırıcılık tespiti, pazar araştırması gibi çeşitli alanlarda kullanılmaktadır. Teknolojideki gelişmelerle birlikte, bir veri setinden farklı kümeleme algoritmaları, farklı parametreler ve farklı öznitelikler kullanılarak çeşitli kümeleme çözümleri elde edilebilmektedir. Kümeleme topluluğu (clustering ensemble), bir veri setinden farklı kümeleme yöntemleri ile elde edilen çözümlerin birleştirilerek fikir birliğine varılan ortak bir çözüm (consensus clustering) oluşturulması için ortaya çıkan güçlü bir araçtır. Böylece, gürbüz (robust) ve doğru (accurate) kümeleme sonuçları elde edilmektedir. Bu çalışmada, kümeleme topluluğu seçimi problemi için çok amaçlı bir yaklaşım önerilerek ortak çözümler üretilmiştir. Önerdiğimiz yaklaşım mevcut kümeleme çözümlerinden temsilciler seçip bu temsilcilerin birleştirilmesiyle bir ortak çözüm üretmektedir. Mevcut çalışmalardan farklı olarak bu çalışmada, bir kümeleme topluluğundan kalite, çeşitlilik ve temsilci sayısına göre

baskın temsilci alt kümeleri seçilmesi amaçlanmaktadır. Alt küme seçim aşamasından önce başlangıç kütüphanesinin özelliklerini incelemek ve ortak çözümü yanıltabilecek ayrık çözümlerin elenmesi hedeflenerek bir ön eleme yöntemi geliştirilmiştir. Önerilen yaklaşımın performansı gerçek sınıf etiketleri bilinen veri setleri üzerinde test edilmiştir. Sonuçlar yaklaşımımızın iyi çalıştığı ve elde edilen ortak çözüm sonuçlarının mevcut çözümler ile kıyaslandığında daha iyi olduğunu göstermektedir.

Anahtar Kelimeler: Denetimsiz öğrenme, Kümeleme topluluğu, Çok amaçlı kümeleme

To my beloved family

ACKNOWLEDGMENTS

First of all, I would like to thank the most to my supervisor Assist. Prof. Dr. Banu Lokman and co-supervisor Assoc. Prof. Dr. Tülin İnkaya. They have not only guided me with their endless patience and support, but also showed me how I would like to be with my students in the future. They were always there for me to show a way out whenever I felt stuck and they have never hesitated to show their support and understanding, both academically and emotionally. I am aware that I was so lucky to experience working with such supervisors.

I would also like to express my sincere thanks to the examining committee, Assist. Prof. Dr. Sakine Batun, Assist. Prof. Dr. Gülşah Karakaya, and Assist. Prof. Dr. Fatma Yerlikaya Özkurt for their valuable feedback that helps us improve our work.

Finally, I would like to thank to my family starting with the ones that are also my dearest friends and colleagues witnessed the most difficult times I had: to Sena Önen Öz and Burak Öz, the couple who took care of me all the time, opened up their house, fed me, gave me pyjamas, and made me stay there with their full support and love; to Melis Özateş Gürbüz from whom I learnt a lot about how to be a colleague and a good friend in this department and especially for being the one who drew me into this research area; to Altan Akdoğan who made me realized that his motivational speeches were actually working in my life, to Cansu Alakuş Çetin and Barış Çetin as I always felt their support and understanding despite the 8000 kms between us; and to Yeti Ziya Gürbüz for being one of the nicest that I have ever met. Thank you all for standing by me, you are the kind of friends that someone ever needs in their life. Last but not least, special thanks to my family for their patience in my pursuit of happiness. It is not possible for me to express my appreciation to my family for always feeling their support in the background.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

ACES	Adaptive Cluster Ensemble Selection
CAS	Cluster and Select
CGM	Consensus Generation Method
DBI	Davies-Bouldin Index
DI	Dunn's Index
DM	Decision Maker
ECS	Efficient Consensus Solution
ES	Efficient Subset
HBGF	Hybrid Bipartite Graph Formulation
HCES	Hierarchical Cluster Ensemble Selection
LGM	Library Generation Method
MOMIP	Multi-objective Mixed Integer Programming
NMI	Normalized Mutual Information Index
PPA	Preprocessing Algorithm
RCA	Representative Clusterings Algorithm
SI	Silhouette Index

CHAPTER 1

INTRODUCTION

Due to the rapid increase in the amount of data generated and collected everyday by various sources, there exists an urgent need for new methods to process data and extract relevant information. This process of extracting knowledge of interest from raw data is called data mining, which is an interdisciplinary field that combines tools and methods from different areas such as computer science, operational research, and statistics. The discovery of data mining dates back to early 90s and now it has been widely used in a variety of application domains such as healthcare, marketing, image processing, recommender systems, and so on. Wide range of applicability brings new techniques and methods to data mining studies since the purpose of collecting data and the characteristics of data are specific to application domain. In addition, as the amount and the pace of data generation increase, the need for computationally efficient and effective methods arises.

Classification and clustering can be considered as one of the main data mining techniques developed for the extraction of knowledge from large datasets. In this study, we are interested in a particular technique, namely clustering. Unlike classification which is categorized as a supervised learning method, clustering is an unsupervised learning method aiming to reveal the true nature of data in the absence of any external knowledge of labels (Jain et al., 1999). It can be summarized as partitioning the whole dataset into subsets such that the points assigned to the same groups are similar, while the ones assigned to different groups are dissimilar with respect to some criteria (Berkhin, 2006). Resulting clustering solution is affected by similarity or dissimilarity measures, desired number of clusters, characteristics of dataset, and parameter settings of the algorithms used. Consequently, there is no single proven

algorithm or technique that performs well for any kind of data and setting (Kuncheva and Hadjitodorov, 2004) suggested by the No Free Lunch Theorem (Wolpert et al., 1995). To use the advantages of different methods' capabilities, combining different solutions into a single solution is studied.

Cluster ensembles emerged as a tool to generate a single consensus clustering solution that reflects the relevant information about the structure of data from a library of clustering solutions. By using a library of solutions, the consensus solution is aimed to be more robust and accurate. A library of solutions can be obtained by using multiple clustering algorithms, different parameter settings, and different representations of data. In the earlier studies of cluster ensembles, it is traditional to use all clustering solutions in the library as the ensemble. Later on, the motivation to use a smaller subset of library that can generate a consensus solution competing with the one obtained by using all of the solutions is introduced and Fern and Lin (2008) show that using all members of library sometimes masks the true nature of data and misleads consensus function in addition to computational effort. As a result, selecting a subset of library as the ensemble is studied.

The aim of cluster ensemble selection is to find a smaller subset of solutions such that the resulting partition performs as well as, or better than the solution obtained by using all of the solutions. Fern and Brodley (2003) suggests that ensemble should be of good quality and diverse to obtain such consensus solutions. Although in the literature different measures are employed, quality mainly stands for the capability of ensemble to reflect the trend in library whereas diversity stands for the ability to attain diverse consensus solutions from ensemble.

In this thesis, we address cluster ensemble selection problem. Given a library of clustering solutions, we develop an algorithm, Representative Clusterings Algorithm (RCA), that finds a subset of solutions representing the library well. Different than the existing approaches, we propose a multi-objective approach that takes quality, diversity, and size of the representative subset. To evaluate the quality of the representative set, RCA assigns a representative clustering solution to each solution in the library and measure the coverage gap by the maximum representation error. The diversity corresponds to the minimum difference in the predictions of representative

solutions. In contrast to the existing literature, we develop a preprocessing algorithm (PPA) and apply to the initial library in order to eliminate the clustering solutions that may mislead the representative selection process. RCA then is applied to generate representative subsets of the preprocessed library. Since RCA aims to minimize the size of the representative set while maximizing the quality and diversity simultaneously, there does not exist a unique solution. RCA is designed to find "efficient subsets (ESs)" for which it is not possible to improve one criterion without sacrificing from another one. RCA iteratively generates ESs by solving a single-objective mathematical model at each iteration. Each ES of clustering solutions is then used to generate "efficient consensus solution (ECS)" by means of a consensus function.

The contribution of this thesis is that efficient subsets are meaningful by themselves as well as efficient consensus solutions obtained by utilizing ESs. Selection of the subsets are based on representation rather than consensus performance. We especially propose this method for the problem domains like customer segmentation and recommender systems where different solutions for different groups are desired to represent a population. We combine the efficient subsets to obtain a consensus solution. Different than the existing studies in ensemble clustering, we also address the case where the true number of clusters is not known. Instead of generating all ESs, we generate a compromise efficient subset based on the three criteria utilizing a scalarization method. Our approach is tested on benchmark datasets and proven to work well. We apply our representative selection approach to a real-life case study where initial clustering solutions are customer perceptions/evaluations instead of generated solutions.

The organization of the thesis is as follows. We give background information on cluster ensemble selection problem and summarize existing methods in Chapter 2. In Chapter 3, we give definitions used in the development of the algorithms. We explain the details of Preprocessing Algorithm (PPA), Representative Clusterings Algorithm (RCA), and Consensus Generation Method (CGM). In Chapter 4, we present computational results of experiments on the benchmark datasets. In Chapter 5, we apply our approach to a real life case study of chocolate candy assortment. We summarize our conclusions and future research directions in Chapter 6.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

In cluster ensembles, *objects* are data points that are subject to grouping/clustering/-partitioning. By applying different strategies, a number of clustering solutions for the same dataset is obtained and the set of clustering solutions is called as *library* or *full-ensemble*. Some studies do not differentiate the terms *ensemble* and library as traditionally all clustering solutions in the library are combined. We use term ensemble for the set of clustering solutions to be combined. By applying a *consensus function* to ensemble, the resulting solution is called as *consensus solution*.

Different than the existing approaches, we focus on selecting a subset to represent the library well with respect to the certain criteria. The aim is to generate better consensus solutions utilizing these representative clustering solutions. In that sense, our approach is similar to a well-known problem in multi-objective optimization, representing nondominated set with a small subset of solutions.

2.1 Cluster Ensemble Problem

Cluster ensembles arise to overcome the restrictions of the traditional methods of clustering. It mainly aims to combine different clustering solutions such that final consensus solution is more robust and accurate than those of the individual members.

Consider a set of N data points $X = \{x_1, x_2, ..., x_N\}$ and $\pi = \{\pi_1, ..., \pi_M\}$ as the set of M clustering solutions obtained by the dataset X. The aim is to combine M clustering solutions and obtain a new clustering solution π^* as the consensus partition of data points in X.

In Figure 2.1 below, the steps of traditional cluster ensemble problem summarized by Boongoen and Iam-On (2018) is presented.

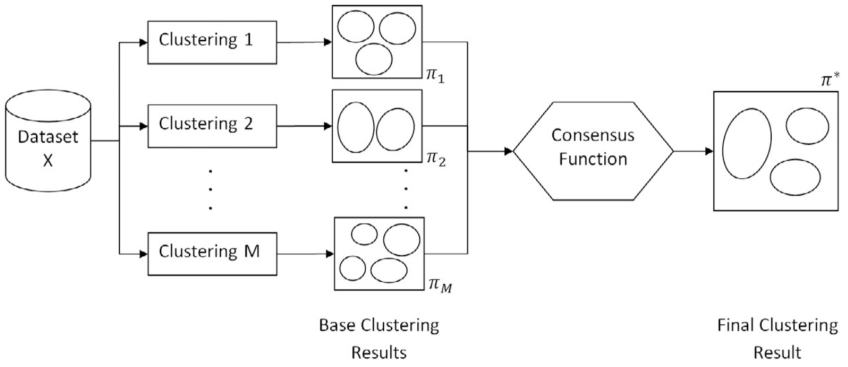


Figure 2.1: Cluster Ensemble Problem

Source: Boongoen and Iam-On (2018)

The cluster ensemble problem originally composes of two stages as library and consensus generation. After a library of solutions is generated, all of the solutions is combined by using a consensus function. In the following Sections 2.1.1 and 2.1.2, main strategies employed in library generation and consensus functions are explained.

2.1.1 Library Generation

According to Fern and Brodley (2003), quality and diversity are two important factors in cluster ensemble problem. It is claimed that including diverse solutions in the ensemble improves resulting consensus solution by Kuncheva and Vetrov (2006). To obtain diverse but consistent library, main strategies used in the literature are summarized below.

- Homogeneous library

  The library consists of solutions obtained by a single nondeterministic algorithm with several initialization settings.

- Different number of clusters

  The library consists of solutions obtained by a single algorithm with changing the number of clusters.

- Random subset of features/subset of data points
  The library consists of solutions that are obtained by randomly selected subsets of features. Similar to the use of random subset of features, randomly selected subsets of data can be used without replacement.

- Heterogeneous library
  The library consists of solutions obtained by different clustering algorithms.

- Mixed methods
  Any combination of the above methods can be used to generate a diverse set of solutions.

In the cluster ensembles literature, k-means algorithm is mostly preferred with the aforementioned methods due to its computational efficiency and performance on large datasets as used by Fern and Lin (2008) Azimi and Fern (2009), Alizadeh et al. (2014), Akbari et al. (2015), Pividori et al. (2016), and Yang et al. (2017). In this study, to generate an initial library of solutions, we use k-means algorithm, as well. Algorithm 1 presents the steps of the k-means algorithm.

Algorithm 1: K-means Algorithm

```
Randomly select k initial centers
while termination condition is not met do
    foreach data point do
        Assign data point to the closest cluster center
    end
    foreach cluster do
        Update cluster center as the mean of its members
    end
end
return Clustering Solution
```

K-means is an iterative algorithm that constructs clusters by assigning points to a cluster center and updating centers as the mean of its members. The algorithm takes the desired number of clusters k as input and seeks for the solution that minimizes the total sum of squared distance between each point and its cluster center. The

algorithm stops when a certain level of convergence such as obtaining the same cluster assignments in successive iterations is achieved (MacQueen et al., 1967).

2.1.2 Consensus Functions

The methods used in combining ensembles can be classified under four categories as direct, feature-based, pairwise-similarity based and graph-based approaches according to Boongoen and Iam-On (2018).

- Direct Approaches
  These approaches use a form of voting for the consensus label of each point. As labels are not meaningful by themselves but the objects having the same labels are in clustering problems, clustering solutions which seem different at the first glance can result in the same groupings. The problem of extracting $unique$ groupings out of a solution is called *label correspondence problem*. They first solve the label correspondence problem for the clustering solutions in the ensemble, then do the voting or find a consensus solution that optimizes label correspondence with ensemble members.

- Feature-based Approaches
  These approaches consider each clustering solution's label information as the features of that solution and decides on the final consensus by considering agreement/disagreement between solutions.

- Pairwise Similarity-based Approaches
  These approaches creates a pairwise-similarity matrix of objects using the ensemble members, then apply clustering to the data using pairwise-similarity matrix constructed based on the clustering solutions.

- Graph-based Approaches
  These approaches represent cluster ensemble with graph representations. Then, partitioning the graph gives the consensus solution.

In this thesis, we use a graph-based approach as the consensus function, namely Hybrid Bipartite Graph Formulation (HBGF). Most of the graph-based approaches either

focus on the relation between data points or between clusters. HBGF makes use of both associations in the graph representation and proven to work well. The method requires the number of clusters desired as an input and it initializes cluster centers randomly (Fern and Brodley, 2004).

2.1.3 Cluster Ensemble Selection Problem

In the earlier studies of cluster ensembles, the focus is mainly on how to generate a library consisting of good quality and diverse solutions rather than selecting solutions to be combined. In this section, we summarize the approaches that are focused on selecting a subset of clustering solutions as ensemble.

Hadjitodorov et al. (2006) introduced the problem of ensemble selection by generating multiple ensembles and selecting among the one with moderate diversity. The authors propose different diversity measures based on a validation metric, Adjusted Rand Index. They calculate the median of diversity and select the corresponding ensemble.

Fern and Lin (2008) are the first to introduce the problem of subset selection based on quality and diversity. The authors define quality of an ensemble as the total similarity of ensemble members to the other solutions in library, and diversity is defined as total pairwise-similarity of solutions in the ensemble by using Normalized Mutual Index (NMI) as a measure of similarity. They propose three methods to deal with two criteria, namely Joint Criterion (JC), Cluster and Select (CAS), and Convex Hull (CH). By Joint Criterion, they aggregate the two criteria by solving a weighted sum problem given the size of the ensemble. Secondly, a lexicographic approach, Cluster and Select is proposed which considers both criteria in a prioritized manner. For a given size of ensemble, clustering solutions are first clustered according to their similarity values and one solution from each cluster is selected for ensemble based on quality values. Grouping of clustering solutions is serving for the diversity while selection of one solution is serving for the quality. Lastly, by Convex hull method, authors represent each pair of solutions by the pair's average quality and pairwise dissimilarity value as diversity. Then, given the size of the ensemble, the pair of solutions with highest quality and diversity are selected.

In adaptive cluster ensemble selection (ACES) studied by Azimi and Fern (2009), the authors propose an adaptive way of subset selection from solutions to be combined by using the characteristics of initial library. The library of solutions is categorized as *stable* and *non-stable*. The authors define four types of subsets by exploiting NMI, as full-ensemble (all solutions), low-diversity ensemble (that half of the solutions more similar to each other), high-diversity ensemble (that half of the solutions less similar to each other), and medium-diversity ensemble (medium half of the solutions from low and high diversity subsets). They suggest that if library is stable, this means dataset is also stable and full-ensemble should be selected. On the other hand, if library is non-stable, high-diversity ensemble should be selected. This approach is computationally efficient and works well, however, subsets are not meaningful by themselves. Subsets are determined to obtain a rule-based method for good consensus solutions.

