NOVEL CLUSTERING AND CLASSIFICATION ALGORITHMS FOR BIG DATA

by

DEBOPRIYA GHOSH

A dissertation submitted to the

Graduate School - Newark

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Management

Written under the direction of

Professors

Michael Katehakis and Javier Cabrera

and approved by

_____

_____

_____

_____

_____

_____

_____

Newark, New Jersey

May, 2020

ABSTRACT OF THE DISSERTATION

Novel Clustering and Classification Algorithms for Big Data

By Debopriya Ghosh

Dissertation Directors:

Professors

Michael Katehakis and Javier Cabrera

This dissertation studies two important problems that arise in the analysis of Big Data: high dimensionality and massive size of pertinent samples.

Unlike traditional datasets where the sample size is moderately large and typically higher than the number of features, Big Datasets are characterized by both massive sample size and high dimensionality. These characteristics render the traditional statistical procedures inappropriate for analyzing Big Data. The massive sample size, which can be in the order of millions or even billions, e.g., in genomics, neuroscience, marketing, and social media, gives rise to intensive computation and reduced scalability. High dimensionality creates spurious correlation and noise accumulation. It also result in incidental endogeneity, a phenomenon in which many unrelated covariates are correlated with residual noises solely by chance. This causes statistical biases and model inconsistencies. In order to address these challenges, we develop three novel algorithms for clustering and classification of Big Data.

In Chapter 2, we present a novel two-way clustering approach by combining model-based and weighted K-means clustering methods. The two-way approach results in

small subgroups of binary features that are of size p or less, so that the possible number of patterns (2^p) is small enough to be efficiently handled by clustering algorithms. This approach can also handle weighted reduced data.

Chapter 3 presents two techniques. First, it derives a weighted probabilistic distance-based clustering technique adjusted for cluster sizes. Second, it derives a probabilistic approximation of capacitated clustering problem, where the cluster sizes are specified as constraints. Both these methods are also capable of handling data that has been assigned weights.

Finally, in Chapter 4 we introduce an ensemble method called Enriched Random Forest for high dimensional data (where $n << p$, n is the number of observations and p are the features). This algorithm can address situations where the dimension is very high but only a very small fraction of these features is truly informative. We evaluated our proposed algorithms both empirically and asymptotically using real and simulated datasets.

Acknowledgements

I wish to express my deep appreciation and gratitude to my advisors Professor Michael Katehakis and Professor Javier Cabrera for introducing me to this area of research. Their guidance and encouragement have made this thesis possible.

I am greatly indebted to Professor Adi Ben-Israel for his time, valuable suggestions and ideas. I express my sincere thanks to Professors Jerome Williams, Dmitri Metaxas, Spiros Papadimitriou for their participation in the dissertation committee. I would also like to thank Mariusz Lubomirski, Director, Janssen Research & Development U.S., for agreeing to serve as the external member on my committee.

I take this opportunity to acknowledge the administrative and financial support I received from Rutgers Business School Ph.D. Program. I am grateful to my friends and colleagues at the Department. Specially, I would like to thank Tanay Talukdar, who has been a great friend and always supported me throughout this journey.

I am also profoundly grateful to my co-authors for their hard work and substantial contribution to uplift the studies presented in this thesis. My heartfelt gratitude to Chancellor Nancy Cantor for her unwavering support during this long academic journey.

Finally, I thank my parents for believing in me and always encouraging me to pursue my goals and follow my dreams. Undoubtedly I could not have done this without them.

Dedication

This thesis is dedicated to my parents

Pinaki and Aninda Ghosh

Thank you for your dedicated partnership for success in my life.

Table of Contents

List of Tables

List of Figures

Part I:

Background and Motivation

Chapter 1

Introduction

Big Data has created enormous opportunities for the modern society and businesses. The ability to process large amounts and variety of data, has pushed the boundaries of traditional computational methods. For example, in genomics, there have been more than 5,00,000 microarrays that are made available for researchers. Each of these arrays contain tens and thousands of molecular expressions. In biomedical engineering, there have been large terabytes of functional magnetic resonance image (fMRI) data with each image containing more than 50,000 voxels. Other examples of Big Data include unstructured text corpus, social media, fiancial time series, e-commerce, retail transactions, and surveillance data.

Big Data does not come without its share of challenges - "Extracting data is not same as extracting information". Massive sample size and high dimensionality introduce unique challenges, including scalability, noise accumulation, spurious correlation, incidental endogeneity and measurement errors. The performance of traditional statistical methods are largely hindered by these unique challenges and underscore the need for methods that are faster and efficient.

This dissertation studies two important problems that arise in the analysis of Big Data. First, the large sample size and second, the high dimensionality. The dissertation is divided into three parts.

PART I: Background and Motivation

This chapter provides a general description of the dissertation. It briefly describes the theme of each chapters and the major contributions.

PART II: Clustering Big Data

This part develops models and algorithms for clustering datasets containing large number of observations and features. Chapters 2 and 3 present two different approaches for clustering observations with weights. Assigning weights helps reduce the size of dataset, thus addressing the scalability issue that arises when dealing with large datasets.

Particularly, chapter 2 describes a two-way clustering approach that combines model-based and weighted K-means clustering methods. The underlying concept of two way clustering approach include sampling and data reduction. It also adopts a divide and conquer strategy by creating small subgroups of the binary features that are of size p or less, so that the possible number of patterns (2^p) is small enough to be efficiently handled by the traditional clustering approaches. The method has been applied on a study that assessed the impact of comorbidity on patient health outcomes.

In Chapter 3, we present two clustering techniques that are suitable for large datasets. First, it derives a weighted probabilistic distance-based technique adjusted for cluster weights. Second, it derives a probabilistic approximation of capacitated clustering problem, where the cluster sizes are specified as constraints. Both these methods are capable of handling weighted reduced data.

PART III: Classification on High-Dimensional Data

Here, we introduce a novel ensemble method that works well on high-dimensional data. In chapter 4, we develop an algorithm called enriched random forest that addresses the limitations of traditional random forests in setting where number of features is significantly large compared to the number of observations, and the percentage of truly informative features is very small.