In hierarchical cluster ensemble selection (HCES) proposed by Akbari et al. (2015), the selection of subset and consensus solution is considered simultaneously by quality and diversity of an ensemble. Similar to CAS, quality and diversity are considered in a lexicographic approach. The authors group clustering solutions for each ensemble size and apply a consensus function. Final consensus solution is chosen based on quality measured by similarity to a reference partitioning. HCES is considered as a generalized version of CAS as it considers size of ensemble at each level and decides on the final consensus solution among the consensus solutions obtained by the ensemble of each level.

A recent study of Yang et al. (2017) proposes another criteria for cluster ensemble selection problem as consistency in addition to quality and diversity. To measure consistency, authors define *must-link* and *cannot-link* constraints indicating that a pair of objects should be in the same cluster, or they should be in different clusters. Then, consistency of an ensemble is measured by the average consistency of ensemble members calculated by the fraction of satisfied constraints. They propose a greedy approach to find a good ensemble with the given ensemble size. The approach searches for the ensemble that maximizes a score based on a weighted objective function of quality, diversity, and consistency where quality and consistency is jointly considered by another measure.

Alizadeh et al. (2014) treat the problem in a different manner. They propose a method that selects base clusters instead of clustering solutions in the ensemble based on stability of partitionings and simplicity of a dataset. In other words, instead of selecting clustering solutions to be combined, authors select clusters itself to be combined. They propose a consensus function for their approach, as well.

In order to compare and evaluate clustering solutions, there are some measures proposed in the literature. In the following Section 2.1.4, some of well-known metrics in evaluating clustering solutions are given.

2.1.4 Evaluating Clustering Solutions

The measures proposed in the literature to evaluate clustering solutions are classified under two main categories as internal validation and external validation metrics (Liu et al., 2010).

External validation metrics evaluate the success of an algorithm based on the accuracy of a clustering solution with respect to true class labels or a reference partition. On the other hand, internal validation measures only use information provided by data and they are also useful in estimating true number of clusters. The metrics we use in this study are explained below.

- **Internal Validation Metrics**

  - Davies-Bouldin index (DBI)

  Let a set of N data points be denoted by $X = \{x_1, x_2, ..., x_N\}$. A clustering solution π_i has k_{π_i} clusters. In each cluster c_i, there exists n_{c_i} objects and the cluster center is denoted by μ_{c_i}. Then, Davies-Bouldin Index of a clustering solution π_i is calculated as follows.

$$DBI_{\pi_i} = \frac{1}{k_{\pi_i}} \sum_{i=1}^{k_{\pi_i}} \underset{i \neq j}{max} \left\{ \frac{\frac{1}{n_{c_i}} (\sum_{x_i \in c_i} ||x_i - \mu_{c_i}||_2) + \frac{1}{n_{c_j}} (\sum_{x_i \in c_j} ||x_i - \mu_{c_j}||_2)}{||\mu_{c_i} - \mu_{c_j}||_2} \right\} \tag{2.1}$$

  The index measures the pairwise similarity of each cluster in a clustering solution. It assigns cluster similarity for each cluster as the maximum similarity

value with the other clusters. Then, by averaging the cluster similarities, DBI is found. It is desired to be minimized so that the clusters will be dissimilar from each other (Davies and Bouldin, 1979).

- Silhouette index (SI)

Let data point x_i is assigned to cluster c_j. $a(x_i)$ is the average dissimilarity of data point x_i to all other objects in c_j. $d(x_i, c_i)$ is the average dissimilarity of data point i to all data points in cluster c_i and $b(x_i)$ is the minimum $d(x_i, c_i)$ where $c_i \neq c_j$, i.e., cluster c_i is the second best choice for x_i. Then, Silhouette Index of a clustering solution is calculated as follows.

$$SI_{\pi_i} = \frac{\sum_{i=1}^{N} s_{x_i}}{N} \tag{2.2}$$

where

$$s(x_i) = \begin{cases} 1 - a(x_i)/b(x_i) & \text{if } a(x_i) < b(x_i) \\ 0 & \text{if } a(x_i) = b(x_i) \\ b(x_i)/a(x_i) - 1 & \text{if } a(x_i) > b(x_i) \end{cases} \tag{2.3}$$

The index measures the silhouette of each data point by its placement. It is desired to be maximized so that between cluster dissimilarity is high, i.e. clusters are separated while within cluster dissimilarity is low, i.e. clusters are compact (Rousseeuw, 1987).

- Dunn's index (DI)

The index measures the ratio of minimum pairwise dissimilarity between clusters and maximum pairwise dissimilarity within clusters. Then, Dunn's Index of a clustering solution is calculated as follows.

$$DI_{\pi_i} = \frac{\underset{\forall m \neq \forall n}{min} \left\{ \underset{\forall x_i \in c_m, \forall x_j \in c_n}{min} ||x_i - x_j||_2 \right\}}{\underset{\forall m}{max} \underset{\forall x_i, x_j \in c_m}{max} ||x_i - x_j||_2} \tag{2.4}$$

It is desired to be maximized so that the closest points in different clusters are as far as possible while the furthest points in a cluster are as close as possible (Dunn, 1974).

- **External Validation Metrics**

  - Normalized Mutual Information (NMI)

  The index is introduced in the study of Strehl and Ghosh (2002) with the geometric mean of entropy of solutions when they first introduce cluster ensemble problem. It measures the mutual information shared by two clustering solutions considering the number of objects in each cluster and the number of clusters in each solution. Then, Normalized Mutual Index between a pair of clustering solutions π_i and π_j is calculated as follows.

$$NMI_{\pi_i,\pi_j} = \frac{\sum_{i=1}^{k_{\pi_i}} \sum_{j=1}^{k_{\pi_j}} n_{c_i,c_j} \log \left(\frac{N n_{c_i,c_j}}{n_{c_i}^{\pi_i} n_{c_j}^{\pi_j}} \right)}{\sqrt{\left(\sum_{i=1}^{k_{\pi_i}} n_{c_i} \log \frac{n_{c_i}}{N} \right) \left(\sum_{j=1}^{k_{\pi_j}} n_{c_j} \log \frac{n_{c_j}}{N} \right)}} \tag{2.5}$$

  The index takes values in [0, 1]. For identical solutions, NMI is equal to 1. Therefore, it is used to compare a clustering solution with the true clustering solution as a measure of *accuracy*.

CHAPTER 3

DEVELOPMENT OF THE ALGORITHM(S)

Given a library of clustering solutions, the aim is to first generate a representative subset of clustering solutions, then to produce a consensus clustering solution. We first develop a preprocessing algorithm (PPA) that eliminates the clustering solutions in the initial library that may mislead the resulting subset. We then develop Representative Clusterings Algorithm (RCA), that finds a subset of solutions that represent the library well. Since RCA aims to minimize the size of the subset while maximizing the diversity and quality, the problem is multi-objective by its nature and there does not exist a unique subset. Therefore, RCA is designed to generate all efficient subsets (ESs) with respect to the three criteria. While RCA could also be employed to generate a consensus solution, we propose alternative methods in Consensus Generation Method (CGM) to find a consensus solution given a subset of clustering solutions.

3.1 Definitions - Notation

In this section, we first present related background information on multi-objective mixed integer programs (MOMIPs). Then, we give definitions and measures we use in our approach.

A multi-objective mixed integer program (MOMIP) with p-objectives can be modelled as follows:

$$\text{“}Max\text{”}\, z = (z_1(x), ..., z_p(x)),$$

subject to;

$$x \in X$$

where x is a feasible vector in decision space, $z_j(x)$ is the objective function value of j^{th} criterion with respect to x, and $z(x) = (z_1(x), ..., z_p(x))$ is the objective vector corresponding to x.

Definition 1. For any $x_1, x_2 \in X$; if $z_j(x_1) \leq z_j(x_2)$ $j = 1, ..., p$ and $z_j(x_1) < z_j(x_2)$ for at least one objective, then $z(x_2)$ is said to be *dominating* $z(x_1)$.
If there does not exists such an $x_2 \in X$, x_1 is called *efficient solution* and $z(x_1)$ is said to be *nondominated point*.

Definition 2. A point whose components are the best values of each objective is called the *ideal point* of an MOMIP.
The *ideal point* is represented as follows.
$z^I = (z_1^I, ..., z_p^I)$ where $z_j^I = \max_{x \in X} z_j(x)$, $j = 1, ..., p$.

Notation we use to represent cluster ensemble problem is as follows.

Given a library of l clustering solutions, $L = \{\pi_1, ..., \pi_l\}$ each having k_{π_i} number of clusters, we first apply PPA to obtain preprocessed library of p clustering solutions, $P = \{\pi_1, ..., \pi_p\}$. By applying RCA to P, the set of m efficient subsets $\tau = \{E_1, ..., E_m\}$ is found. A representative subset E_i consisting of s clustering solutions is denoted as $E_i = \{\pi_1, ..., \pi_s\}$ where $E_i \subseteq P$ and $P \subseteq L$. Ensemble member π_i represents the solutions in the set R_{π_i}, where $R_{\pi_1} \cup R_{\pi_2} ... \cup R_{\pi_S} = P$. In CGM, we apply a consensus function $\Phi(E_i, k)$ to combine the solutions in efficient subset E_i with the desired number of clusters k, resulting efficient consensus solution is denoted by $\pi_{E_i}^*$ or we apply a special case application of RCA, resulting consensus solution is denoted with π_i^*. We first present how clustering solutions are compared.

Definition 3. The *similarity* between two clustering solutions π_i and π_j is measured by the normalized mutual information shared.

$$sim_{\pi_i, \pi_j} = NMI(\pi_i, \pi_j) \tag{3.1}$$

where 0 refers to completely different solutions and 1 refers to identical solutions.

Definition 4. The *dissimilarity* between two clustering solutions π_i and π_j is measured as follows.

$$dis_{\pi_i, \pi_j} = 1 - NMI(\pi_i, \pi_j) \tag{3.2}$$

We next present the measures based on which PPA considers a solution as an outlier.

Definition 5. A solution's *agreement* with the library is measured by its average pairwise similarity with the rest of the solutions.

$$agree_{\pi_i} = \frac{\sum_{\pi_{j \neq i} \in L} sim_{\pi_i, \pi_j}}{|L - 1|} \tag{3.3}$$

Definition 6. *Mean agreement* in the library is calculated as follows.

$$\overline{agree} = \frac{\sum_{\pi_i \in L} agree_{\pi_i}}{|L|} \tag{3.4}$$

Definition 7. *Z-score* of a solution with respect to its agreement value is calculated as follows.

$$zscore_{\pi_i} = \frac{agree_{\pi_i} - \overline{agree}}{\sigma_{agree}} \tag{3.5}$$

where σ_{agree} is the standard deviation of agreement values.

Definition 8. The solution with *Minimum Z-score* has index i^*.

$$zscore_{\pi_i^*} = \min_{\pi_i \in L} zscore_{\pi_i} \tag{3.6}$$

We next discuss how RCA measures the quality, diversity, and size.

Definition 9. *Representation error* caused by an ensemble member π_i is measured by the solution it represents worst. It is calculated as follows.

$$\alpha_{\pi_i} = \max_{\pi_j \in R_{\pi_i}} dis_{\pi_i, \pi_j} \tag{3.7}$$

Definition 10. *Coverage gap* of an ensemble is measured by the maximum representation error caused by its members. We measure *Quality* of an ensemble by the coverage gap. Minimizing coverage gap is equivalent to maximizing quality of the ensemble to represent the library well.

$$CoverageGap_{E_j} = \max_{\pi_i \in E_j} \alpha_{\pi_i} \tag{3.8}$$

Given a library of solutions, the best quality value is obtained when the size of the ensemble is equal to the cardinality of the library as each solution is represented by

itself and the representation error caused by each ensemble member equals to 0. Thus, resulting coverage gap is equal to the minimum possible value, 0.

Definition 11. *Diversity* of an ensemble is measured by the minimum difference in the predictions of its members. Maximizing diversity is desired to represent the library well.

$$Diversity_{E_j} = \min_{\pi_i, \pi_j \in E_j} dis_{\pi_i, \pi_j} \tag{3.9}$$

Given a library of solutions, the best diversity value is obtained when the size of the ensemble is equal to two as the ensemble is composed of the pair of most dissimilar solutions.

Definition 12. *Size* of an ensemble is measured by the number of solutions selected as representatives. Representing the library with minimum number of solutions is desired.

$$Size_{E_j} = |E_j| \tag{3.10}$$

Given a library of solutions, the best size value is two as diversity is not defined for an ensemble of size one. The worst size value equals to the cardinality of the library where each solution is represented by itself.

Coverage gap, diversity, and size are normalized such that the best values take the value of 0 and the worst values take the value of 1 regardless of the direction of the objective.

Definition 13. *Normalized Coverage Gap* of an ensemble is calculated as follows.

$$CoverageGap^{'}_{E_j} = \frac{CoverageGap_{E_j} - CGap_{best}}{CGap_{worst} - CGap_{best}} \tag{3.11}$$

where

$$CGap_{best} = \min_{E_j \in \tau} CoverageGap_{E_j},\ CGap_{worst} = \max_{E_j \in \tau} CoverageGap_{E_j} \tag{3.12}$$

Coverage gap is desired to be minimized to maximize quality. Thus, it is normalized such that the minimum coverage gap value takes the value of 0 while the maximum takes the value of 1.

Definition 14. *Normalized Diversity* of an ensemble is calculated as follows.

$$Diversity_{E_j}^{'} = 1 - \frac{Diversity_{E_j} - Diversity_{worst}}{Diversity_{best} - Diversity_{worst}} \tag{3.13}$$

where

$$Diversity_{best} = \max_{E_j \in \tau} Diversity_{E_j}, \qquad Diversity_{worst} = \min_{E_j \in \tau} Diversity_{E_j} \tag{3.14}$$

Diversity is desired to be maximized. Thus, it is normalized such that the maximum diversity value takes the value of 0 while the minimum diversity value takes the value of 1.

Definition 15. *Normalized Size* of an ensemble is calculated as follows.

$$Size_{E_j}^{'} = \frac{Size_{E_j} - Size_{best}}{Size_{worst} - Size_{best}} \tag{3.15}$$

where

$$Size_{best} = \min_{E_j \in \tau} Size_{E_j}, \qquad Size_{worst} = \max_{E_j \in \tau} Size_{E_j} \tag{3.16}$$

Size is desired to be minimized. Thus, it is normalized such that the minimum size value of two takes the value of 0 and the worst size value of the cardinality of the library takes the value of 1.

We next present how clustering solutions in an ensemble are evaluated to have an estimate on k. Due to the different range of indices, all indices are normalized such that the lowest value takes the value of 0 while the highest value takes the value of 1.

Definition 16. *Normalized Davies-Bouldin Index* of a clustering solution π_i in ensemble E_i is calculated as follows.

$$DBI_{\pi_i}^{'} = \frac{DBI_{\pi_i} - DBI_{best}}{DBI_{worst} - DBI_{best}} \tag{3.17}$$

where

$$DBI_{best} = \min_{\pi_i \in E_i} DBI_{\pi_i}, \qquad DBI_{worst} = \max_{\pi_i \in E_i} DBI_{\pi_i} \tag{3.18}$$

Definition 17. *Normalized Silhouette Index* of a clustering solution π_i in ensemble E_i is calculated as follows.

$$SI_{\pi_i}^{'} = \frac{SI_{\pi_i} - SI_{worst}}{SI_{best} - SI_{worst}} \tag{3.19}$$

where

$$SI_{best} = \max_{\pi_i \in E_i} SI_{\pi_i}, \qquad SI_{worst} = \min_{\pi_i \in E_i} DBI_{\pi_i} \tag{3.20}$$

Definition 18. *Normalized Dunn's Index* of a clustering solution π_i in ensemble E_i is calculated as follows.

$$DI'_{\pi_i} = \frac{DI_{\pi_i} - DI_{worst}}{DI_{best} - DI_{worst}} \tag{3.21}$$

where

$$DI_{best} = \max_{\pi_i \in E_i} DI_{\pi_i}, \qquad DI_{worst} = \min_{\pi_i \in E_i} DI_{\pi_i} \tag{3.22}$$

Definition 19. *Score* of a clustering solution π_i in a subset E_i based on normalized internal validation indices is calculated as follows.

$$Score_{\pi_i} = SI'_{\pi_i} + DI'_{\pi_i} - DBI'_{\pi_i} \tag{3.23}$$

As SI and DI are desired to be maximized while DBI is desired to be minimized, we sum normalized values of SI and DI and subtract DBI in calculating the score. Then, the solution with the highest score value is desired.

Definition 20. The solution with *Maximum Score* has index i*.

$$Score_{\pi_i^*} = \max_{\pi_i \in E_i} Score_{\pi_i} \tag{3.24}$$

Definition 21. *Accuracy* of a clustering solution π_i with respect to true clustering solution Υ is calculated as follows.

$$Acc_{\pi_i} = NMI(\pi_i, \Upsilon) \tag{3.25}$$

Accuracy values close to 1 are desired so as to perfectly match with the true clustering solution.

Figure 3.1 presents general framework of our approach.