1.1 Paradigm Shifts

Big Data ushers a new era of empiricism, wherein the volume of data and advanced data analytics have caused paradigm shifts across multiple disciplines. Such shift has led to significant progress in development of algorithms that are scalable to massive high-dimensional data.

1.2 Clustering Big Data

Clustering algorithms have emerged as powerful meta-learning tool for accurately analyzing massive volume of data generated by modern applications. The main goal of clustering algorithms is to categorize data into clusters such that objects grouped in the same cluster are in some sense similar. There is a vast body of literature that focus on clustering and there has been attempts to analyze and categorize these methods for large number of applications. But when it comes to Big Data, new challenges are raised and there is lack of clear consensus amongst practitioners as to which algorithm would be the most appropriate for a given Big Dataset.

A comprehensive study of clustering algorithms pointed out that no clustering algorithm performs well for all the evaluation criteria. Expectation Maximization (EM) and fuzzy clustering algorithms performs well on moderate size data with respect to cluster quality but fail for high-dimensional data. These algorithms suffer from high computational time requirements. Another problem that often arises is cluster instability.

Two well known approaches for clustering Big Data include single-machine clustering and multiple-machine clustering. Single-machine algorithms are based on sampling and dimension reduction techniques. Sampling attempts to improve speed and scalability by performing clustering on a sample of the datasets and then generalizing to whole dataset. Dimension reduction techniques project the original dataset to a lower dimensional space and perform clustering on the projected space.

Multiple-machine clustering techniques have attracted more attention owing to higher scalability and faster response time. Parallel clustering and MapReduce based clustering use distributed computing and offer impressive scalability and speed compared to serial counterparts. However, the complexity of implementing these algorithms is a challenge.

In part II of this dissertation, we considered the single-machine approach and focused on reducing the large sample size by assigning weights to observations. Say, an observation has weight two, which would imply that there are two almost identical

observations in the original dataset. Traditional clustering algorithms when extended to handle observation weights can improve scalability on large datasets. We also proposed another technique based on divide and conquer approach to tackle the high-dimensionality problem. We apply a two-way clustering to partition the feature space in to smaller subspaces and then perform clustering on these subspaces.

1.3 Building Classifiers on Big Data

When applying machine learning algorithms on high-dimensional data, a critical issue is the "curse of dimensionality". When data becomes sparser in high-dimension, it adversely affects the algorithms that were designed for low-dimensional space. Accumulation of noise is also severe in high dimensions and may dominate the true signals. The discriminative power of the classifiers becomes very low due to too many weak features. In other words, variable selection becomes increasingly important.

However, in high dimensions variable selection is challenging due to spurious correlation. That means many uncorrelated random variables may have high correlations in high dimensions. Spurious correlations can give rise to false scientific discoveries and wrong statistical inferences. Cross-validation methods help to attenuate this problem.

Another subtle issue raised by high-dimensionality is incidental endogeneity. In regression setting, this would imply that some predictors correlate with the residual noise. The exogenous assumption that residual noises are uncorrelated is crucial for model consistency and its violation could result in models being statistically invalid.

Unlike spurious correlation, incidental endogeneity refers to genuine existence of unintentional correlation among variables due to high-dimensionality. For instance, spurious correlation could be analogous to finding two identical individuals but have no genetic relation, where as incidental endogeneity is likened to occasionally running into an acquaintance in a big city. More generally, endogeneity occurs as a result of selection biases, measurement error and omitted variables.

Keeping these challenges in mind, in part III, we considered a specific problem of the high-dimensional setting where the sample size is extremely small compared to the

feature dimension. Ensemble techniques such as random forests works well in general high-dimensional datasets. However, when the number of features is extremely large compared to the number of samples and the percentage of truly informative feature is very small, performance of traditional random forest significantly decline. Chapter 4 describes the problem in detail and presents an enhanced technique that can address this problem. We refer to it as Enriched random Forest(ERF).

Part II:

Clustering Big Data

Chapter 2

Comorbidity Patterns and its Impact on Health Outcomes: Two-way Clustering Analysis

A paper appeared in IEE Transactions on Big Data, 2016

Debopriya Ghosh, Javier Cabrera, Tarek N. Adam, Petros Levounis, Nabil R. Adam

Abstract

Comorbidity greatly increases the complexity of managing disease in patients. Approximately 27% of the US population have two or more concurrent comorbid conditions. Traditional models for assessing the impact of patient demographic and comorbidity burden on patient health outcomes, represented comorbidity conditions by Charlson Comorbidity Index. In this paper, we develop a novel two-way clustering approach combining model-based and weighted K-means clustering methods for characterizing and summarizing a patient's comorbid conditions. Our two-way approach helps reduce the size of the data to a manageable size, thus being practical for big data applications. Another novel aspect of our approach is the ability to handle weighted observations. Assigning weights to observations helps reduce the size of the dataset, thus addressing the scalability challenge of algorithms when dealing with big data. Using the National Inpatient Sample database for 2008-2013, we evaluate the performance of our approach by the use of logistic regression and support vector machine models by applying them to patients whose primary diagnosis is cardiovascular disease. In addition to evaluating our proposed method using empirical test data, we use asymptotic statistics. Both evaluation methods show that the proposed approach improves the prediction of patient health outcomes; specifically, hospital length of stay.

2.1 Introduction

Cormorbidity is defined as the presence of one or more medical or psychiatric conditions in addition to an index disease in one patient [1]. Multimorbidity, on the other hand, refers to the co-occurrence of multiple medical or psychiatric conditions within one patient without any reference to an index disease. More than one in four Americans (approximately 27%) have two or more concurrent comorbid conditions, including, for example arthritis, asthma, chronic respiratory conditions, diabetes, heart disease, human immunodeficiency virus infection, and hypertension [2]. In addition to comprising physical and medical conditions, comorbidities also include problems such as substance use and addiction disorders, mental illness, dementia, neurocognitive impairment disorders, and developmental disorders. The prevalence of comorbidity is substantial among older adults, even though there are many Americans with comorbid conditions under the age of 65 years. As the number of comorbid conditions in a patient increases, the risk of various health outcomes such as mortality, poor functional status, unnecessary hospitalizations, adverse drug effects, duplicative tests and conflicting medical advice increases. Resource implications for addressing multiple comorbid conditions are immense. Reports ascertain that 66% of total healthcare spending is directed towards care for the approximately 27% Americans with comorbid conditions [2].

In the U.S., close to 80% of Medicare spending is devoted to beneficiaries with 4 or more chronic conditions, with costs increasing exponentially as the number of comorbid conditions increases [1]. Patients with comorbidity face substantial challenges related to the out-of-pocket costs for their care, including higher costs for prescription drugs and total out-of-pocket healthcare. Overall, the population with comorbidity is characterized by tremendous clinical heterogeneity and substantially varies in the number of comorbid conditions, severity of illness, and functional limitations. Developing means for determining homogeneous sub-groups among this heterogeneous population is viewed as an important step in the effort to improve the health status of the total population and only recently is beginning to be addressed by researchers. Neither the treatment of comorbidity nor the impact of comorbidity on patients' health status over

time have been well characterized in the literature [2]. Although many long term national surveys have been conducted worldwide in order to determine the impact and magnitude of health problems in terms of comorbidity, as well as the role of health programs and healthcare providers, there is a fundamental lack of knowledge about how to appropriately measure comorbidity within one patient and quantify the heterogeneity in comorbidity patterns among patients.

As suggested in [2], research identifying the most common patterns of comorbidity can help in targeting specific interventions for the specific subgroups and monitoring the impact of those interventions. In the area of diabetes, for example, Piette et al. [3] studied the impact of comorbid conditions on diabetes care. Multiple chronic conditions that are common among patients with diabetes account for much of the morbidity they face. Health problems that used to be treated in inpatient settings are increasingly managed within outpatient care, thus straining the provider resources for addressing diabetes specific management goals. They [3], presented a framework for considering the ways in which comorbid chronic conditions can influence diabetic patients' medical care, self-management, and health outcomes. Such a framework may assist healthcare systems and researchers in developing more effective models for improving diabetes care in the context of comorbidities.

In the area of bipolar disorder, Kilbourne et al.[4] pointed out the lack of comprehensive population-based studies on the prevalence of general medical comorbidities among patients with bipolar disorder. Their research, which used the Veterans Administration National Patient Care Database, focused on treatment of coexisting medical comorbidities that may reduce the risk of adverse outcomes among patients with bipolar disorder. A better understanding of the burden of general medical conditions is an important step toward improving outcomes for patients with bipolar disorder. Their results show that the most prevalent conditions among patients with bipolar disorder included cardiovascular (e.g., hypertension, 35%), endocrine (e.g., hyperlipidemia, 23%; diabetes, 17%), and alcohol use disorder (25%).

As mentioned above, a better understanding of comorbid conditions may result in improved patient health outcomes. A widely used method for assessing a patient's

comorbidity is the Charlson Comorbidity Index (CCI) [5], which uses the sum of the number of diagnosed diseases without any weighting. This method has several limitations [6], including equal scoring of all diagnoses without accounting for the impact of different diseases' severity on patient health outcomes; ignoring potentially important relationships among diseases that might differ from their simple sum, for example the interaction between chronic obstructive pulmonary disease and congestive heart failure might exceed the simple sum, whereas cardiovascular disease related to diabetes might be over weighted in an index that counts both independently, thus resulting in incorrect realization of the impact of the comorbidity. Furthermore, as numerical indices do not account for multimorbidity by chance they often require clinical judgment for gathering information on each medical condition.

Cornell et.al. [7], describes and illustrates the application of cluster analysis to identify clinically relevant multimorbidity groups. The authors elucidate that application of cluster analysis involves a sequence of critical methodological and analytic decisions that influence the quality and meaning of the clusters produced. In their paper they illustrate the application of cluster analysis to identify multimorbidity clusters in a set of 45 chronic illnesses in primary care patients (N = 1,327,328), with 2 or more chronic conditions, served by the Veterans Health Administration. Six clinically useful multimorbidity clusters were identified: a Metabolic Cluster, an Obesity Cluster, a Liver Cluster, a Neurovascular Cluster, a Stress Cluster and a Dual Diagnosis Cluster. Ng [8] proposed a new theoretical framework, using a two-way clustering approach to identify clusters of most significant non-random comorbid conditions and disparities in multimorbidity patterns among patients. The author applied a clustering-based approach to determine the association between multimorbidity patterns and patient health outcomes and to calculate a multimorbidity score for each patient.

In this paper we address the problem of assessing a patient's comorbidity. we develop a novel two-way clustering approach combining model-based and weighted K-means clustering methods for characterizing and summarizing a patient's comorbid conditions that is practical for big data applications. The model based clustering is based on correlation estimates among comorbidities that take into account the occurrence by chance

of coexisting conditions, controlling for the false discovery rate, thus avoiding spurious correlation among comorbid conditions. Based on our proposed approach, different patterns of comorbid conditions in patients are captured through cluster indicators. These comorbidity cluster indicators are then used to predict patient health outcomes, specifically, hospital length of stay. These new cluster indicators are used as predictors in logistic regression and support vector machine models to improve the prediction of patient health outcomes such as hospital length of stay. Another novel aspect of our approach is the ability to handle weighted observations. Assigning weights to observations helps reduce the size of the dataset, thus addressing the scalability challenge of algorithms when dealing with big data. Furthermore, our two-way approach helps reduce the big size of the data to a manageable size. Specifically, our approach results in few small subgroups of comorbid conditions that are of size k or less, so that the possible number of patient comorbidity patterns 2^k is small enough to be easily and efficiently handled by our patient clustering algorithms. Considering available commodity computing resources, a value of k that is not greater than 16 seems to be best.