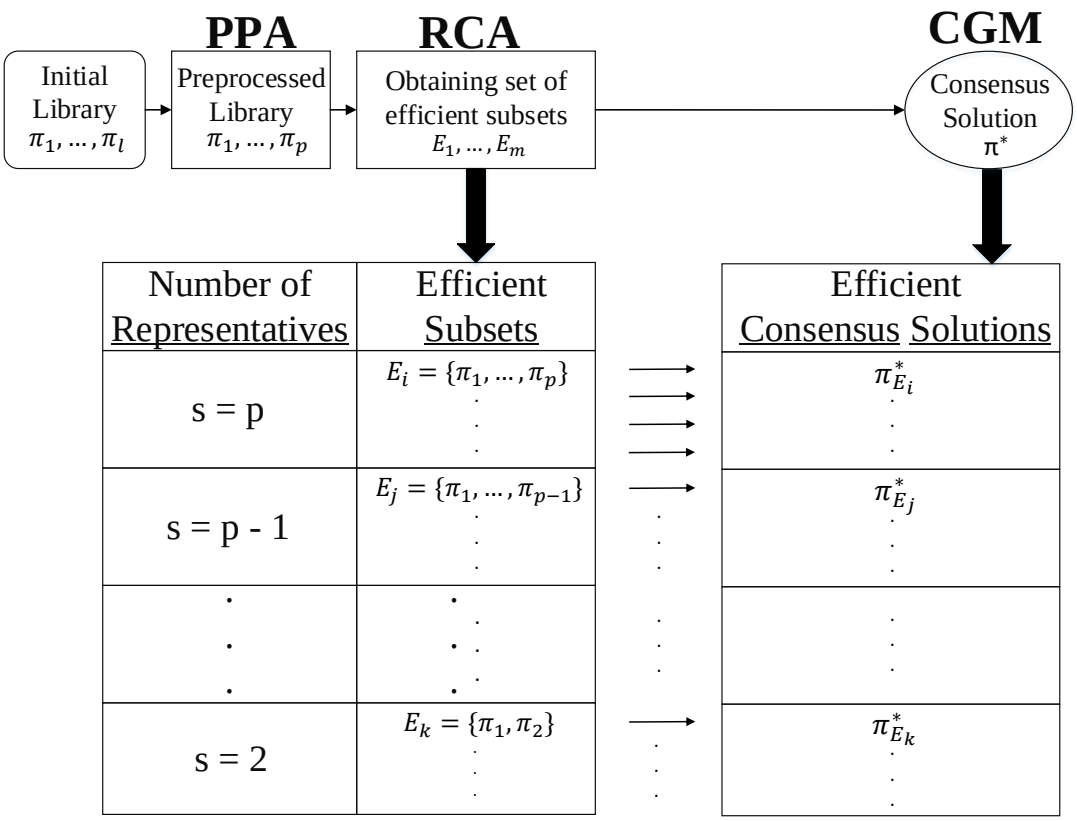


Figure 3.1: Framework of Our Approach

3.2 Preprocessing Algorithm (PPA)

Since our quality metric ensures that all clustering solutions are represented well by the subset generated by RCA, the subset selection process may be misleaded by the extreme solutions. Therefore, we develop a preprocessing algorithm to detect and eliminate such solutions. The process is also useful in reducing computational time and complexity of the problem.

PPA defines each solution by its agreement with the current library. The solutions that are not similar to the trend in the library more than a certain level are considered as outliers. However, we do not define a threshold agreement value. Instead, we make use of z-score values which implicitly sets the threshold value with respect to the library characteristics. As a rule of thumb, data points that lie beyond ± 3 standard deviations from the mean signals anomaly in data collection if data is either normally

distributed or The Central Limit Theorem (CLT) is applicable (Hines et al., 2008). Exploiting the rule of thumb in outlier definition, we only consider solutions that are having agreement values less than three standard deviations of the mean. Solutions that are in agreement with the library more than three standard deviations of the mean are not considered as outliers as they are expected to have high representation capabilities over library. In each iteration, PPA eliminates one solution which has the least agreement with the rest of the library if it is beyond -3 standard deviations and the number of solutions in the library is greater than 30 for CLT to be applicable. In each iteration, agreement values are calculated concerning remaining solutions in the library and procedure is repeated until no other solution is considered as outlier. Algorithm 2 presents the steps of PPA.

Algorithm 2: Preprocessing Algorithm

Set $P = L$
foreach *solution* $\pi_i \in P$ **do**
    Calculate $agree_{\pi_i}$, $zscore_{\pi_i}$
end
Find $zscore_{\pi_i^*} = \min_{\pi_i \in P} zscore_{\pi_i}$
while $zscore_{\pi_i^*} \leq -3$ *and* $|P| > 30$ **do**
    Update $P = P - \{\pi_i^*\}$
    foreach *solution* $\pi_i \in P$ **do**
        Calculate $agree_{\pi_i}$, $zscore_{\pi_i}$
    end
    Find $zscore_{\pi_i^*} = \min_{\pi_i \in P} zscore_{\pi_i}$
end
return P

Note that, when the standard deviation is relatively small, the solutions that are not extremely different than the rest of the library are considered as outliers. In that case, we still have solutions that are similar to the eliminated solutions.

3.3 Representative Clusterings Algorithm (RCA)

Due to the multi-objective nature of the problem, there does not exists a unique subset to represent library well but a set of efficient subsets (ESs). In order to generate ESs in three criteria, we adapt the modified epsilon constraint method (Steuer, 1986). RCA first fixes the size of the representative subset and then generates all nondominated points with respect to the quality and diversity criteria. To do this, RCA solves Model 1 in order to generate a representative set of a fixed size that minimizes the coverage gap of the representative subset while keeping the diversity above a certain level. It then systematically changes the lower bound for the diversity and generate new ESs. After RCA generates all nondominated points for a given size of the representative set, we change the size of the representative subset and repeat the process. The subsets obtained are guaranteed to be nondominated in terms of quality and divesity but not in terms of all three criteria, some subsets can be dominated. RCA finally eliminates the dominated subsets. We summarize the steps of RCA in Algorithm 3.

Table 3.1: Notation for Model 1

Sets	
L	Library of clustering solutions
Parameters	
sim_{ij}	Similarity between solution i and solution j
dis_{ij}	Dissimilarity between solution i and solution j
s	Size of the ensemble
$DiversityLB$	Lower bound on diversity value
Nonnegative Decision Variables	
$CoverageGap$	Maximum representation error caused by representatives
$Diversity$	Minimum difference in the predictions of representatives
Binary Decision Variables	
e_{ij}	1 if solution i is represented by solution j,
	0 otherwise
p_{ij}	1 if solutions i and j are representatives,
	0 otherwise

Minimize *CoverageGap* - $\varepsilon_1 \times$ *Diversity* (3.26)

subject to;

$$\sum_{j \in L} e_{ij} = 1, \qquad \forall i \in L \tag{3.27}$$

$$e_{ij} \leq e_{jj,} \qquad \forall i \in I, \forall j \in J \tag{3.28}$$

$$\sum_{j \in L} e_{jj} = s, \tag{3.29}$$

$$p_{ij} \leq e_{ii}, \qquad \forall i \in L, \forall j \in L \tag{3.30}$$

$$p_{ij} \leq e_{jj}, \qquad \forall i \in L, \forall j \in L \tag{3.31}$$

$$p_{ij} \geq e_{ii} + e_{jj} - 1, \qquad \forall i \in L, \forall j \in L \tag{3.32}$$

$$CoverageGap \geq e_{ij} \times dis_{ij}, \qquad \forall i \in L, \forall j \in L, i \neq j \tag{3.33}$$

$$Diversity \leq p_{ij} \times dis_{ij} + (1 - p_{ij}), \qquad \forall i \in L, \forall j \in L, i \neq j \tag{3.34}$$

$$Diversity \geq DiversityLB, \tag{3.35}$$

$$e_{ij} \in \{0, 1\}, \qquad \forall i \in L, \forall j \in L \tag{3.36}$$

$$p_{ij} \in \{0, 1\}, \qquad \forall i \in L, \forall j \in L \tag{3.37}$$

$$CoverageGap \geq 0, \tag{3.38}$$

$$Diversity \geq 0. \tag{3.39}$$

where the parameters and decision variables are defined in Table 3.1.

Equation 3.27 makes sure that each solution should be represented by exactly one solution and Equation 3.28 links representing relations. Equation 3.29 sets the number of representatives to the predefined size value of s. Equations 3.30, 3.31, and 3.32 serve to linearize p_{ij}. If at least one of the solutions i or j is not a representative, it forces p_{ij} to take the value of zero while when both solutions i and j are representatives, it forces p_{ij} to take the value of one. Equation 3.33 defines coverage gap such that it should be greater than or equal to the maximum representation error. For a pair of solutions without a representation relation, right hand side takes the value of 0. Equation 3.34 defines diversity such that it should be less than or equal to the minimum difference in the predictions of representatives. For the pair of solutions that are not both representatives, the right hand side takes the value of 1. Equation 3.35 is

defined to search objective space. In the first iteration, $DiversityLB$ equals to zero and at each iteration it is updated according to the $Diversity$ found in the previous iteration. Remaining set of constraints makes sure that binary variables take binary values and nonnegative variables take nonnegative values.

Algorithm 3: Representative Clusterings Algorithm

Initialize $s = |P|$
while $s \geq 2$ **do**
    Initialize $DiversityLB = 0$
    while *there exists a feasible subset* **do**
        Solve *Model 1*
        Return subset with size s, $Diversity$, $CoverageGap$
        Set $DiversityLB = Diversity + \varepsilon_2$
    end
    Set $s = s - 1$
end
return all subsets obtained by *Model 1*
Set $\tau \leftarrow$ all subsets obtained
foreach *subset* $i \in \tau$ **do**
    Set $bool = 0$
    Set $j = 1$
    while *check* < *1 and j* < |τ| **do**
        if *subset* j *dominates subset* i **then**
            Set *bool* = 1
            Update $\tau = \tau - \{i\}$
        end
    end
    Set $j = j + 1$
end
return τ

3.4 Consensus Generation Method (CGM)

We present three main approaches in consensus generation method. The first approach generates corresponding efficient consensus solutions (ECSs) by combining the solutions in each ESs generated by RCA. Then, we present decision maker (DM) all distinct ECSs. The second approach selects a compromise ES among all ESs and generates a compromise ECS to present DM. The third approach generates a consensus solution as a special case application of RCA. Applying $RCA_{s=1}$ to P, we obtain the solution that has the least representation error as the consensus solution. To combine clustering solutions in Approaches 1 and 2, we apply a graph-based consensus function, HBGF. The consensus function requires desired number of clusters k for the consensus clustering solution, however; most of the time this information is not available for the unsupervised learning methods. We develop a preliminary estimation method for the case where the true number of clusters is not known or anticipated by DM. We obtain an estimate on the number of clusters from a compromise subset among ESs. General framework of CGM is presented in Figure 3.2.

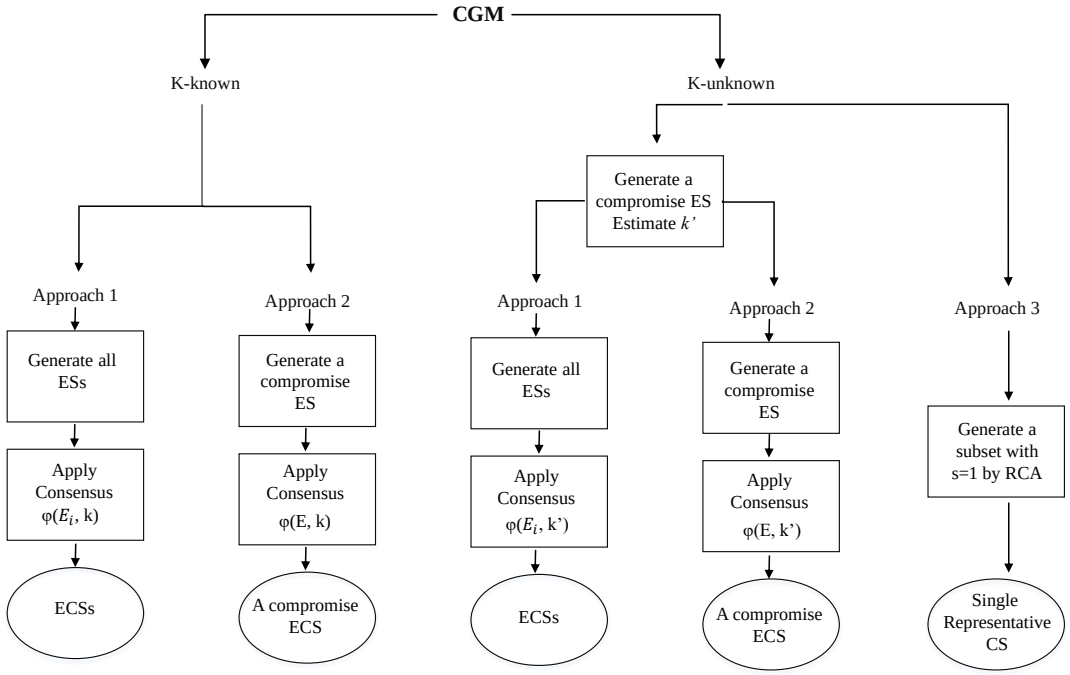


Figure 3.2: General framework of CGM

To select a compromise efficient subset, we use Tchebychev Distance to the Ideal Point based on normalized criteria values. Regardless of the direction of the objective, a subset's normalized criteria values are equivalent to its distance to the ideal point in terms of that criteria due to our normalization. Subset E_j's Tchebychev Distance to the ideal point is defined by its furthest criterion (Cha, 2007).

$$Dist_{E_j}^{Tchebychev} = \max\{CoverageGap'_{E_j}, Diversity'_{E_j}, Size'_{E_j}\} \tag{3.40}$$

We select subset E_j^* that is the closest to the ideal point.

$$Dist_{E_j^*}^{Tchebychev} = \min_{E_j \in \tau} Dist_{E_j}^{Tchebychev} \tag{3.41}$$

We formulate an updated version of Model 1, which we call as Model 2 to generate the closest subset to ideal point without generating ESs. As we use normalized distance metrics, we first generate solutions with the best and the worst criteria values using Model 1. Then we solve Model 2 to find the closest solution in terms of Normalized Tchebychev Distance to the ideal point. We summarize the steps to obtain closest subset in Algorithm 4.

Table 3.2: Additional Notation for Model 2

Parameters	
$CGap_{best}$	Best coverage gap value in the feasible region
$CGap_{worst}$	Worst coverage gap value in the feasible region
D_{best}	Best diversity value in the feasible region
D_{worst}	Worst diversity value in the feasible region
S_{best}	Best size value in the feasible region
S_{worst}	Worst size value in the feasible region
Nonnegative Decision Variables	
$CGap'$	Normalized coverage gap, distance to ideal point in terms of quality
$Diversity'$	Normalized diversity, distance to ideal point in terms of diversity
$Size'$	Normalized size, distance to ideal point in terms of size
$Dist^{Tchebychev}$	Normalized Tchebychev Distance

Minimize $Dist^{Tchebychev}$ (3.42)

subject to;

$$\sum_{j \in L} e_{ij} = 1, \quad \forall i \in L \tag{3.43}$$

$$e_{ij} \leq e_{jj}, \quad \forall i \in I, \forall j \in J \tag{3.44}$$

$$\sum_{j \in L} e_{jj} = Size, \tag{3.45}$$

$$p_{ij} \leq e_{ii}, \quad \forall i \in L, \forall j \in L \tag{3.46}$$

$$p_{ij} \leq e_{jj}, \quad \forall i \in L, \forall j \in L \tag{3.47}$$

$$p_{ij} \geq e_{ii} + e_{jj} - 1, \quad \forall i \in L, \forall j \in L \tag{3.48}$$

$$CoverageGap \geq e_{ij} \times dis_{ij}, \quad \forall i \in L, \forall j \in L, i \neq j \tag{3.49}$$

$$Diversity \leq p_{ij} \times dis_{ij} + (1 - p_{ij}), \quad \forall i \in L, \forall j \in L, i \neq j \tag{3.50}$$

$$CoverageGap^{'} = \frac{CoverageGap - CGap_{best}}{CGap_{worst} - CGap_{best}}, \tag{3.51}$$

$$Diversity^{'} = 1 - \frac{Diversity - D_{best}}{D_{best} - D_{worst}}, \tag{3.52}$$

$$Size^{'} = \frac{Size - S_{best}}{S_{worst} - S_{best}}, \tag{3.53}$$

$$Dist^{Tchebychev} \geq Quality^{'}, \tag{3.54}$$

$$Dist^{Tchebychev} \geq Diversity^{'}, \tag{3.55}$$

$$Dist^{Tchebychev} \geq Size^{'}, \tag{3.56}$$

$$D_{worst} \leq Diversity \leq D_{best}, \tag{3.57}$$

$$CGap_{best} \leq CoverageGap \leq CGap_{worst}, \tag{3.58}$$

$$S_{best} \leq Size \leq S_{worst}, \tag{3.59}$$

$$e_{ij} \in \{0, 1\}, \quad \forall i \in L, \forall j \in L \tag{3.60}$$

$$p_{ij} \in \{0, 1\}, \quad \forall i \in L, \forall j \in L \tag{3.61}$$

$$CoverageGap^{'}, Diversity^{'}, Size^{'} \geq 0, \tag{3.62}$$

where additional parameters and decision variables are defined in Table 3.2.

The first eight set of constraints starting from Equation 3.43 to Equation 3.50 are exactly the same with the ones in *Model 1* except in Equation 3.45 Size is not a pa-

rameter but a decision variable. Equations 3.51, 3.52, and 3.53 are used to normalize the criteria values. Equations 3.54, 3.55, and 3.56 together serve $Dist^{Tchebychev}$ to take the furthest distance value in terms of all criteria. Equations 3.57, 3.58, and 3.59 make sure that we search the same objective space as we do with *Model 1*. In other words, it prevents from obtaining different solutions with *Model 1* due to rounding off. Remaining equations are for sign and set constraints.