The remainder of the paper is organized as follows. A discussion of the related work is presented in section 2.2. In section 2.3, we discuss the real-world dataset and the details of the proposed approach. The experimental evaluation of the proposed approach and discussion of the results are included in sections 2.4 and 2.5. The conclusion and future work are presented in section 2.6.

2.2 Related Work

The co-occurrence of two or more diseases in a given patient is determined by a number of factors, including treatment-induced, environmental, or lifestyle-related factors. Comorbidity can be viewed as disease-disease association. Studying disease-disease association contributes towards improving our current knowledge of disease relationships which may lead to further improvement in disease diagnosis, prognosis, and treatment. Statistically significant correlations between the underlying structure of biological networks and disease comorbidity patterns can be identified through a combination of

information on cellular interactions, disease-gene associations, and population-level disease patterns extracted from clinical databases and electronic health records. Network theory has provided insight into the properties of biological networks, which enabled addressing some fundamental properties of the genes involved in disease.

Several studies have considered data on shared genes, protein-protein interactions, and co-expression patterns to identify co-occurrence of two or more diseases in a given patient. For example, in Charlson et al. [9] the authors identified disease neighborhoods from protein-protein interaction networks. Using protein level data, one can then identify coexisting diseases. They developed a novel Disease Module Detection algorithm to explore the local network neighborhood around a given set of known disease proteins. Examples of other similar studies are [10]-[18].

Hwang et al. [19] performed supervised co-clustering of phenotypes and genes simultaneously by integrating various sources of phenotypic and genomic data as well as prior knowledge. Their approach enabled discovery of disease classes based on the molecular underpinnings of the phenotypes and the molecular interactions in a network.

In [20], the authors proposed a phenomenological comorbidity network of diseases that is based on medical claims data. The network was made up of two layers. The first layer contains links representing the conditional probability for a comorbidity while links that contain respective statistical significance are in the second layer. They showed that the network undergoes dramatic structural changes across the lifetime of patients. To understand the spreading of diseases at the population level, they introduced a simple diffusion model and were able to show that patients mostly develop diseases that are in close network proximity to disorders that they already suffer.

In a study that aimed at finding the groups of ICD-9 diagnosis codes from electronic health records (EHRs) that can predict the improvement of urinary incontinence of home health care patients and are also interpretable to domain experts, Dey et al. [21] proposed two approaches for increasing the interpretability of the obtained groups of ICD-9 codes. First, they incorporated prior information available from the clinical classification system. This is followed by incorporating additional types of clinical information for the same patients, e.g., demographic, behavioral physiological, and

psychosocial variables (available from survey questions during the hospital visits). They finally developed a hybrid framework that combines both prior information and the data-driven clinical information in the predictive model framework. By applying sparse-canonical correlation analysis they were able to find the relationship between the ICD-9 codes and the clinical survey variables.

To better understand the disease evolving patterns, Liu et al. [22] proposed a novel graph based representation for patient EHRs, which captures temporal relationships among distinct medical events. The temporal graph can capture temporal relationships of the medical events in event sequence, thus it is informative in predictive modeling as well as other challenging analytics tasks. The temporal graphs provide a summary of the longitudinal data and as such are resistant to noisy and irregular observations. Furthermore, by expressing the temporal graphs with the phenotypes, the expressing coefficients can be used for such applications as patient segmentation, personalized medicine, and disease diagnosis.

A number of factors including treatment-induced, environmental, or lifestyle related factors can result in co-occurrence of two or more diseases in a patient. Studying disease-disease association contributes to identifying correlation among various diseases to group clinically relevant comorbidities. Disease-disease associations can be identified based on similarity of clinical phenotypes as well as based on underlying biological mechanisms of the diseases. The above mentioned studies [9]-[18] used either biological network datasets or a combination of biological network and clinical datasets. These studies showed that disease-disease associations predicted by different network based methods are correlated with associations derived from standard disease classification systems (ICD-9) and comorbidity data (see section 2.3.4). In contrast, our study uses clinical datasets only to characterizing and summarizing a patient's comorbid conditions that are relevant to patient health outcomes. Clinical datasets, specifically similar to the one we are using in this study, provide patient demographics, health outcomes, and prevalence of comorbid conditions, it does not, however, provide genetic, proteomic and metabolic network data, thus not making it possible for us to apply network based methods to identify correlation among diseases as opposed to using prevalence based

method. While biological networks are usually incomplete and include noise, clinical datasets are more prone to selection and ascertainment bias and also capture correlations that are not only attributed to genetic modifications but also treatment induced and environmental factors.

2.3 Preliminaries

Our proposed approach can be summarized as follows: First we determine the correlation among comorbid conditions. Next, we cluster comorbidities based on those correlation coefficients and for each cluster of comorbidities we determine the patient sub-clusters by applying weighted K-means. The result is a set of clusters; each includes patients who have similar pattern of comorbid conditions. Each of these clusters is assigned a cluster indicator. We then develop logistic regression and support vector machine (SVM) models to predict hospital length of stay (LoS). Next, we perform two sets of experiments; one set where we include in each of the logistic regression and SVM models the proposed cluster indicators along with the demographic variables. In the second set of experiments, we replace the proposed cluster indicators with the traditional CCI.

Here we develop both linear logistic regression models as well as SVM models. On the one hand, a linear logistic regression model is advantageous in that it enables us to assess the importance of predictors by statistical significance and to interpret the model in detail. In addition, for big data with many predictors and noisy response, linear logistic regression is easier and faster to fit. On the other hand, SVM is the "gold standard" in prediction, so it is always useful to compare it with logistic regression to make sure the logistic model is adequate for prediction.