Algorithm 4: Generating a Compromise Subset

Set $s = |P|$
Set $DiversityLB = 0$
while *there exists a feasible subset* **do**
    Solve *Model 1*
    Return subset with size s, $Diversity$, $CoverageGap$
    Set $DiversityLB = Diversity + \varepsilon_2$
end
return $(CGap_{best}, D_{worst})$
Set $s = 2$
Set $DiversityLB = 0$
while *there exists a feasible subset* **do**
    Solve *Model-1*
    Return subset with size s, $Diversity$, $CoverageGap$
    Set $DiversityLB = Diversity + \varepsilon_2$
end
return $(D_{best}, CGap_{worst})$
Set $S_{best} = 2$
Set $S_{worst} = |P|$
Solve *Model 2*
return Subset

By evaluating obtained subset's members in terms of internal validation metrics DBI, SI, and DI, we obtain an estimate of k. Kryszczuk and Hurley (2010) suggest that combination of metrics increases the accuracy in detecting correct number of clusters as each index captures different aspects of clustering solutions. DBI is desired to

be minimized while SI and DI are desired to be maximized so that partition is well separated and compact. Given a subset, we normalize the values of internal metrics and calculate an equally-weighted score based on normalized values. We select the highest score representative solution's number of clusters to continue with consensus generation. Steps of this procedure can be found in Algorithm 5.

Algorithm 5: Estimating Number of Clusters

1 Call **Algorithm 4** to generate a compromise subset, E_i

2 Given subset $E_i = \{\pi_1, ..., \pi_s\}$

3 **foreach** *solution* $\pi_i \in E_i$ **do**

4 Calculate DBI_{π_i}, SI_{π_i}, and DI_{π_i}

5 Calculate $Score_{\pi_i}$

6 **end**

7 Find $score_{\pi_i^*} = \max_{\pi_i \in P} score_{\pi_i}$

8 **return** $k_{\pi_i^*}$

In the following section, we present an example to apply PPA, RCA, and CGM.

3.5 Example

To illustrate our approach, we generate an instance of 25 data points with two features. The example we generate includes five classes whose three and two of the classes are not easily distinguished by inspection. Figure 3.3 presents distribution of data points represented by the two features.

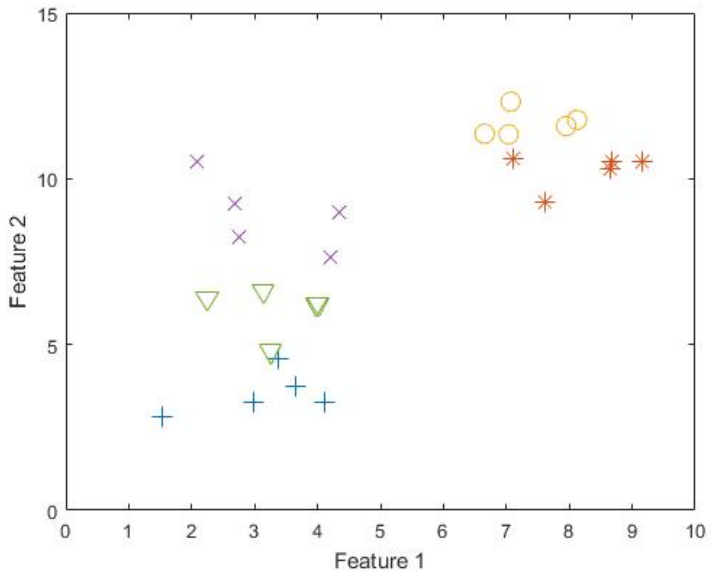


Figure 3.3: Distribution of Data Points

3.5.1 Library Generation Method (LGM)

We generate our initial library using k-means algorithm described in Algorithm 1 for different number of clusters k. We take $k \in [2, m]$ where m is suggested to be $\sqrt{n}$ as a rule of thumb according to Fred and Jain (2002). For each k, we initialize the algorithm r times. Resulting library includes $r \times (\lfloor \sqrt{n} \rfloor - 1)$ clustering solutions. The algorithm might converge to the same solution for the same k. For computational efficiency, we check identical solutions and eliminate them from the library. Algorithm 6 summarizes the steps in library generation.

Algorithm 6: Library Generation Method

Initialize r, n, m
Set $L = \varnothing$
for i = *2: m* **do**
    for j = *1 : r* **do**
        Call **K-means Algorithm**
        Return solution π_{ij}
        Update L = L + $\{\pi_{ij}\}$
    end
end
for i = *1 : |L|* **do**
    for j = *1 : i* **do**
        Calculate $sim(\pi_i, \pi_j)$
        if $sim(\pi_i, \pi_j)$ = *1 and* $i \neq j$ **then**
            Update L = $L - \{\pi_j\}$
        end
    end
end

3.5.2 PPA Application

We report the minimum, mean, and maximum agreement, accuracy, and average DBI, SI, and DI values before and after preprocessing. Table 3.3 presents the results.

Table 3.3: Initial and Preprocessed Library Characteristics

	# of Solutions	Min. Agreement	Avg. Agreement	Max. Agreement	Min. Accuracy	Avg. Accuracy	Max. Accuracy	Avg. DBI	Avg. SI	Avg. DI
Initial	53	0.553	0.737	0.788	0.528	0.754	0.935	0.242	0.609	0.755
Processed	38	0.650	0.745	0.791	0.623	0.756	0.910	0.253	0.623	0.725

According to the results, while initial library consists of 53 solutions, preprocessed library consists of 38 solutions. Minimum agreement increases since PPA eliminates the solutions that are the most dissimilar with the rest of the library on the average. Also, the average accuracy of the library increases although the best solution in the library in terms of accuracy is also considered as one of the outliers by PPA. Some of our inspected performance measures slightly worsen, however; PPA works as intended and it eliminates solutions that may mislead RCA due to representation error within seconds. Figures 3.4 and 3.5 presents distribution of solutions before and after PPA.

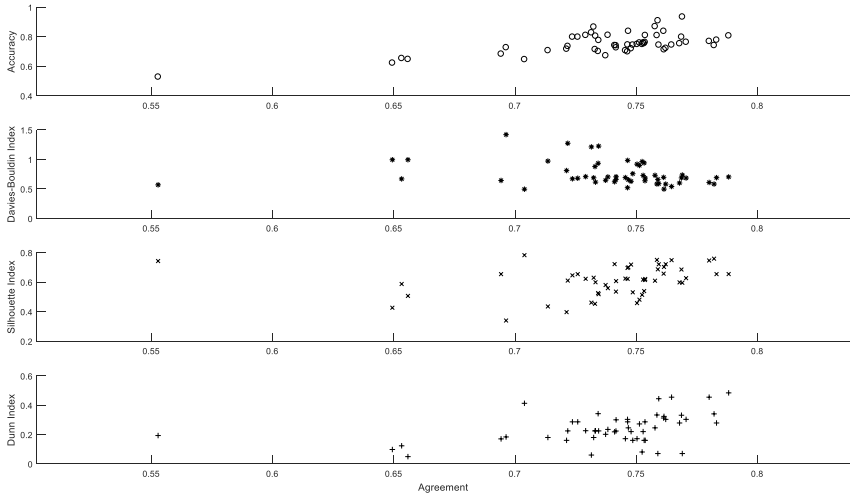


Figure 3.4: Initial Library: Distribution of Solutions

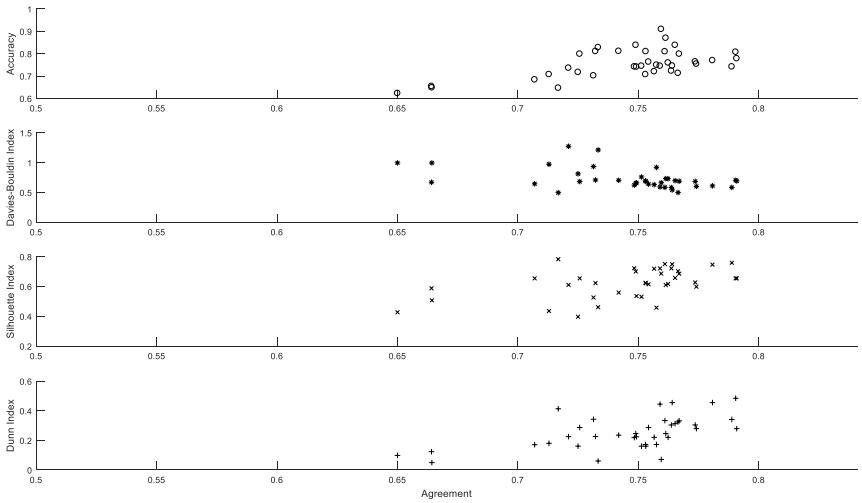


Figure 3.5: Preprocessed Library: Distribution of Solutions

When we analyze the relationship between performance measures and agreement, we observe that good solutions generally have more in common with the rest of the library compared to the poor solutions.

3.5.3 RCA Application

We apply RCA to the preprocessed library of solutions. Figure 3.6 presents ESs with respect to the criteria.

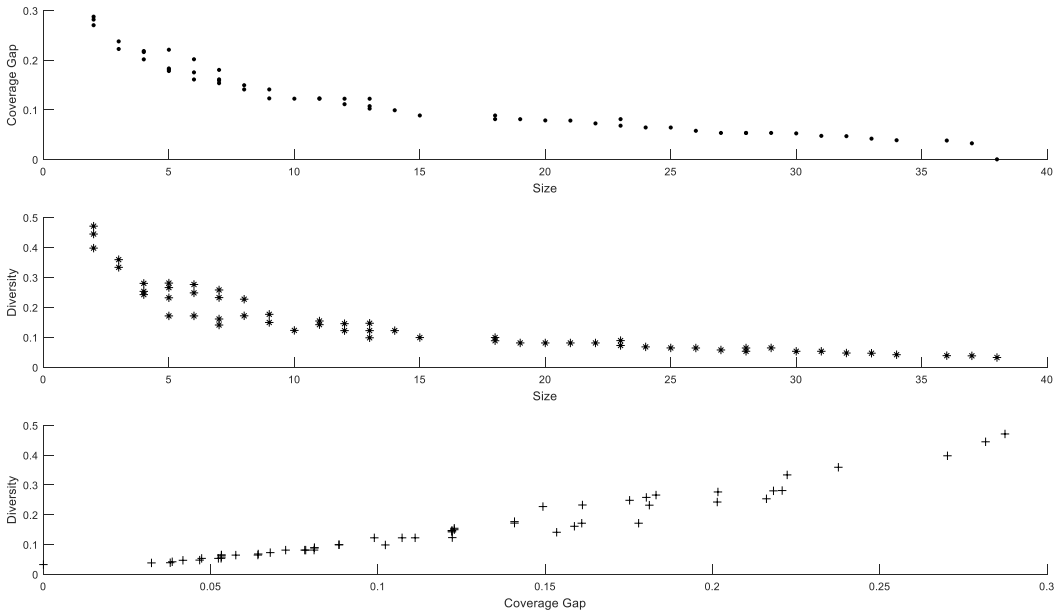


Figure 3.6: Efficient Subsets

As the size increases, coverage gap decreases and quality improves. Considering the extreme case where the size is equal to the number of solutions in the preprocessed library, each solution is represented by itself and coverage gap is equal to zero. As the diversity is measured by the minimum pairwise dissimilarity of the representatives, the best value is obtained when the size is equal to 2. The trade-off between quality and diversity results from their behaviour with size. A subset with the best coverage gap and quality has the worst diversity value and a subset with the best diversity has the worst coverage gap and quality value. For a DM to deal with the clustering solutions, size is considered as the third criterion which is desired to be minimized.

3.5.4 CGM Application

In CGM, we have three main approaches, two of which are applicable where k is known and unknown. Approach 3 is considered for the case where k is unknown. In Approach 1, all ESs are generated and corresponding ECSs are found with the final partitioning of objects into k clusters. In Approach 2, without generating all of the ESs, only a compromise ES is found and corresponding ECS is found with the final partitioning of objects into k clusters. If k is unknown, then it is estimated and the desired number of clusters in the final partitioning of objects is given as k'. In Approach 3, a special case of RCA is applied where the size of the ensemble is selected as 1. A consensus solution is obtained without the need for a consensus function and an estimation on k. Resulting consensus solution is the solution having the minimum representation error in the library. Figures 3.7, 3.8, and 3.9 below summarize the approaches.

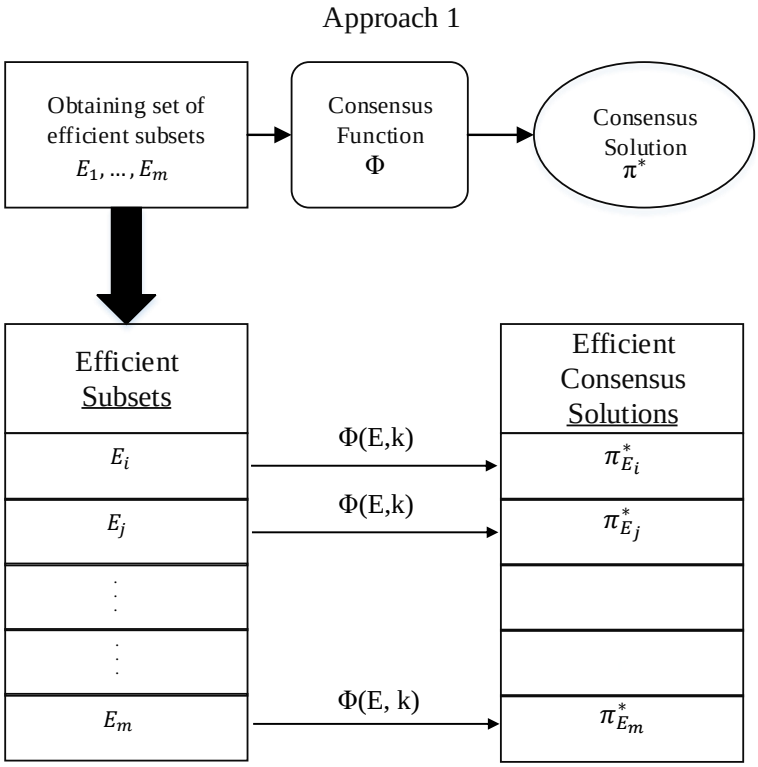


Figure 3.7: CGM: Approach 1 Summary

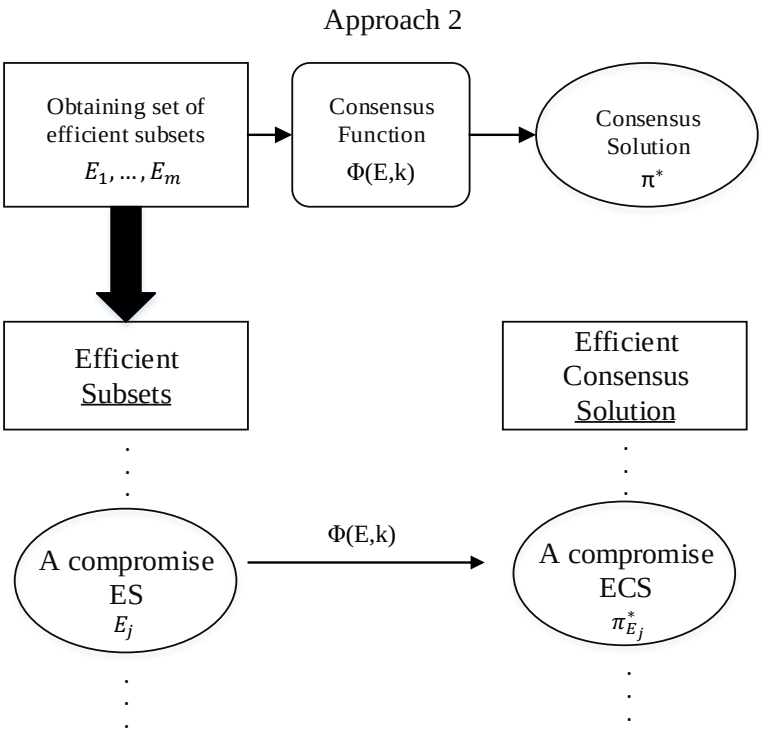


Figure 3.8: CGM: Approach 2 Summary

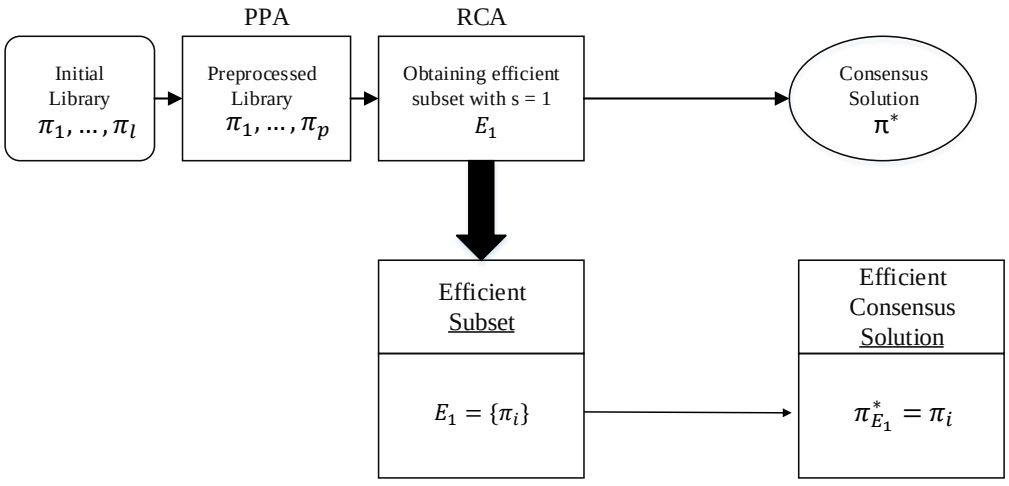


Figure 3.9: CGM: Approach 3 Summary

Case 1: k-known

Approach 1 and Approach 2

We obtain ECSs corresponding to each ES by applying the consensus function HBGF with k is equal to 5. We report the internal and external validation metrics of each ECS in Table 3.4. ES indicated as bold corresponds to the compromise ES.