2.3.1 Dataset

For our study we use the National Inpatient Sample (NIS) data, which is available through Healthcare Cost and Utilization Project (HCUP). Starting with 2012, HCUP replaced the Nationwide Inpatient Sample with the National Inpatient Sample. The

NIS uses a redesigned sampling method and contains a sample of all discharges rather than a sample of all hospitals. The NIS implements an improved sample design for more accurate representation of national discharge data. In addition, the 2012 NIS redesign excluded long term acute care hospital data. NIS contains charge information on all patients, regardless of payer, including persons covered by Medicare, Medicaid, private insurance, and the uninsured. Hospitals are divided into nine geographical divisions (i.e., New England, Mid Atlantic, East North Central, West North Central, South Atlantic, East South Central, West South Central, Mountain, and Pacific) based on the standard definition developed by the U.S. Bureau of the Census. Hospitals are also classified according to population density and educational mission into rural, urban teaching, and urban non-teaching.

In order to perform multi-year or trends analyses using the NIS, the Agency for Healthcare Research and Quality (AHRQ) developed discharge trend weights associated with each discharge record. On average the weight associated with a discharge record is approximately four. Our analysis and results presented in this paper are based on these weights. Our dataset is made up of approximately 48 million discharge records (close to 8 million discharge records per year for the years 2008 through 2013). It is important to note that our dataset is a sample from the total dataset of 192 million records (4×48) in order to balance the sample to make it more representative of the total dataset, weights are assigned to each discharge record. Hence, having the weights reduces the size of the dataset. This weighting scheme is similar to the one use by the US Census Bureau 5% sample [23].

Each dataset record consists of 126 clinical and non-clinical attributes for each visit. Nonclinical attributes include patient demographics (age at admission, race, and gender), admission date, HCUP hospital identification number, hospital state, hospital zip code, length of stay in hospital in days (LoS), and total charges. Hospital level attributes include location of hospital, division, hospital bed size, and teaching status of hospital. Clinical attributes include procedures, procedure categories, diagnosis codes and diagnosis categories. The diagnosis codes are represented using the International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM).

Each record contains an array of diagnosis codes(15 diagnosis codes prior to 2008 and 25 diagnosis codes from 2009), including principal diagnosis and up to 24 secondary diagnoses.

2.3.2 Institutional Review Board Approval

This study has been approved by the Rutgers Biomedical and Health Sciences Institutional Review Board. After approval and the completion of a data user agreement, the datasets were obtained from the Agency for Healthcare Research and Quality. According to the data use agreement, individual cell counts of 10 cannot be reported to avoid the risk of identification of individual patients.

2.3.3 Selection of Cases

All hospitalizations with a primary ICD-9 diagnosis codes for cardiovascular disease have been selected for analysis in our study. According to the NIS documentation, the primary diagnosis refers to the primary reason for hospitalization. For the specific ICD-9 codes see http://www.icd9data.com/2013/Volume1/default.html

2.3.4 Comorbid Conditions

An estimate of the comorbidity burden among hospitalized patients was derived using the comorbidity software version 3.7 (written in SAS), which, is one of the software tools developed as part of the HCUP. We used SAS version 9.4. The 29 comorbid conditions were as follows [24]: congestive heart failure, valvular disease, pulmonary circulation disorders, peripheral vascular disease, and hypertension (both uncomplicated and complicated), paralysis, other neurological disorders, chronic pulmonary disease, diabetes without chronic, diabetes with chronic, hypothyroidism, renal failure, liver disease, chronic peptic ulcer disease (includes bleeding only if obstruction is also present), HIV and AIDS (Acquired Immune Deficiency Syndrome), lymphoma, metastatic cancer, solid tumor without metastasis, rheumatoid arthritis or collagen vascular, coagulation deficiency, obesity, weight loss, fluid and electrolyte disorders, blood loss anemia, deficiency anemias, alcohol abuse, drug abuse, psychoses, depression.

2.3.5 Variables Examined

The primary outcome variable of interest is hospital length of stay in days (LoS). The independent variables of interest include the patient demographics and clinical conditions. The demographic variables are: patient's age at the time of hospitalization – race (White, Black, Hispanic, Asian or Pacific Islander, Native American, and other), payer (Medicare, Medicaid, private including HMO, self-pay, no charge, and other), patient household income level quartile according to zip code (zip income). The household income level quartiles are identified by values of 1 to 4, indicating the poorest to wealthiest populations. These values are derived from ZIP Code-demographic data obtained from Claritas. These estimates are updated annually; the value ranges vary by year.

2.4 Two-Way Clustering Approach

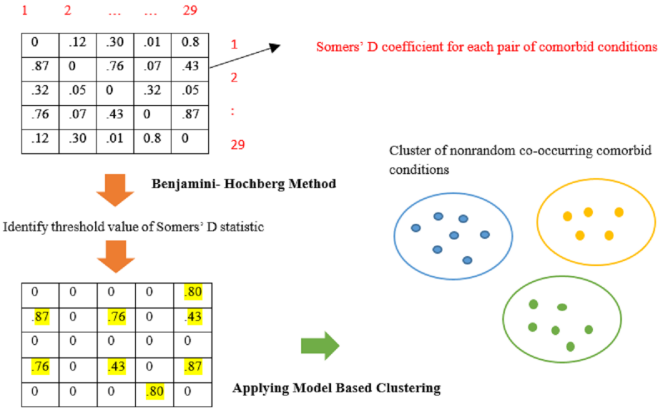
Below is an overview of our approach, this is followed by a detailed discussion of each step.

1. Estimation of Non-random Comorbidities. To estimate the co-occurrences of two conditions, we need to correct for occurrences by chance, thus avoiding overestimation of non-random comorbid conditions. We apply asymmetric version of weighted Somers'D statistic [5] to provide a quantitative measure of comorbidity that accounts for co-occurrence of conditions by chance and controls for the false discovery rate.