Table 3.4: CGM : Approach 1 and Approach 2 | k-known

ES ID	Coverage Gap	Size	Diversity	DBI	SI	DI	Accuracy	ES ID	Coverage Gap	Size	Diversity	DBI	SI	DI	Accuracy
1	0.000	38	0.032	0.704	0.655	0.485	0.807	29	0.111	12	0.122	0.727	0.604	0.366	0.870
2	0.032	37	0.038	0.727	0.604	0.366	0.870	30	0.122	12	0.145	0.662	0.536	0.224	0.741
3	0.038	36	0.039	0.727	0.604	0.366	0.870	31	0.122	11	0.142	0.662	0.536	0.224	0.741
4	0.039	34	0.042	0.727	0.604	0.366	0.870	32	0.123	11	0.154	0.727	0.604	0.366	0.870
5	0.042	33	0.047	0.727	0.604	0.366	0.870	33	0.122	10	0.123	0.727	0.604	0.366	0.870
6	0.047	32	0.047	0.727	0.604	0.366	0.870	34	0.123	9	0.149	0.727	0.604	0.366	0.870
7	0.047	31	0.053	0.727	0.604	0.366	0.870	35	0.141	9	0.177	0.731	0.610	0.246	0.870
8	0.052	30	0.053	0.727	0.604	0.366	0.870	36	0.141	8	0.172	0.731	0.610	0.246	0.870
9	0.053	29	0.064	0.727	0.604	0.366	0.870	**37**	**0.149**	**8**	**0.227**	**0.704**	**0.655**	**0.485**	**0.807**
10	0.053	28	0.053	0.727	0.604	0.366	0.870	38	0.153	7	0.141	0.727	0.604	0.366	0.870
11	0.053	28	0.064	0.727	0.604	0.366	0.870	39	0.159	7	0.161	0.871	0.485	0.309	0.714
12	0.053	27	0.058	0.727	0.604	0.366	0.870	40	0.161	7	0.232	0.704	0.655	0.485	0.807
13	0.058	26	0.064	0.727	0.604	0.366	0.870	41	0.180	7	0.258	0.735	0.578	0.220	0.737
14	0.064	25	0.064	0.727	0.604	0.366	0.870	42	0.161	6	0.171	0.704	0.655	0.485	0.807
15	0.064	24	0.068	0.727	0.604	0.366	0.870	43	0.175	6	0.248	0.905	0.482	0.091	0.754
16	0.068	23	0.072	0.727	0.604	0.366	0.870	44	0.202	6	0.276	0.727	0.604	0.366	0.870
17	0.081	23	0.089	0.727	0.604	0.366	0.870	45	0.178	5	0.171	0.731	0.610	0.246	0.870
18	0.072	22	0.081	0.727	0.604	0.366	0.870	46	0.181	5	0.232	0.715	0.586	0.180	0.792
19	0.078	21	0.081	0.727	0.604	0.366	0.870	47	0.183	5	0.266	0.727	0.604	0.366	0.870
20	0.078	20	0.081	0.727	0.604	0.366	0.870	48	0.221	5	0.281	0.793	0.520	0.246	0.786
21	0.081	19	0.081	0.727	0.604	0.366	0.870	49	0.201	4	0.242	0.685	0.621	0.161	0.810
22	0.081	18	0.088	0.727	0.604	0.366	0.870	50	0.216	4	0.253	0.743	0.586	0.225	0.831
23	0.088	18	0.099	0.727	0.604	0.366	0.870	51	0.218	4	0.280	0.731	0.610	0.246	0.870
24	0.088	15	0.099	0.727	0.604	0.366	0.870	52	0.222	3	0.333	0.743	0.586	0.225	0.831
25	0.099	14	0.122	0.704	0.655	0.485	0.807	53	0.238	3	0.359	0.631	0.430	0.294	0.709
26	0.102	13	0.098	0.731	0.610	0.246	0.870	54	0.270	2	0.398	0.704	0.655	0.485	0.807
27	0.107	13	0.122	0.727	0.604	0.366	0.870	55	0.282	2	0.445	2.257	0.200	0.123	0.682
28	0.122	13	0.147	0.727	0.604	0.366	0.870	56	0.287	2	0.471	0.890	0.333	0.123	0.690

With the given ESs, we generate consensus solutions combining 56 ESs and obtain 14 unique ECS solutions. For a given size of ensemble, the solutions with higher accuracy are mostly obtained by the ESs towards better quality with moderate diversity. Considering all ESs, when the diversity and size is the best, solution performance in terms of accuracy is the worst among ECSs. On the other hand, some solutions of high accuracy are obtainable with smaller ensembles. For instance, the ensemble of size 37 and one of the efficient subsets of size 5 result in exactly the same solution.

We compare our results with the most accurate solution in the library and the full-ensemble consensus solution of initial library. Table 3.5 presents evaluation metrics.

Table 3.5: Comparison: Approach 1 and Approach 2 | k-known

	DBI	SI	DI	Accuracy
Full Ensemble	0.704	0.655	0.485	0.807
Library Best	0.737	0.596	0.071	0.935
Average ECSs	0.758	0.585	0.330	0.837
Compromise ECS	0.704	0.655	0.485	0.807

According to internal validation metrics, full-ensemble consensus returns a better solution in terms of the final partitioning's compactness and separation. However, the best solution in the library corresponds to 90% whereas full-ensemble consensus corresponds to 80% of accuracy. Our first approach results in a better consensus solution than full-ensemble solution on the average with 37 out of 56 efficient consensus solutions with smaller ensemble sizes obtained by RCA and generates all efficient consensus solutions in approximately 8 hours. With the compromise efficient subset, we obtain the same solution with the full-ensemble. However, while full ensemble combines 53 solutions, our subset is composed of 8 solutions, which is introduced as the main motivation for cluster ensemble selection. Generating compromise subset and compromise consensus solution without generating all ESs takes approximately 3 hours. Resulting ECS partitions data points into 5 clusters as shown in Figure 3.10

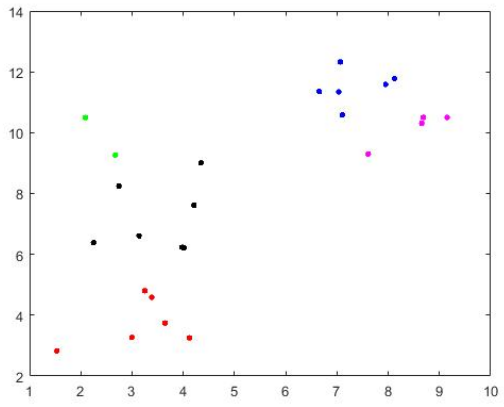


Figure 3.10: K-known - Approach 2: Consensus Solution

Case 2: k-unknown

Approach 1, Approach 2, and Approach 3

To estimate k, we select the ES that is closest to the ideal point in terms of Normalized Tchebychev Distance. We then apply HBGF with estimated k to obtain corresponding ECSs. For the evaluation of k, compromise subset's performance metrics are given in Table 3.6. Ensemble member indicated as bold corresponds to the solution with maximum score. We report the internal and external validation metrics of ECSs generated where k is unknown and estimated in Table 3.7. ES indicated as bold corresponds to the compromise ES.

Table 3.6: Compromise ES: Estimate on k

Representative	# of clusters	DBI	SI	DI	Accuracy	DBI'	SI'	DI'	Score
1	3	0.995	0.427	0.098	0.623	1.000	0.093	0.000	-0.907
2	5	0.812	0.397	0.161	0.717	0.498	0.000	0.255	-0.243
3	5	0.704	0.559	0.235	0.812	0.203	0.503	0.560	0.861
4	4	0.670	0.587	0.123	0.654	0.111	0.592	0.103	0.584
5	**3**	**0.630**	**0.719**	**0.220**	**0.720**	**0.000**	**1.000**	**0.499**	**1.499**
6	5	0.685	0.627	0.304	0.764	0.152	0.714	0.841	1.403
7	5	0.935	0.526	0.343	0.702	0.835	0.401	1.000	0.566
8	5	0.682	0.654	0.287	0.799	0.142	0.799	0.772	1.429

Out of 8 representatives, 5 of them are the solutions with the true number of clusters. However, by calculating the score of each solution with the normalized indices, the maximum score corresponds to the fifth representative. The representative has the best values of indices for DBI and SI, and a moderate value for DI. Selecting $k = 3$ instead of 5 is expected in this case as the clusters are not well separated and creating smaller number of clusters for this dataset is expected to improve the internal measures. Moreover, evaluation of scores is done within seconds; therefore, estimation can be assumed to take as long as the generation of compromise efficient subset.

Table 3.7: CGM : Approach 1 and Approach 2 | k-unknown

ES ID	Coverage Gap	Size	Diversity	DBI	SI	DI	Accuracy	ES ID	Coverage Gap	Size	Diversity	DBI	SI	DI	Accuracy
1	0.000	38	0.032	0.584	0.759	0.341	0.742	29	0.111	12	0.122	0.630	0.719	0.220	0.720
2	0.032	37	0.038	0.584	0.759	0.341	0.742	30	0.122	12	0.145	0.630	0.719	0.220	0.720
3	0.038	36	0.039	0.584	0.759	0.341	0.742	31	0.122	11	0.142	0.584	0.759	0.341	0.742
4	0.039	34	0.042	0.584	0.759	0.341	0.742	32	0.123	11	0.154	0.630	0.719	0.220	0.720
5	0.042	33	0.047	0.584	0.759	0.341	0.742	33	0.122	10	0.123	0.584	0.759	0.341	0.742
6	0.047	32	0.047	0.584	0.759	0.341	0.742	34	0.123	9	0.149	0.630	0.719	0.220	0.720
7	0.047	31	0.053	0.584	0.759	0.341	0.742	35	0.141	9	0.177	0.630	0.719	0.220	0.720
8	0.052	30	0.053	0.630	0.719	0.220	0.720	36	0.141	8	0.172	0.584	0.759	0.341	0.742
9	0.053	29	0.064	0.630	0.719	0.220	0.720	**37**	**0.149**	**8**	**0.227**	**0.584**	**0.759**	**0.341**	**0.742**
10	0.053	28	0.053	0.584	0.759	0.341	0.742	38	0.153	7	0.141	0.630	0.719	0.220	0.720
11	0.053	28	0.064	0.630	0.719	0.220	0.720	39	0.159	7	0.161	0.584	0.759	0.341	0.742
12	0.053	27	0.058	0.630	0.719	0.220	0.720	40	0.161	7	0.232	0.584	0.759	0.341	0.742
13	0.058	26	0.064	0.584	0.759	0.341	0.742	41	0.180	7	0.258	0.630	0.719	0.220	0.720
14	0.064	25	0.064	0.630	0.719	0.220	0.720	42	0.161	6	0.171	0.630	0.719	0.220	0.720
15	0.064	24	0.068	0.630	0.719	0.220	0.720	43	0.175	6	0.248	0.630	0.719	0.220	0.720
16	0.068	23	0.072	0.630	0.719	0.220	0.720	44	0.202	6	0.276	0.584	0.759	0.341	0.742
17	0.081	23	0.089	0.630	0.719	0.220	0.720	45	0.178	5	0.171	0.630	0.719	0.220	0.720
18	0.072	22	0.081	0.630	0.719	0.220	0.720	46	0.181	5	0.232	0.630	0.719	0.220	0.720
19	0.078	21	0.081	0.630	0.719	0.220	0.720	47	0.183	5	0.266	0.585	0.751	0.334	**0.810**
20	0.078	20	0.081	0.630	0.719	0.220	0.720	48	0.221	5	0.281	0.584	0.759	0.341	0.742
21	0.081	19	0.081	0.630	0.719	0.220	0.720	49	0.201	4	0.242	0.584	0.759	0.341	0.742
22	0.081	18	0.088	0.630	0.719	0.220	0.720	50	0.216	4	0.253	0.585	0.751	0.334	**0.810**
23	0.088	18	0.099	0.630	0.719	0.220	0.720	51	0.218	4	0.280	0.630	0.719	0.220	0.720
24	0.088	15	0.099	0.630	0.719	0.220	0.720	52	0.222	3	0.333	0.585	0.751	0.334	**0.810**
25	0.099	14	0.122	0.630	0.719	0.220	0.720	53	0.238	3	0.359	0.585	0.751	0.334	**0.810**
26	0.102	13	0.098	0.630	0.719	0.220	0.720	54	0.270	2	0.398	0.558	0.729	0.255	0.718
27	0.107	13	0.122	0.630	0.719	0.220	0.720	55	0.282	2	0.445	0.584	0.759	0.341	0.742
28	0.122	13	0.147	0.630	0.719	0.220	0.720	56	0.287	2	0.471	0.881	0.450	0.133	0.670

By combining 56 ESs, 6 unique ECSs are obtained. Similar to the results of Approach 1 when k is known, the ensemble with the best diversity resulted in the poorest consensus solution in terms of accuracy. In contrast, we obtain the best solutions from the subsets with lower quality and higher diversity when k is unknown. Our ECSs resulted in no worse than full ensemble consensus for 21 out of 56 efficient subsets. Similar to the case where k is known, with smaller subsets the same or better solutions are achievable. Moreover, four of the ECS resulted in even better consensus solutions compared to the full-ensemble consensus solution when k is known.

We compare our results with the most accurate solution in the library and the full-ensemble consensus solution of initial library. Table 3.8 presents evaluation metrics.

Table 3.8: Comparison: Approach 1, Approach 2, Approach 3 | k-unknown

	DBI	SI	DI	Accuracy
Full Ensemble	0.584	0.759	0.341	0.742
Library Best	0.737	0.596	0.071	0.935
Average ECSs	0.614	0.730	0.268	0.733
Compromise ECS	0.584	0.759	0.341	0.742
Single Representative	0.469	0.782	0.413	0.646

With the compromise efficient subset, we obtain the same solution with the full-ensemble. However, while full ensemble combines 53 solutions, our subset is composed of 8 solutions. Resulting ECS partitions data points into 3 clusters as shown in Figure 3.11

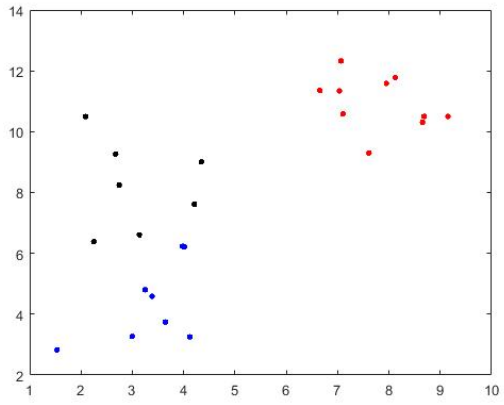


Figure 3.11: K-unknown - Approach 2: Consensus Solution

We apply $RCA_s = 1$ to generate a consensus solution, in other words, to select a single representative among the solutions. The resulting partitioning of objects is given in Figure 3.12.

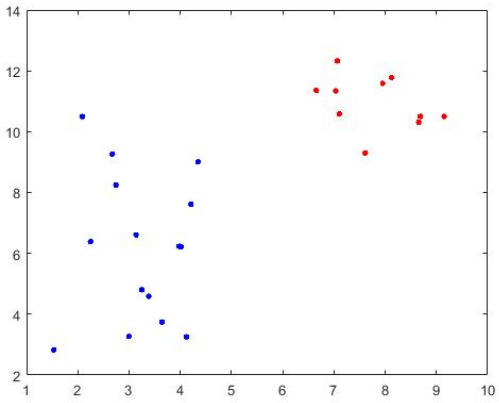


Figure 3.12: K-unknown - Approach 3: Consensus Solution

The approach did not result in estimating the true k value, however; it returns a solution that is representing the library well. Corresponding representation error is 0.35. From this approach, we expect to obtain solutions with smaller number of clusters as increasing the number of clusters decrease agreement.

In this example, we apply PPA, RCA, and CGM to the dataset and we report approximate computational times. PPA and consensus function application are done within seconds and negligible. However, for the approaches requiring an optimization model to be solved as Approach 1 and Approach 2, computational times are dependent to the size of the library and the structure of the similarity and dissimilarity matrices.

CHAPTER 4

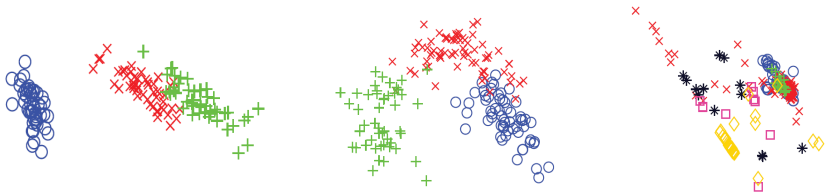
COMPUTATIONAL RESULTS

4.1 Datasets

The approach we propose is tested on three benchmark classification datasets that have different characteristics from UCI Machine Learning Repository (Dua and Graff, 2017). The datasets includes true class labels of objects which can be used as true cluster information. Although labels themselves are not meaningful for clustering problems, objects with the same label should be in the same cluster while objects with different labels should be in different clusters. The approach is applied to Iris, Wine, and Glass datasets and consensus solutions are evaluated with both internal and external metrics. Table 4.1 presents the number of data points, features, and true class labels in the datasets and Figure 4.1 presents the distribution of data points in two dimensional space after Principal Component Analysis is employed (Pividori et al., 2016).