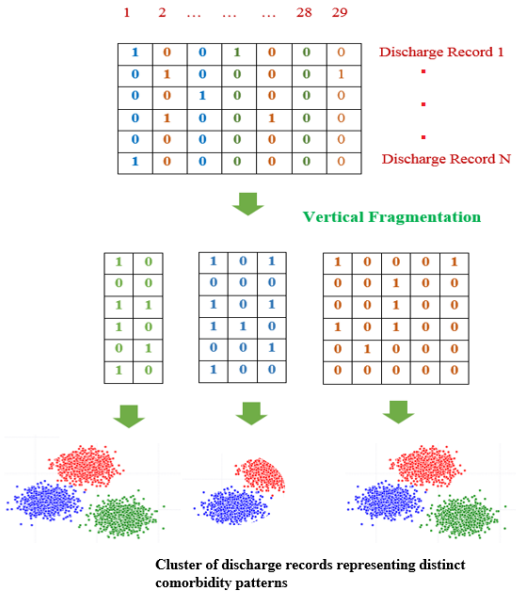
2. Clustering of comorbidities. In order to identify the clusters of co-occurring comorbidities, we apply model based clustering. Further, to identify distinct comorbidity patterns within each of these major clusters we apply weighted K-means.

2.4.1 Estimation of Non-random Comorbidities

In general, the proportion of co-occurrence of comorbid conditions is small relative to co-occurrence by chance. Employing odds ratio or relative risks to estimate co-occurrence



(a) Clustering of Comorbid Conditions



(b) Clustering of Patient Comorbidity Patterns

Figure 2.1: Two phase clustering method.

of comorbid conditions may result in overestimation of nonrandom comorbidity even though the comorbidity is not strong. To illustrate, consider the observed counts of Epilepsy and Stroke in the 2007 Australian National Survey of Mental Health and Wellbeing Dataset (see Table 2.1). As discussed in [5], the odds ratio of epilepsy for '(stroke absent)' is 0.0087 and the odds ratio of epilepsy for '(stroke present)' is 0.0308. The odds ratio for stroke is 3.556 (ad/bc), which is significant with a 95% confidence interval of 1.62 to 7.81. For comorbidity, only 7 out of 81 (about8.6%) subjects who have had a stroke have epilepsy. Similarly, only 7 out of 234 (about 3.0%) subjects with epilepsy have had a stroke. Thus, association does not necessarily imply comorbidity since odds ratio does not directly measure the amount of co-occurrence, nor does it adequately separate non-random comorbidity from random coincidental comorbidity.

Table 2.1: Observed Counts for Epilepsy and Stroke in 2007 Australian National Survey of Mental Health and Wellbeing Dataset

		Epilepsy		
		Absent	Present	Total
Stroke	Absent	a = 8,533	b = 74	R1= 8,607
	Present	c = 227	d = 7	R2 = 234
	Total	C1 = 8,760	C2 = 81	N= 8,841

To overcome this overestimation problem, similar to [5], we apply Somers'D statistic which takes into account the occurrence by chance.

Somers'D statistic [5] explicitly adjusts for expected coincidental comorbidity by chance. We consider the asymmetric version of Somers'D statistic for measuring the comorbidity, which is defined as

$$Somers'D \leftarrow s\frac{(P_Q)}{\min(W_r, W_c)} \tag{2.1}$$

where, $P = ad$ (concordant pairs) ; $Q = bc$ (discordant pairs); $Wr = P + Q + T_R$ and $Wc = P + Q + T_C$. $T_R = (ab + cd)$ and $T_C = (ac + bd)$ are the numbers of tied pairs on row ordinal variables only and column ordinal variables only respectively.

After computing the pairwise comorbidity measure in terms of the asymmetric version of Somers'D, we consider the significance of the multiplicity problem [25] and determine the cut off value beyond which the pairwise comorbidity is considered significant. We apply the Benjamini-Hochberg procedure [25] for controlling for the false discovery rate (FDR). Considering 29 comorbid conditions, as mentioned above, the proximity matrix corresponding to the $n_s = 29(29-1)/2 = 406$ pairwise Somers'D statistics among the health conditions is computed. In this study, we set the threshold value for a significant Somers'D, statistic to the 90th percentile of the p-value.

2.4.2 Measure of Patient Comorbidity

Traditionally, patient comorbidity has been measured using numeric indices that were originally developed and validated for specific diseases. Examples of such indices include, the Kaplan index for diabetes, and the Charlson index for the prediction of mortality (although the Charlson comorbidity index has been adapted for use with other outcomes, including length of stay). In particular, Charlson comorbidity index is used to model many public health and pharmaceutical indicators such as cost of treatment [26], [27]. These indices do not account for comorbidity by chance.

To address these limitations of current methods of measuring comorbidity, we propose a two-way clustering-based method. In the first phase we apply model-based clustering to group the 29 comorbid conditions into a set of clusters (major clusters). To determine the optimal number of clusters, i.e., k, we repeatedly compute the Silhouette Statistic (Silhouette width) for each $k = 1, 2, \cdots, 29$ and k is chosen having maximum Silhouette width. Assuming the data is clustered into k clusters, for each datum i, let $a(i)$ be the average dissimilarity of i with all other data within the same cluster. We then define the average dissimilarity of point i to a cluster c as the average of the distance from i to all points in c. Let $b(i)$ be the lowest average dissimilarity of i to any other cluster, of which i is not a member. The silhouette statistic can be defined as:

$$s(i) \leftarrow \frac{a(i) - b(i)}{max(a(i), b(i))} \tag{2.2}$$

In the second phase, we identify sub-clusters within each of the major clusters, where each of these sub-clusters is to be formed through different patterns of comorbid conditions prevalent among patients. There are two justifications for phase one of the procedure. First, we want to combine comorbidities that are correlated. Second, our clustering algorithm compresses the data into unique observations with a weight. For example, if we have 29 comorbid conditions then there are $2^{2^9} = 536,870,912$ maximum number of possible combinations resulting in a very large clustering problem. However, if we consider a group of 13 comorbidities there will be only 8,192 possible combinations that are easy to cluster. Again this is one more way that we use to reduce the big size of the data to a manageable size.