Table 4.1: Properties of Datasets

Dataset	# of Data Points (n)	# of Features (f)	# of True Labels (k)
Iris	150	4	3
Wine	178	13	3
Glass	214	10	6



(a) Iris (b) Wine (c) Glass

Figure 4.1: Iris - Wine - Glass : Distribution of datapoints

Source: Pividori et al. (2016)

Iris is a well known dataset consisting of features for plants with the class information of type of iris. Wine dataset is a result of a chemical analysis on wines derived from different regions with the class information of cultivar. Classification of glass is motivated by criminal investigation and the dataset consists of class information about possible environments such as headlamps and building windows for that glass to come from.

Due to the randomness in both library generation with k-means and consensus application with HBGF, we test our approach by several independent runs. We generate 10 libraries for the same dataset, where each library generation consists of the solutions obtained by 5 random initializations of k-means, and we apply HBGF as the consensus function 10 times each with different random initialization of cluster centers. We report average consensus solution values.

As the benchmark datasets requires a unique solution to compare with true clusters, in this section we first report our compromise efficient consensus solution in Approach 2 and single representative consensus solution in Approach 3 of CGM before and after applying PPA for each library. To evaluate our consensus solutions, we present library characteristics and properties of the best solutions in terms of accuracy in each library for initial and preprocessed libraries of Iris, Wine, and Glass datasets. It should be noted that it is not straightforward to identify those solutions without knowing the true labels. Then, we present a summary table comparing average results for each approach of CGM.

Table 4.2: Iris: Initial Library Characteristics

Initial Library Characteristics										
Library ID	# of Solutions	Min. Agreement	Avg. Agreement	Max. Agreement	Min. Accuracy	Avg. Accuracy	Max. Accuracy	Avg. DBI	Avg. SI	Avg. DI
1	53	0.544	0.733	0.782	0.582	0.653	0.782	0.082	0.514	0.980
2	50	0.539	0.754	0.810	0.596	0.656	0.758	0.087	0.515	0.985
3	52	0.542	0.736	0.786	0.557	0.647	0.758	0.08	0.511	0.999
4	51	0.548	0.747	0.801	0.584	0.659	0.758	0.087	0.519	0.980
5	52	0.535	0.756	0.804	0.588	0.646	0.758	0.086	0.507	1.007
6	51	0.550	0.768	0.821	0.587	0.668	0.758	0.092	0.542	0.985
7	53	0.549	0.746	0.796	0.583	0.655	0.758	0.081	0.526	0.971
8	51	0.555	0.770	0.824	0.602	0.674	0.758	0.086	0.540	0.969
9	53	0.557	0.731	0.789	0.575	0.654	0.758	0.082	0.534	0.973
10	52	0.548	0.748	0.799	0.571	0.655	0.782	0.080	0.518	0.986

Table 4.3: Iris: Preprocessed Library Characteristics

Preprocessed Library Characteristics										
Library ID	# of Solutions	Min. Agreement	Avg. Agreement	Max. Agreement	Min. Accuracy	Avg. Accuracy	Max. Accuracy	Avg. DBI	Avg. SI	Avg. DI
1	30	0.568	0.736	0.775	0.587	0.669	0.782	0.081	0.564	0.923
2	49	0.682	0.763	0.817	0.596	0.656	0.758	0.087	0.509	0.997
3	30	0.578	0.738	0.800	0.572	0.662	0.758	0.078	0.551	0.950
4	30	0.575	0.746	0.784	0.592	0.678	0.758	0.083	0.572	0.926
5	51	0.688	0.765	0.809	0.588	0.645	0.758	0.086	0.500	1.019
6	30	0.565	0.762	0.812	0.587	0.683	0.742	0.089	0.586	0.920
7	30	0.599	0.741	0.792	0.583	0.669	0.758	0.073	0.569	0.925
8	50	0.680	0.779	0.829	0.602	0.674	0.758	0.086	0.534	0.980
9	30	0.590	0.733	0.789	0.588	0.664	0.758	0.077	0.564	0.945
10	35	0.572	0.741	0.796	0.571	0.665	0.782	0.073	0.544	0.963

According to the initial and preprocessed library characteristics presented in Tables 4.2 and 4.3, minimum agreement values increase for all libraries since PPA aims to eliminate the most dissimilar solution if it is above a certain level. However, for some libraries like the sixth, average agreement decreases. This means that some of the eliminated solutions are highly similar to some of the remaining solutions. Due to the same reason, maximum agreement values also decrease for those libraries. When we compare accuracy values, we observe that minimum accuracy values increase and

maximum accuracy values do not decrease in most of the libraries indicating that eliminated solutions are mostly the ones with low accuracy. We report the same metrics for the most accurate solutions in each library. Table 4.4 presents characteristics of the most accurate solutions for initial and preprocessed libraries.

Table 4.4: Iris: Initial and Preprocessed Library Best Solution Characteristics

	Initial Library					Preprocessed Library				
Library ID	# of clusters	DBI	SI	DI	Accuracy	# of clusters	DBI	SI	DI	Accuracy
1	5	0.916	0.619	0.124	0.782	5	0.916	0.619	0.124	0.782
2	3	0.662	0.735	0.099	0.758	3	0.662	0.735	0.099	0.758
3	3	0.662	0.735	0.099	0.758	3	0.662	0.735	0.099	0.758
4	3	0.662	0.735	0.099	0.758	3	0.662	0.735	0.099	0.758
5	3	0.662	0.735	0.099	0.758	3	0.662	0.735	0.099	0.758
6	3	0.662	0.735	0.099	0.758	3	0.666	0.734	0.109	0.742
7	3	0.662	0.735	0.099	0.758	3	0.662	0.735	0.099	0.758
8	3	0.662	0.735	0.099	0.758	3	0.662	0.735	0.099	0.758
9	3	0.662	0.735	0.099	0.758	3	0.662	0.735	0.099	0.758
10	5	0.916	0.619	0.124	0.782	5	0.916	0.619	0.124	0.782
True Clustering	3	0.751	0.657	0.059	1.000	3	0.751	0.657	0.059	1.000

When we compare true clustering solution and best solutions in the libraries, we observe that the best solutions performs better than true clustering solution in terms of all three internal indices except for the first and tenth libraries. In other words, k-means algorithm generates a more compact and separate clustering solution than the true clustering solution of Iris. As accuracy depends only on the true class labels, it is possible not to be able to reach a high level of accuracy when the true clusters are not well separated and compact.

For the case where the true number of clusters is known or anticipated, we apply Approach 2 to generate a compromise efficient consensus solution with k is equal to 3. For a given subset, we apply HBGF 10 times to minimize the effect of random center initialization on our comparison. Table 4.5 and 4.6 presents the average metrics for resulting consensus solutions of our approach and full-ensemble across 10 runs with and without preprocessing, respectively.

Table 4.5: Iris: CGM Results - *k*-known

Iris: k is known - Approach 2 CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	3	0.660	0.734	0.127	0.792	3	0.666	0.734	0.109	0.742
2	3	0.701	0.689	0.102	0.845	3	0.671	0.725	0.130	0.819
3	3	0.660	0.735	0.126	0.778	3	0.659	0.735	0.094	0.751
4	3	0.654	0.735	0.133	0.798	3	0.659	0.734	0.138	0.806
5	3	0.657	0.735	0.115	0.770	3	0.660	0.735	0.116	0.778
6	3	0.659	0.734	0.138	0.806	3	0.660	0.735	0.116	0.778
7	3	0.659	0.734	0.138	0.806	3	0.819	0.647	0.064	0.648
8	3	0.660	0.735	0.126	0.778	3	0.659	0.734	0.138	0.806
9	3	0.654	0.735	0.148	0.798	3	0.856	0.686	0.076	0.649
10	3	0.659	0.734	0.149	0.806	3	0.659	0.734	0.138	0.806
Average	-	0.662	0.730	0.130	0.798	-	0.697	0.720	0.112	0.758

Table 4.6: Iris: Full Ensemble Results - *k*-known

Iris: k is known - Full Ensemble CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	3	0.654	0.735	0.133	0.798	3	0.654	0.735	0.133	0.798
2	3	0.654	0.735	0.136	0.798	3	0.654	0.735	0.133	0.798
3	3	0.659	0.734	0.148	0.806	3	0.659	0.734	0.138	0.806
4	3	0.654	0.735	0.133	0.798	3	0.654	0.735	0.133	0.798
5	3	0.654	0.735	0.136	0.798	3	0.654	0.735	0.133	0.798
6	3	0.659	0.734	0.148	0.806	3	0.654	0.735	0.133	0.798
7	3	0.654	0.735	0.133	0.798	3	0.659	0.734	0.138	0.806
8	3	0.658	0.734	0.139	0.804	3	0.659	0.734	0.138	0.806
9	3	0.654	0.735	0.148	0.798	3	0.654	0.735	0.133	0.798
10	3	0.654	0.735	0.142	0.798	3	0.654	0.735	0.133	0.798
Average	-	0.655	0.734	0.139	0.800	-	0.656	0.734	0.135	0.800

Both full-ensemble consensus solutions and our approach perform better than library best solutions for Iris. PPA worsen the performance of our approach as Iris consists of mostly similar solutions. The solutions we eliminate provide diversity, and working with smaller subsets, we cannot achieve the diversity captured by the full-ensemble.

Table 4.7: Iris: CGM Results - Approach 2 *k*-unknown

Iris: k is unknown - Approach 2 CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	5	0.895	0.563	0.108	0.647	4	0.856	0.613	0.055	0.674
2	2	0.384	0.846	0.339	0.761	5	0.912	0.556	0.062	0.673
3	2	0.384	0.846	0.339	0.761	3	0.659	0.735	0.094	0.751
4	3	0.654	0.735	0.133	0.798	3	0.659	0.734	0.138	0.806
5	2	0.384	0.846	0.339	0.761	4	0.789	0.664	0.137	0.735
6	2	0.384	0.846	0.339	0.761	4	0.850	0.613	0.052	0.713
7	3	0.659	0.734	0.138	0.806	3	0.819	0.647	0.064	0.648
8	2	0.384	0.846	0.339	0.761	4	0.789	0.664	0.137	0.735
9	2	0.384	0.846	0.339	0.761	3	0.856	0.686	0.076	0.649
10	2	0.384	0.846	0.339	0.761	3	0.659	0.734	0.138	0.806
Average	-	0.489	0.795	0.275	0.758	-	0.785	0.665	0.095	0.719

Table 4.8: Iris: CGM Results - Approach 3 *k*-unknown

Iris: k is unknown - Approach 3 CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	5	0.764	0.579	0.055	0.667	5	0.897	0.564	0.068	0.650
2	4	0.764	0.679	0.082	0.704	6	0.916	0.559	0.083	0.657
3	4	0.856	0.613	0.055	0.674	5	0.897	0.564	0.068	0.650
4	4	0.863	0.608	0.055	0.678	3	0.863	0.608	0.055	0.678
5	3	0.662	0.735	0.099	0.758	5	1.008	0.503	0.062	0.644
6	4	0.852	0.614	0.053	0.688	4	0.852	0.614	0.053	0.688
7	4	0.852	0.614	0.053	0.688	4	0.852	0.614	0.053	0.688
8	5	0.742	0.631	0.137	0.707	6	0.987	0.531	0.085	0.696
9	3	0.662	0.735	0.099	0.758	5	0.907	0.559	0.062	0.670
10	4	0.856	0.613	0.055	0.674	4	0.856	0.613	0.055	0.674
Average	-	0.787	0.642	0.074	0.700	-	0.903	0.573	0.064	0.670

Table 4.9: Iris: Full Ensemble Results - *k*-unknown

Iris: k is unknown - Full Ensemble CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	4	0.848	0.610	0.080	0.724	4	0.813	0.641	0.070	0.732
2	2	0.384	0.846	0.339	0.761	5	0.917	0.550	0.062	0.680
3	2	0.384	0.846	0.339	0.761	3	0.659	0.734	0.138	0.806
4	4	0.789	0.664	0.137	0.735	4	0.789	0.664	0.137	0.735
5	2	0.384	0.846	0.339	0.761	4	0.793	0.659	0.137	0.740
6	2	0.384	0.846	0.339	0.761	5	0.813	0.666	0.137	0.698
7	3	0.654	0.735	0.133	0.798	6	0.916	0.554	0.085	0.653
8	2	0.384	0.846	0.339	0.761	6	0.839	0.632	0.058	0.682
9	2	0.384	0.846	0.339	0.761	5	0.917	0.550	0.062	0.680
10	2	0.384	0.846	0.339	0.761	6	0.919	0.554	0.083	0.655
Average	-	0.498	0.793	0.272	0.759	-	0.838	0.620	0.097	0.706

In all PPA applied results of CGM, average consensus accuracy is higher than average library accuracy when Approach 2 is employed, however, we observe that consensus solutions obtained without preprocessing mostly result in more accurate solutions for Iris dataset. Although we eliminate less accurate solutions, their inclusion in ensemble increases resulting partition's accuracy. Comparing Approach 2 with the full-ensemble, we observe that without knowing the true k value, our approach works better than full-ensemble of preprocessed library. Although we applied PPA, this is an example where using a subset instead of all of the solutions work better. Moreover, PPA increases the accuracy of true number of cluster estimation. Out of 10 libraries, PPA applied estimation finds the true k as the selected subset is less concerned with representing extreme solutions. Approach 3 returns moderate solutions in terms of accuracy as a single representative that has minimum representation error is desired. For both cases and approaches, consensus solutions returned are better than average library performance for Iris dataset.

We next discuss the applications on Wine dataset. Tables 4.10 4.11 present initial and preprocessed library characteristics in terms of agreement, accuracy and validation indices.

Table 4.10: Wine: Initial Library Characteristics

Initial Library Characteristics										
Library ID	# of Solutions	Min. Agreement	Avg. Agreement	Max. Agreement	Min. Accuracy	Avg. Accuracy	Max. Accuracy	Avg. DBI	Avg. SI	Avg. DI
1	50	0.508	0.776	0.809	0.347	0.376	0.429	0.029	0.669	0.564
2	50	0.507	0.780	0.821	0.346	0.374	0.443	0.028	0.670	0.587
3	56	0.514	0.757	0.801	0.345	0.376	0.443	0.035	0.688	0.549
4	53	0.498	0.763	0.811	0.339	0.372	0.443	0.030	0.676	0.571
5	55	0.511	0.757	0.799	0.340	0.376	0.443	0.033	0.682	0.570
6	51	0.503	0.778	0.833	0.340	0.372	0.443	0.022	0.662	0.593
7	54	0.517	0.757	0.806	0.339	0.378	0.443	0.034	0.681	0.560
8	48	0.495	0.780	0.825	0.343	0.371	0.429	0.027	0.671	0.579
9	54	0.506	0.768	0.811	0.345	0.374	0.443	0.033	0.686	0.556
10	52	0.504	0.772	0.833	0.343	0.373	0.443	0.024	0.673	0.591

Table 4.11: Wine: Preprocessed Library Characteristics

Preprocessed Library Characteristics										
Library ID	# of Solutions	Min. Agreement	Avg. Agreement	Max. Agreement	Min. Accuracy	Avg. Accuracy	Max. Accuracy	Avg. DBI	Avg. SI	Avg. DI
1	31	0.682	0.773	0.811	0.364	0.383	0.429	0.028	0.679	0.550
2	30	0.515	0.775	0.819	0.350	0.383	0.443	0.027	0.683	0.565
3	30	0.535	0.746	0.791	0.360	0.386	0.429	0.031	0.707	0.529
4	30	0.512	0.745	0.784	0.354	0.382	0.443	0.030	0.693	0.548
5	30	0.527	0.747	0.804	0.363	0.388	0.429	0.034	0.702	0.541
6	30	0.513	0.767	0.817	0.358	0.383	0.443	0.021	0.677	0.572
7	30	0.529	0.750	0.801	0.356	0.390	0.443	0.035	0.699	0.546
8	30	0.511	0.764	0.807	0.361	0.379	0.423	0.026	0.686	0.553
9	30	0.523	0.761	0.809	0.363	0.387	0.429	0.031	0.699	0.538
10	30	0.520	0.748	0.795	0.361	0.383	0.443	0.026	0.693	0.556

Similar to Iris dataset, with PPA, minimum agreement values increase for all libraries and for some libraries like the second, average agreement decreases indicating that some of the eliminated solutions are highly similar to some of the remaining solutions. Due to the same reason, maximum agreement values also decrease for those libraries like the second. This is observed more commonly for the libraries of Wine dataset compared to those of Iris indicating that initial library consists of more diverse set of solutions. When we compare accuracy values, we observe that average accuracy

values increase for all libraries. We report the same metrics for the most accurate solutions in each library. Table 4.12 presents characteristics of the most accurate solutions for initial and preprocessed libraries.