Algorithm 1 Algorithm for Two-way Clustering

Input: $C \in \mathbb{R}_{N \times q}$, patient comorbidities
Output: $CI \in \mathbb{R}_{N \times k}$, cluster indicators

Initialization :

Define, $C \leftarrow \{C_1, C_2, ..., C_N\}$, set of patient comorbidities for N patients, where $C_i \leftarrow \{c_1, c_2, ..., c_q\}$ is the set of comorbid conditions in patient i and $c_j \in \{0, 1\}$ $\forall$ $q \leftarrow 1, ..., 29$
Compute the proximity matrix, $S \leftarrow SomersD(C)$;
Compute the distance matrix, $D \leftarrow 1 - \frac{\|S\|}{quantile(\|S\|, 0.9)}$;
$CL \leftarrow MClust(D)$, where $CL \leftarrow \{CL_1, CL_2, ..., CL_k\}$
$j \leftarrow 1$;

LOOP Process

for $i \leftarrow 1$ to k **do**
    $cols \leftarrow \{CL_i\}$
    $X \leftarrow C[\ , cols]$;
    $P \leftarrow weightedKMeans(X, n)$ where n is the number of clusters;
    $CI[\ , j] \leftarrow P$;
    $j \leftarrow j + 1$;
end for
return P

Next we discuss our clustering methodology. We would like to cluster a dataset with n observations and p variables and each observation has a weight w_i that may represent repeated observations or a case importance. In addition, variables could also be weighted according to decisive relevance V_j. We want to cluster the n observations

into $k \leq n$ clusters. As mentioned above, having unique cases with weights greatly reduces the size of the dataset. In the era of big data, size and dimensionality reduction are critical to achieving computational efficiency of learning algorithms. A popular choice for clustering is the K-means algorithm. Given the presence of weights assigned to each observation, there is a need for extending the traditional K-means algorithm such that it can: (i) handle both sample and variable weights, (ii) overcome the large computational cost of dealing with big data. Here, we develop a modified version of the K-means algorithm, called the weighted K-means, and apply that to our data. The weighted K-means (WKM) algorithm is similar to the standard K-means algorithm except that it incorporates the case and variable weights to the within sum of squares (WSS) criterion that is minimized.

The basic idea of the WKM clustering algorithm is as follows: Suppose there are K clusters, $k = 1, \cdots, K$, and the cluster means are $\mu_1, \cdots, \mu_K$. Then our new criterion function is

$$WSS(k) \leftarrow \sum_{i,j,k} W_{i,k} V_j (x_{i,j,k} - \mu_{k,j})^2 \tag{2.3}$$

where $\mu_{k,j}$ is the j^{th} component of μ_k, and $x_{i,j,k}$ is the i^{th} row and j^{th} column of the observations belonging to k^{th} cluster, and W_{ik} is the weight corresponding to the i^{th} observation of k^{th} cluster. The working principle of the weighted K-means is same as that of traditional K-means. Its a partitional clustering algorithm, that iteratively minimizes the within-cluster variances. Given an initial set of K means $\mu_1, \cdots, \mu_K$, the algorithm proeeds by alternating between two steps:

(i) Assignment step: Each observation is assigned to the nearest cluster, determined by the distance from cluster centers.

(ii) Update Step: Re-calculate cluster centers for observations assigned to each cluster.

The algorithm converges when the assignments no longer change. The algorithm does not guarantee global optimum. The objective function is to minimize the within cluster sum of squares. For weighted K-means method, the objetive function is thus

written as,

$$\min_{\mu_1,\cdots,\mu_k} \quad \sum_{i,j,k} W_{i.k} V_j (x_{i,j,k} - \mu_{k,j})^2 \tag{2.4}$$

The algorithm stop when the centers no longer change.

This two-way clustering results in assigning each patient to a cluster which is characterized by a unique pattern of co-occurring comorbid conditions. The cluster indicator serves as a measure of patient comorbidity. Thus, rather than having a numeric index, the cluster indicator captures the severity and the non-random co-occurrence of comorbid conditions in a patient.

2.5 Experimental Evaluation

We implemented the proposed approach and used the NIS dataset to evaluate its effectiveness of capturing patients' comorbidity patterns as compared to the traditional CCI when predicting patients' outcomes.

Using the twenty-nine comorbid conditions, we applied the asymmetric Somers'D statistic to identify the non-random co-occurring comorbidities. The result of this step, is a proximity matrix of significant non-random co-occurring comorbid conditions. We then applied model-based clustering on this proximity matrix. This resulted in three major clusters of sizes 16, 5, and 8 comorbid conditions each. Each of these clusters represents a pattern of a set of comorbid conditions that non-randomly occur together.

Our next step was to group patients with similar pattern of comorbidities. For each of the three major clusters we grouped the patient discharge records according to the different patterns of co-occurrence of the comorbid conditions making up this cluster. We fragment the dataset into three vertical fragments corresponding to comorbid conditions making up the three major clusters. The first vertical fragment contained patient discharge records having the ten comorbid conditions of the first major cluster, the second fragment contained patient discharge records having the six comorbid conditions of the second major cluster, and the third fragment contained the discharge records associated with the thirteen comorbid conditions making up the third major

cluster. Figure 2.1 depicts the three subsets.

We apply weighted K-means to each of the three dataset fragments, resulting in six, seven, and seven sub-clusters for each of the three major clusters respectively. Here the six sub-clusters represent six different patterns of co-occurrence of comorbid conditions among patients. Similarly, the first seven and the second seven sub-clusters. As a result of this process we are able to identify the pattern of comorbid conditions of each patient based on the sub-cluster it belongs to. Thus, each discharge record has a sub-cluster indicator(for short a cluster indicator).

2.5.1 Effect of Comorbidities on Health Outcomes

In order to assess the effectiveness of the proposed measure of comorbidity, we develop various linear logistic regression and SVM models for predicting patient outcomes, specifically, hospital length of stay. The models include the standard Charlson comorbidity sum index (CCI), as well as our proposed cluster indicators, that represent specific pattern of comorbid conditions. We group the outcome variable into two ordered categories: $LoS \leq 3$ days and $LoS > 3$ days. The cutoff of 3 days was chosen since it balances the two response groups. Next we perform linear logistic regression using the package glmnet in R [29] that uses modern penalized optimization in order to train the model. The second method was SVM implemented by the e1071 R package [30].