Table 4.12: Wine: Initial and Preprocessed Library Best Solution Characteristics

	Initial Library					Preprocessed Library				
Library ID	# of clusters	DBI	SI	DI	Accuracy	# of clusters	DBI	SI	DI	Accuracy
1	3	0.534	0.732	0.016	0.429	3	0.534	0.732	0.016	0.429
2	2	0.479	0.821	0.023	0.443	2	0.479	0.821	0.023	0.443
3	2	0.479	0.821	0.023	0.443	3	0.534	0.732	0.016	0.429
4	2	0.479	0.821	0.023	0.443	2	0.479	0.821	0.023	0.443
5	2	0.479	0.821	0.023	0.443	3	0.534	0.732	0.016	0.429
6	2	0.479	0.821	0.023	0.443	2	0.479	0.821	0.023	0.443
7	2	0.479	0.821	0.023	0.443	2	0.479	0.821	0.023	0.443
8	3	0.534	0.732	0.016	0.429	2	0.482	0.819	0.015	0.423
9	2	0.479	0.821	0.023	0.443	3	0.534	0.732	0.016	0.429
10	2	0.479	0.821	0.023	0.443	2	0.479	0.821	0.023	0.443
True Clustering	3	1.516	0.250	0.005	1.000	3	1.516	0.250	0.005	1.000

When we compare the best solutions and the true cluster solution, we observe that true labels correspond to poor clustering solutions in terms of compactness and separation. Similar to the results of Iris, k-means algorithm generates more compact and separate clusters of objects and due to the characteristics of the dataset, those solutions do not match with true labels. This is more drastic in Wine dataset as the highest accuracy values are below 50 percent whereas with Iris about 80 percent accuracy could be reached. Tables 4.13, 4.15, 4.16 present results of CGM and Tables 4.14 and 4.17 present full-ensemble consensus solutions when k is known and unknown, respectively.

Table 4.13: Wine: CGM Results - *k*-known

Wine: k is known - Approach 2 CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	3	0.526	0.732	0.054	0.397	3	0.533	0.729	0.024	0.430
2	3	0.533	0.723	0.028	0.430	3	0.533	0.720	0.035	0.432
3	3	0.557	0.641	0.022	0.371	3	0.545	0.729	0.016	0.426
4	3	0.555	0.692	0.021	0.400	3	0.551	0.701	0.017	0.408
5	3	0.554	0.699	0.021	0.402	3	0.534	0.732	0.025	0.429
6	3	0.550	0.663	0.019	0.386	3	0.554	0.650	0.019	0.377
7	3	0.551	0.716	0.017	0.412	3	0.550	0.728	0.018	0.418
8	3	0.556	0.651	0.020	0.377	3	0.552	0.724	0.032	0.404
9	3	0.540	0.704	0.051	0.384	3	0.566	0.576	0.021	0.338
10	3	0.560	0.622	0.024	0.357	3	0.560	0.622	0.024	0.357
Average	-	0.548	0.684	0.028	0.392	-	0.548	0.691	0.023	0.402

Table 4.14: Wine: Full Ensemble Results - *k*-known

Wine: k is known - Full Ensemble CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	3	0.533	0.709	0.022	0.424	3	0.534	0.732	0.024	0.432
2	3	0.534	0.732	0.036	0.432	3	0.534	0.732	0.034	0.432
3	3	0.534	0.732	0.034	0.432	3	0.558	0.624	0.022	0.363
4	3	0.533	0.709	0.018	0.424	3	0.534	0.732	0.024	0.432
5	3	0.534	0.732	0.029	0.432	3	0.534	0.732	0.024	0.432
6	3	0.558	0.624	0.020	0.363	3	0.573	0.528	0.021	0.317
7	3	0.534	0.732	0.024	0.432	3	0.558	0.620	0.029	0.365
8	3	0.533	0.719	0.025	0.427	3	0.556	0.629	0.022	0.367
9	3	0.534	0.731	0.024	0.431	3	0.534	0.732	0.031	0.432
10	3	0.533	0.729	0.024	0.430	3	0.534	0.731	0.033	0.431
Average	-	0.536	0.715	0.026	0.423	-	0.545	0.679	0.027	0.400

Table 4.15: Wine: CGM Results - Approach 2 *k*-unknown

Wine: k is unknown - Approach 2 CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	2	0.482	0.819	0.015	0.423	3	0.533	0.729	0.024	0.430
2	2	0.484	0.817	0.029	0.439	2	0.479	0.821	0.023	0.443
3	2	0.479	0.821	0.023	0.443	2	0.496	0.806	0.043	0.431
4	2	0.496	0.806	0.043	0.431	2	0.487	0.814	0.033	0.437
5	2	0.482	0.819	0.015	0.423	2	0.482	0.819	0.015	0.423
6	2	0.489	0.813	0.029	0.427	2	0.496	0.806	0.043	0.431
7	2	0.482	0.819	0.015	0.423	2	0.479	0.821	0.023	0.443
8	2	0.482	0.819	0.015	0.423	2	0.496	0.806	0.043	0.431
9	2	0.482	0.819	0.015	0.423	2	0.482	0.819	0.015	0.423
10	2	0.489	0.813	0.029	0.427	2	0.487	0.814	0.033	0.437
Average	-	0.485	0.817	0.023	0.428	-	0.492	0.805	0.030	0.433

Table 4.16: Wine: CGM Results - Approach 3 *k*-unknown

Wine: k is unknown - Approach 3 CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	5	0.483	0.731	0.042	0.369	6	0.529	0.706	0.018	0.400
2	5	0.554	0.673	0.012	0.384	4	0.554	0.673	0.012	0.384
3	5	0.483	0.731	0.042	0.369	5	0.483	0.731	0.042	0.369
4	4	0.549	0.730	0.038	0.407	4	0.549	0.730	0.038	0.407
5	5	0.483	0.731	0.042	0.369	4	0.483	0.731	0.042	0.369
6	4	0.549	0.730	0.038	0.407	4	0.549	0.730	0.038	0.407
7	5	0.483	0.731	0.042	0.369	5	0.483	0.731	0.042	0.369
8	4	0.546	0.727	0.042	0.382	3	0.546	0.727	0.042	0.382
9	4	0.544	0.727	0.025	0.378	4	0.544	0.727	0.025	0.378
10	5	0.483	0.731	0.042	0.369	6	0.584	0.696	0.035	0.366
Average	-	0.516	0.724	0.036	0.380	-	0.531	0.718	0.033	0.383

In contrast to Iris, Approach 2 and PPA together provide better solutions when k is unknown and estimated on the average in terms of all performance measures. PPA improves resulting consensus solutions in all approaches compared to the consensus solutions obtained without preprocessing. As discussed previously, initial libraries of

Wine is more diverse than Iris which causes the elimination of some of the extreme solutions providing moderate diversity and improving resulting consensus solution regardless of the value of k. Moreover, our approaches perform better than full-ensemble of both initial and preprocessed libraries.

Table 4.17: Wine: Full Ensemble Results - *k*-unknown

Wine: k is unknown - Full Ensemble CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	2	0.496	0.806	0.043	0.431	7	0.564	0.686	0.019	0.380
2	2	0.496	0.806	0.043	0.431	2	0.496	0.806	0.043	0.431
3	2	0.496	0.806	0.043	0.431	2	0.496	0.806	0.043	0.431
4	6	0.517	0.698	0.026	0.386	6	0.520	0.706	0.028	0.390
5	2	0.489	0.813	0.029	0.427	5	0.538	0.715	0.028	0.408
6	5	0.567	0.653	0.030	0.406	2	0.496	0.806	0.043	0.431
7	6	0.531	0.710	0.028	0.396	6	0.543	0.704	0.028	0.399
8	2	0.496	0.806	0.043	0.431	2	0.496	0.806	0.043	0.431
9	7	1.329	0.675	0.042	0.369	2	0.488	0.814	0.026	0.426
10	4	0.547	0.717	0.050	0.379	2	0.496	0.806	0.043	0.431
Average	-	0.597	0.749	0.038	0.409	-	0.513	0.765	0.035	0.416

We consider the same aspects of our approach for Glass dataset. Tables 4.18 and 4.19 present the effect of PPA considering initial and preprocessed library characteristics.

Table 4.18: Glass: Initial Library Characteristics

Initial Library Characteristics										
Library ID	# of Solutions	Min. Agreement	Avg. Agreement	Max. Agreement	Min. Accuracy	Avg. Accuracy	Max. Accuracy	Avg. DBI	Avg. SI	Avg. DI
1	64	0.492	0.691	0.740	0.260	0.387	0.463	0.037	0.430	1.082
2	64	0.373	0.678	0.739	0.161	0.380	0.440	0.043	0.427	1.054
3	64	0.504	0.683	0.744	0.251	0.377	0.445	0.034	0.407	1.114
4	68	0.333	0.664	0.734	0.153	0.370	0.448	0.035	0.403	1.132
5	66	0.205	0.656	0.724	0.114	0.380	0.440	0.049	0.447	1.058
6	64	0.348	0.685	0.762	0.161	0.382	0.496	0.039	0.410	1.095
7	62	0.501	0.699	0.754	0.264	0.387	0.444	0.038	0.408	1.066
8	62	0.501	0.700	0.755	0.264	0.392	0.449	0.035	0.417	1.072
9	65	0.494	0.697	0.753	0.260	0.372	0.435	0.03	0.394	1.105
10	65	0.339	0.671	0.741	0.147	0.371	0.444	0.033	0.401	1.115

Table 4.19: Glass: Preprocessed Library Characteristics

Preprocessed Library Characteristics										
Library ID	# of Solutions	Min. Agreement	Avg. Agreement	Max. Agreement	Min. Accuracy	Avg. Accuracy	Max. Accuracy	Avg. DBI	Avg. SI	Avg. DI
1	34	0.548	0.697	0.765	0.260	0.372	0.463	0.041	0.490	1.078
2	30	0.428	0.668	0.738	0.161	0.372	0.440	0.053	0.519	1.023
3	37	0.540	0.672	0.742	0.251	0.359	0.440	0.035	0.444	1.126
4	30	0.346	0.623	0.712	0.153	0.333	0.426	0.041	0.456	1.183
5	30	0.214	0.635	0.738	0.114	0.361	0.440	0.068	0.552	1.019
6	47	0.521	0.690	0.767	0.264	0.381	0.496	0.040	0.435	1.092
7	33	0.539	0.698	0.752	0.264	0.376	0.444	0.042	0.473	1.046
8	30	0.550	0.704	0.766	0.264	0.378	0.447	0.037	0.477	1.069
9	30	0.565	0.695	0.740	0.260	0.342	0.435	0.030	0.430	1.125
10	30	0.474	0.672	0.745	0.147	0.348	0.444	0.035	0.478	1.100

Similar to Wine dataset, initial libraries of Glass consists of more extreme solutions compared to Iris as indicated by the range of minimum and maximum agreement. In contrast to Wine, applying PPA does not eliminate neither poor nor good solutions in terms of accuracy. This means that average accuracy solutions are relatively few and considered as extreme solutions. Therefore, preprocessing results in a decrease in the average accuracy for all libraries. Table 4.20 presents the most accurate solutions' characteristics.

Table 4.20: Glass: Initial and Preprocessed Library Best Solution Characteristics

	Initial Library					Preprocessed Library				
Library ID	# of clusters	DBI	SI	DI	Accuracy	# of clusters	DBI	SI	DI	Accuracy
1	8	0.922	0.609	0.052	0.463	8	0.922	0.609	0.052	0.463
2	4	0.888	0.757	0.163	0.440	4	0.888	0.757	0.163	0.440
3	15	1.125	0.356	0.039	0.445	9	1.041	0.635	0.045	0.440
4	15	1.035	0.352	0.032	0.448	4	0.893	0.757	0.167	0.426
5	4	0.888	0.757	0.163	0.440	4	0.888	0.757	0.163	0.440
6	5	1.023	0.697	0.156	0.496	5	1.023	0.697	0.156	0.496
7	10	0.974	0.551	0.035	0.444	10	0.974	0.551	0.035	0.444
8	11	1.041	0.396	0.028	0.449	8	1.143	0.404	0.025	0.447
9	7	1.156	0.380	0.023	0.435	7	1.156	0.380	0.023	0.435
10	6	0.989	0.595	0.043	0.444	6	0.989	0.595	0.043	0.444
True Clustering	6	3.736	-0.248	0.015	1.000	6	3.736	-0.248	0.015	1.000

When true clustering solution is compared with the best solutions in each library with respect to validity indices, library bests are more or less close to each other while true clustering solution is performing poorly in terms of compactness and separation.

Table 4.21: Glass: CGM Results - *k*-known

Glass: k is known - Approach 2 CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	6	1.003	0.576	0.044	0.430	6	1.032	0.560	0.045	0.395
2	6	0.884	0.560	0.077	0.397	6	1.149	0.407	0.032	0.376
3	6	0.984	0.605	0.053	0.426	6	1.099	0.325	0.019	0.399
4	6	1.198	0.435	0.030	0.375	6	1.102	0.350	0.029	0.384
5	6	1.130	0.552	0.060	0.422	6	1.543	0.584	0.061	0.409
6	6	1.026	0.588	0.028	0.440	6	1.041	0.566	0.026	0.431
7	6	1.085	0.367	0.020	0.364	6	1.092	0.350	0.023	0.366
8	6	0.962	0.609	0.047	0.452	6	1.104	0.355	0.017	0.369
9	6	1.097	0.357	0.018	0.364	6	1.093	0.357	0.023	0.392
10	6	1.148	0.402	0.023	0.373	6	1.157	0.340	0.037	0.347
Average	-	1.052	0.505	0.040	0.404	-	1.141	0.419	0.031	0.387

Table 4.22: Glass: Full Ensemble Results - *k*-known

Glass: k is known - Full Ensemble CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	6	0.963	0.610	0.048	0.444	6	1.126	0.382	0.038	0.353
2	6	1.095	0.379	0.031	0.353	6	1.105	0.360	0.031	0.376
3	6	0.974	0.622	0.048	0.421	6	0.975	0.623	0.048	0.426
4	6	1.104	0.377	0.021	0.345	6	1.135	0.353	0.017	0.368
5	6	1.095	0.380	0.029	0.354	6	0.965	0.626	0.053	0.429
6	6	1.101	0.374	0.055	0.345	6	1.096	0.381	0.018	0.358
7	6	1.106	0.377	0.022	0.359	6	1.101	0.385	0.023	0.356
8	6	0.985	0.598	0.045	0.457	6	1.101	0.352	0.011	0.373
9	6	1.095	0.378	0.024	0.353	6	1.101	0.366	0.024	0.390
10	6	0.982	0.602	0.044	0.439	6	1.097	0.389	0.024	0.376
Average	-	1.050	0.470	0.037	0.387	-	1.080	0.422	0.029	0.381

When the number of true clusters are known, PPA worsen the performance of Approach 2; on the other hand, it improves for full-ensemble. As for Glass, library consists of extreme solutions in terms of accuracy, selected subsets are affected by the lack of moderate solutions to balance the diversity.

Table 4.23: Glass: CGM Results - Approach 2 *k*-unknown

Glass: k is unknown - Approach 2 CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	4	1.203	0.566	0.029	0.396	8	1.121	0.423	0.032	0.382
2	2	1.089	0.692	0.098	0.297	2	1.085	0.685	0.089	0.323
3	9	1.152	0.357	0.028	0.389	5	0.954	0.589	0.037	0.413
4	3	1.186	0.446	0.020	0.264	2	1.034	0.695	0.080	0.270
5	4	0.898	0.753	0.124	0.424	2	1.028	0.690	0.089	0.269
6	2	1.092	0.676	0.090	0.346	6	1.041	0.566	0.026	0.431
7	10	1.089	0.362	0.035	0.403	4	0.988	0.549	0.048	0.365
8	5	0.929	0.606	0.032	0.415	5	0.954	0.609	0.027	0.393
9	7	1.165	0.363	0.022	0.412	7	1.210	0.374	0.023	0.427
10	6	1.148	0.402	0.023	0.373	2	1.071	0.689	0.089	0.328

Table 4.24: Glass: CGM Results - Approach 3 *k*-unknown

Average	-	1.095	0.522	0.050	0.372	-	1.049	0.587	0.054	0.360
Glass: k is unknown - Approach 3 CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	5	0.957	0.600	0.049	0.392	5	0.957	0.600	0.049	0.392
2	4	0.888	0.757	0.163	0.440	4	0.888	0.757	0.163	0.440
3	6	0.967	0.621	0.029	0.428	6	0.967	0.621	0.029	0.428
4	4	1.020	0.567	0.031	0.350	3	1.020	0.567	0.031	0.350
5	4	0.896	0.754	0.127	0.413	4	0.896	0.754	0.127	0.413
6	4	1.023	0.563	0.020	0.349	4	1.023	0.563	0.020	0.349
7	5	0.907	0.593	0.049	0.380	4	1.029	0.549	0.036	0.349
8	5	1.071	0.548	0.036	0.382	7	1.069	0.611	0.029	0.428
9	5	1.236	0.304	0.013	0.284	4	1.050	0.534	0.015	0.318
10	4	1.020	0.567	0.031	0.350	3	1.020	0.567	0.031	0.350
Average	-	0.998	0.587	0.055	0.377	-	0.992	0.612	0.053	0.382

As PPA eliminates moderate solutions in terms of accuracy, it improves the performance of Approach 3.

Table 4.25: Glass: Full Ensemble Results - *k*-unknown

Glass: k is unknown - Full Ensemble CSs										
	Initial Library					Preprocessed Library				
Library ID	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy	# of clusters	Average DBI	Average SI	Average DI	Average Accuracy
1	3	1.008	0.732	0.109	0.434	3	1.166	0.438	0.025	0.268
2	4	1.037	0.577	0.031	0.379	4	1.024	0.573	0.020	0.365
3	4	1.122	0.532	0.025	0.317	3	1.135	0.494	0.025	0.324
4	3	1.213	0.481	0.025	0.279	3	1.188	0.480	0.015	0.285
5	4	0.874	0.747	0.117	0.442	2	1.032	0.687	0.089	0.301
6	2	1.090	0.654	0.091	0.377	4	1.091	0.549	0.043	0.336
7	4	1.042	0.565	0.020	0.355	4	1.023	0.568	0.020	0.360
8	3	1.191	0.454	0.025	0.282	4	1.032	0.567	0.028	0.385
9	2	1.169	0.668	0.089	0.376	2	1.071	0.689	0.089	0.328
10	3	1.213	0.481	0.025	0.279	3	1.116	0.485	0.025	0.308
Average	-	1.096	0.589	0.056	0.352	-	1.088	0.553	0.038	0.326

Considering all the results we obtain for these three different benchmark datasets, we conclude that when initial library does not consist of a certain level of diversity, PPA eliminates different solutions by considering them as outliers and worsen the performance of our algorithm. The subsets cannot achieve the diversity that full-ensemble has. When initial library consists of diverse solutions, PPA improves the performance of our algorithm as it eliminates the extremes without losing the capability of generating different consensus solutions. Furthermore, the resulting solutions from our Approaches when k is unknown, are slightly worse than the solutions that are the best in the initial library, which are not straightforward to distinguish.

Table 4.26: Summary of All Results

	k-known				k-unknown			
	Approach 1				Approach 1			
	Avg. DBI	Avg. SI	Avg. DI	Avg. Acc.	Avg. DBI	Avg. SI	Avg. DI	Avg. Acc.
Iris	0.663	0.735	0.099	0.757	0.663	0.735	0.099	0.757
Wine	0.534	0.732	0.024	0.432	0.525	0.708	0.026	0.398
Glass	1.160	0.436	0.033	0.367	1.024	0.564	0.036	0.387
	Approach 2				Approach 2			
	Avg. DBI	Avg. SI	Avg. DI	Avg. Acc.	Avg. DBI	Avg. SI	Avg. DI	Avg. Acc.
Iris	0.697	0.720	0.112	0.758	0.785	0.665	0.095	0.719
Wine	0.548	0.691	0.023	0.402	0.492	0.805	0.030	0.433
Glass	1.141	0.419	0.031	0.387	1.049	0.587	0.054	0.360
	Full Ensemble				Full Ensemble			
	Avg. DBI	Avg. SI	Avg. DI	Avg. Acc.	Avg. DBI	Avg. SI	Avg. DI	Avg. Acc.
Iris	0.656	0.734	0.135	0.800	0.838	0.620	0.097	0.706
Wine	0.545	0.679	0.027	0.400	0.513	0.765	0.035	0.416
Glass	1.080	0.422	0.029	0.381	1.088	0.553	0.038	0.326
	Library Best				Approach 3			
	Avg. DBI	Avg. SI	Avg. DI	Avg. Acc.	Avg. DBI	Avg. SI	Avg. DI	Avg. Acc.
Iris	0.713	0.712	0.105	0.761	0.903	0.573	0.064	0.670
Wine	0.490	0.804	0.022	0.440	0.531	0.718	0.033	0.383
Glass	1.004	0.545	0.074	0.450	0.992	0.612	0.053	0.382

According to the results in 4.26, when k is known, for Iris dataset, full-ensemble consensus solution gives the most accurate results on the average. Then, it is followed by Approach 2 and Approach 1. For Wine dataset, the most accurate solutions are obtained by Approach 1. Then it is followed by Approach 2 and full-ensemble consensus solution gives the least accurate solutions on the average. For Glass dataset, the most accurate solutions are obtained by Approach 2 and followed by full-ensemble consensus solution and Approach 1. For the case where k is known, we conclude that for these three benchmark dataset, Approach 2 provides either the best or the second best solutions in terms of accuracy on the average. When k is unknown, for Iris dataset, Approach 3 provides the poorest solutions while Approach 1 and Approach 2 provide the best and second best solutions on the average. Similar to Iris, for Wine dataset, Approach 3 provides the poorest results while Approach 2 and full-ensemble consensus solution provides the best and second best solutions on the average. In

contrast, for Glass dataset, Approach 3 provides the second best solutions while Approach 1 gives the best solutions on the average.

The computational time required for each dataset and library changes however we notice that Approach 1 and Approach 2 requires the least time for Wine dataset and it is followed by Iris dataset while Glass dataset requires the most computational time due to the structure of the similarity and dissimilarity matrices.

CHAPTER 5

CASE STUDY

5.1 Problem Definition

We apply our multi-objective cluster ensemble selection approach to a customer segmentation problem. For these kind of problems, customers have their own perceptions of objects which can be used as initial library of clustering solutions. Depending on the perceptions, customers have a natural grouping as well. Our main idea is to find a good representative subsets of customers to identify natural grouping.

In this study, we use a demo version of the chocolate candy assortment example studied by Santi et al. (2016). The authors propose a mathematical model that groups customers and finds a consensus clustering solution of chocolates for each group of individuals simultaneously. They collect data on customer perceptions by an online study consisting of 189 undergraduate students. The students are asked to put the chocolates they think that are similar in someway into the same pile and dissimilar into different piles by using as many piles as they want. For 20 type of chocolates, the number of piles that the customers use ranges from 2 to 12 with the average of 5.73 piles which shows that not everybody has the same perception of the same objects.

Due to their formulation, the authors need to turn the pile information of clusters provided by each individual into a similarity/dissimilarity matrix of chocolates. Then, they group the customers into g number of groups such that the group assignments of individuals and their consensus solution simultaneously minimize total sum of within cluster dissimilarity of objects. In this study, we use the pile information directly as our initial library with the advantage of cluster ensemble's not requiring to access the original features or pairwise similarity of objects and the consensus solutions for each

group are the representative customers' clustering solutions.

For this problem, the objects or data points are the chocolate types subject to the study. Those are Almond Joy, Baby Ruth, Butterfinger, Hershey (Almond), Hershey (Plain), Junior Mints, Kit Kat, M & M (Peanut), M & M (Plain), Mars Bar, Milky Way, Mounds Bar, Nestle's Crunch, Oh Henry!, Payday, Reece's Cups, Snickers, Three musketeers, Twix, and York Mint. Initial Library of Solutions consists of 35 clustering solutions obtained by the online study. Ensemble is the representative subset of customers and chocolate partitioning obtained by combining the perceptions of representative customers is the consensus solution.

An example of survey evaluation where 4 piles are used to partition the chocolates is presented in Figure 5.1.

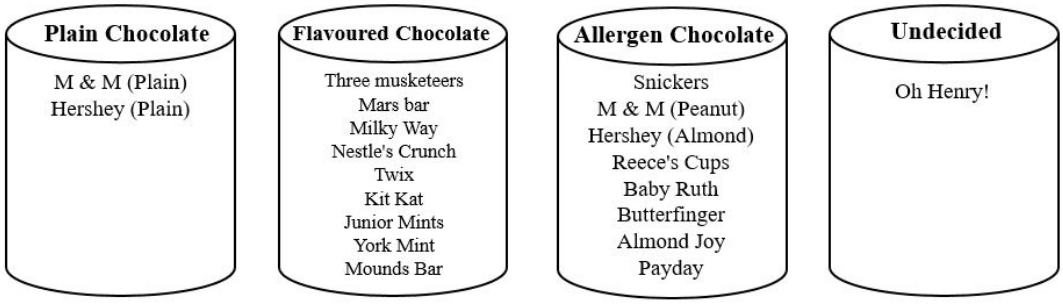


Figure 5.1: Chocolate Candy Assortment: Pile Example

According to the study, the students filling the survey also give a score on the level of confidence about their piles. We choose an initial library of 35 clustering solutions to demonstrate an example of our approach among the surveys having highest level of confidence.

5.2 Results

Among 35 solutions, PPA eliminates only 1 solution. So, we apply RCA to the remaining 34 solutions to identify the natural grouping of customers. At this point, the number of groups that the customers form can be supported by the DM and the subsets corresponding to that size values can be presented. As there is no external

information, we present efficient subsets with additional measures on representation over population. Therefore, for each efficient subset, we also calculate the agreement between a representative and the solutions it represents as follows.

$$agree'_{\pi_i} = \frac{\sum_{\pi_j \in R_{\pi_i}} sim_{\pi_i,\pi_j}}{|R_{\pi_i}|} \tag{5.1}$$

Then, we report minimum, average, and maximum agreement among the representatives of each subset in addition to quality, size, and diversity as presented and the compromise subset is indicated as bold in Table 5.1 and an example efficient subset is given in Figure 5.2.

Table 5.1: Chocolate Candy Assortment Example: ESs

ES ID	Coverage Gap	Size	Diversity	Min. Agr.	Avg. Agr.	Max. Agr.	ES ID	Coverage Gap	Size	Diversity	Min. Agr.	Avg. Agr.	Max. Agr.
1	0.000	34	0.182	1.000	1.000	1.000	25	0.337	12	0.254	0.560	0.841	1.000
2	0.182	33	0.194	0.709	0.991	1.000	26	0.368	12	0.369	0.545	0.850	1.000
3	0.194	32	0.212	0.709	0.988	1.000	27	0.345	11	0.271	0.642	0.862	1.000
4	0.212	31	0.221	0.709	0.982	1.000	28	0.369	11	0.378	0.527	0.852	1.000
5	0.221	29	0.239	0.651	0.971	1.000	29	0.365	10	0.254	0.540	0.860	1.000
6	0.231	28	0.239	0.651	0.968	1.000	30	0.378	10	0.394	0.644	0.861	1.000
7	0.239	27	0.256	0.651	0.963	1.000	31	0.368	9	0.254	0.539	0.838	1.000
8	0.250	26	0.254	0.651	0.957	1.000	32	0.383	9	0.394	0.564	0.864	1.000
9	0.254	25	0.256	0.657	0.955	1.000	33	0.425	9	0.495	0.539	0.777	1.000
10	0.256	24	0.258	0.657	0.948	1.000	34	0.378	8	0.254	0.575	0.823	1.000
11	0.258	23	0.266	0.657	0.941	1.000	35	0.394	8	0.399	0.541	0.819	1.000
12	0.261	22	0.279	0.619	0.938	1.000	36	0.402	7	0.369	0.627	0.817	1.000
13	0.276	21	0.281	0.661	0.941	1.000	37	0.435	7	0.484	0.386	0.749	1.000
14	0.279	20	0.284	0.619	0.934	1.000	38	0.424	6	0.406	0.622	0.770	1.000
15	0.298	19	0.318	0.619	0.913	1.000	39	0.450	5	0.369	0.596	0.718	1.000
16	0.299	18	0.318	0.684	0.927	1.000	40	0.460	5	0.470	0.563	0.707	1.000
17	0.318	17	0.324	0.675	0.907	1.000	41	0.492	5	0.579	0.589	0.639	0.734
18	0.318	16	0.259	0.661	0.880	1.000	42	0.460	4	0.271	0.585	0.718	1.000
19	**0.335**	**16**	**0.361**	**0.596**	**0.900**	**1.000**	43	0.464	4	0.404	0.611	0.733	1.000
20	0.324	15	0.337	0.659	0.853	1.000	44	0.474	4	0.521	0.566	0.629	0.788
21	0.330	14	0.271	0.685	0.887	1.000	45	0.471	3	0.502	0.533	0.557	0.576
22	0.352	14	0.412	0.630	0.834	1.000	46	0.521	3	0.618	0.574	0.622	0.690
23	0.335	13	0.254	0.571	0.844	1.000	47	0.502	2	0.536	0.578	0.581	0.585
24	0.356	13	0.402	0.675	0.872	1.000	48	0.553	2	0.665	0.258	0.399	0.539

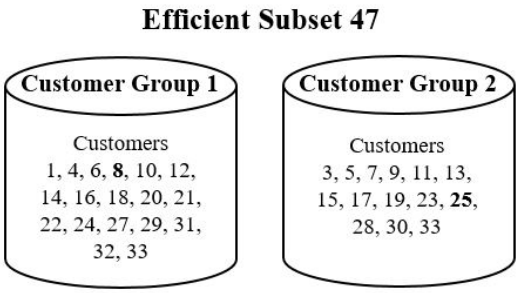


Figure 5.2: Chocolate Candy Assortment: Efficient Subset Example

The first group of customers is represented by customer 8 and the second group is represented by customer 25. The main usage of representative subset selection in this problem context is that for the customers that are in the same group and represented by the same representative, the same marketing strategies are applicable whereas the customers that are in different groups require different strategies. Therefore, a representative subset is useful and practical for the DM.

CHAPTER 6

CONCLUSIONS

Due to the challenges in cluster analysis, different clustering solutions are obtained with different clustering algorithms, different parameter settings, and different representations of data. Thus, there is no proven single algorithm that works well under any kind of data and setting. Cluster ensembles emerge as a tool to combine multiple clustering solutions to obtain a single consensus solution that uses the advantages of different methods. To obtain more accurate and robust consensus solutions, rather than using all of the solutions in the library, cluster ensemble selection is studied. Considering some application areas such as recommender systems and customer segmentation, representing a library of solutions with a small subset is useful and practical for decision makers. In this thesis, we address cluster ensemble selection problem and propose a multi-objective approach to generate efficient subsets. We evaluate the quality of our subsets by the maximum representation error and the diversity of our subsets by the minimum difference in the predictions of our representatives. Different than the existing approaches in the literature, size of the ensemble is also considered and we address the case where the true number of clusters are not known.

Our approach starts with Preprocessing Algorithm (PPA) to eliminate the solutions that may mislead the representative selection and resulting consensus solution. Then, we apply Representative Clusterings Algorithm (RCA) to generate efficient subsets. For the problems when a single clustering solution is desired, we generate efficient consensus solutions by applying a Consensus Generation Method (CGM) to combine efficient subsets.

PPA is developed based on statistical outlier detection. We analyze the library characteristics and eliminate the solutions that are dissimilar than the rest of the library

more than a certain level and hard to represent. As the existence of some outliers might mask the existence of others, we eliminate one solution at a time and repeat the procedure. Results on the benchmark datasets show that PPA is useful in eliminating poor solutions and reducing computational time. It improves the performance of our approach when initial library consists of diverse set of solutions.

RCA is developed to generate efficient subsets considering the three criteria, quality, diversity, and size. The algorithm first fixes the size of the representative subset and generates all nondominated points with respect to the quality and diversity utilizing epsilon constraint method. The algorithm stops after eliminating dominated subsets.

CGM is composed of three approaches. The first approach generates all efficient consensus solutions corresponding to efficient subsets obtained by RCA. Results obtained by the benchmark datasets show that with smaller subsets, consensus solutions as well as, or better than full-ensemble solutions are obtainable. The second approach generates a compromise efficient consensus solution corresponding to a compromise efficient subset. An updated version of RCA is modelled to generate the compromise efficient subset without generating all nondominated points. Results show that with the compromise subset, full-ensemble solutions are achievable. While the first and the second approaches are both applicable to the cases where the true number of clusters is known and unknown, we propose an estimate on the true number of clusters by using a score based evaluation of the solutions in the compromise subset. We continue with the highest scored solution's number of clusters as our estimate. Moreover, as a special case application of RCA, Approach 3 generates a single representative solution as the consensus without the need for the true number of clusters and a consensus function. Results show that the single representative consensus solution is an average solution in terms of performance but representing library well.

As future research, a classification of the datasets can be considered according to the need for preprocessing. PPA can be suggested to be applied for a dataset while it is not employed for another depending on the characteristics of data. Secondly, due to the computational time required by exact methods, RCA can be employed with faster methods for larger and complicated libraries. For the problems where the clustering solutions are not generated like chocolate candy assortment, initial library

of solutions is more diverse and computationally complicated than the libraries of generated solutions. Lastly, other metrics can be employed to generate a compromise efficient subset that leads to the efficient consensus solutions performing better than full-ensemble consensus solutions.

REFERENCES

Akbari, E., Dahlan, H. M., Ibrahim, R., and Alizadeh, H. (2015). Hierarchical cluster ensemble selection. *Engineering Applications of Artificial Intelligence*, 39:146–156.

Alizadeh, H., Minaei-Bidgoli, B., and Parvin, H. (2014). To improve the quality of cluster ensembles by selecting a subset of base clusters. *Journal of Experimental & Theoretical Artificial Intelligence*, 26(1):127–150.

Azimi, J. and Fern, X. (2009). Adaptive cluster ensemble selection. In *Twenty-First International Joint Conference on Artificial Intelligence*.

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

Boongoen, T. and Iam-On, N. (2018). Cluster ensembles: A survey of approaches with recent extensions and applications. *Computer Science Review*, 28:1–25.

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104.

Fern, X. Z. and Brodley, C. E. (2003). Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 186–193.

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.

Fern, X. Z. and Lin, W. (2008). Cluster ensemble selection. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 1(3):128–141.

Fred, A. L. and Jain, A. K. (2002). Data clustering using evidence accumulation. In *Object recognition supported by user interaction for service robots*, volume 4, pages 276–280. IEEE.

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

Hines, W. W., Montgomery, D. C., and Borror, D. M. G. C. M. (2008). *Probability and statistics in engineering*. John Wiley & Sons.

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

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

Kuncheva, L. I. and Hadjitodorov, S. T. (2004). Using diversity in cluster ensembles. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 2, pages 1214–1219. IEEE.

Kuncheva, L. I. and Vetrov, D. P. (2006). Evaluation of stability of k-means cluster ensembles with respect to random initialization. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1798–1808.

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

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.

Pividori, M., Stegmayer, G., and Milone, D. H. (2016). Diversity control for improving the analysis of consensus clustering. *Information Sciences*, 361:120–134.

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.

Santi, É., Aloise, D., and Blanchard, S. J. (2016). A model for clustering data from heterogeneous dissimilarities. *European Journal of Operational Research*, 253(3):659–672.

Steuer, R. E. (1986). *Multiple criteria optimization: theory, computation, and application*, volume 233.

Strehl, A. and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617.

Wolpert, D. H., Macready, W. G., et al. (1995). No free lunch theorems for search. Technical report, Technical Report SFI-TR-95-02-010, Santa Fe Institute.

Yang, F., Li, T., Zhou, Q., and Xiao, H. (2017). Cluster ensemble selection with constraints. *Neurocomputing*, 235:59–70.