We develop four models separately for three sets of predictors which are (i) All demographic variables, (ii) CCI, and (iii) Cluster indicators (CI1, CI2, CI3) as follows,
(i) Demographics- M1
(ii) Demographics and CCI- M2
(iii) Demographics and the three cluster indicators- M3
(iv) Demographics, CCI, and the three cluster indicators- M4

To evaluate our models, we apply the holdout approach. Specifically, the dataset (consisting of 8,00,000 observations) was partitioned by random sampling into training (75%) and testing (25%) set. We fit the models on the training data and evaluated the prediction on the test data. The performance of each model was estimated by

calculating the percentage-correct prediction of LoS group on the test set. Threshold value of 0.55 was chosen as the optimum probability cutoff for maximum accuracy on the training set. Further, we compare the performance of the linear logistic regression models to SVM. Given the large training set we used, only a random sample of 80,000 observations was considered for training and 25,000 observations for testing. In order to get the best accuracy on the training set, we chose a probability cutoff of 0.58.

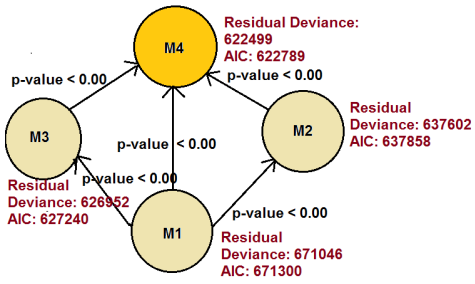


Figure 2.2: Hierarchical comparison model to evaluate proposed methodology.

Another way to compare the hierarchy of models in Figure 2.2, is to perform likelihood ratio chi-square tests for pairs of models that are nested one within the other and evaluate whether the likelihood improvement by adding new variables is statistically significant. We also compare the sizes of the likelihood ratios in order to identify the larger differences.

All statistical analyses were performed using R Version 3.1.1 and R Studio Version 0.98.1056 (The R Foundation, Vienna, Austria) statistical software.

2.6 Discussion

Descriptive statistics (frequencies and means) have been used to summarize the prevalence of estimates of hospitalizations related to patients whose primary diagnosis is cardiovascular disease. Table 2.2 includes the number of admissions and average LoS stratified by age, gender, race, payer, and zip-income. For each of the two LoS groups (≤ 3 and > 3), the table includes the corresponding number of admissions.

Table 2.3 includes the results of the performance of the different logistic regression

Table 2.2: Descriptive Statistics

	N	LoS ≤ 3	> 3	Avg. LoS
Total cases:	1,012,005			
Cases Analyzed :	811,923			
AGE (yrs.)				
≤ 20	3,133	1,753	1,380	4.72
21-40	38,769	27,247	11,522	3.46
41-60	292,014	187,623	104,391	4.08
61-80	453,728	227,660	226,068	5.40
> 80	224,361	101,312	123,049	5.38
GENDER				
Male	572,914	318,744	254,170	4.86
Female	439,091	226,851	212,240	5.04
RACE				
White	741,120	394,892	346,228	4.96
Black	134,087	76,723	57,364	4.66
Hispanic	76,226	41,608	34,618	4.97
Others	60,572	32,372	28,200	5.23
PAYER				
1	584,874	279,911	304,963	5.41
2	69,295	39,444	29,851	5.12
3	258,491	160,499	97,992	4.19
4	64,982	44,324	20,658	3.80
5	6,794	4,452	2,342	3.88
6	27,569	16,965	10,604	4.37
ZIP INCOME				
1	300,358	162,984	137,374	4.91
2	263,610	143,307	120,303	4.87
3	238,067	128,083	109,984	4.95
4	209,970	111,221	98,749	5.06

models. It shows that, including the CCI into the models does not improve the performance of the any of the models. On the other hand, when we include the proposed cluster indicators, we observe a performance improvement in all cases. This demonstrates the fact that the proposed cluster indicators are able to better capture the comorbidity patterns as compared to the CCI.

Table 2.3: Classification Performance of Various Logistic Regression Models

Model	Predictor Variables	Performance			
		Training		Testing	
		LoS ≤ 3	LoS > 3	LoS ≤ 3	LoS > 3
(i)	Demographics	64%	61%	64%	61%
(ii)	Demographics + CCI	64%	61%	65%	61%
(iii)	Demographics + Cluster Indicators	69%	65%	69%	65%
(iv)	Demographics + cluster indicators + CCI	69%	65%	69%	65%

Table 2.4 shows similar values to Table 3, indicating that, in this case, SVM performance is similar to that of the linear logistic regression models.

Table 2.4: Classification Performance of Various SVM Models

Model	Predictor Variables	Performance			
		Training		Testing	
		LoS ≤ 3	LoS > 3	LoS ≤ 3	LoS > 3
(i)	Demographics	64%	61%	63%	61%
(ii)	Demographics + CCI	64%	60%	64%	61%
(iii)	Demographics + Cluster Indicators	69%	67%	68%	67%
(iv)	Demographics + cluster indicators + CCI	69%	67%	68%	66%

As shown in Figure 2.2, Likelihood ratio chi-square tests (LRTs) show that models M3 and M4 perform better than the rest of the models and M4 is marginally better than M3.

In addition to the above analysis, we also apply AIC (Akaike Information Criterion)

to the four linear logistic regression models, as shown in Table 3. AIC estimates the quality of each model in relation to the other models, thus providing a means for model selection. Our results of the AIC measure for each of the four models respectively are: 671300, 637858, 627240, 622499. These results show that the linear logistic regression model that includes the cluster indicators along with CCI has the minimum AIC value. Also residual variability or deviances for our four models (671046, 637602, 626952, 622789) show a similar pattern as AIC, again confirming that M4 and M3 are very close compared to the others and are the preferred models. It is important to note that applying two conceptually different evaluation methods, using empirical test data (results displayed in tables 2.3 and 2.4), and using asymptotic statistics (results shown in Figure 2.2) result in the same outcome: the proposed comorbidity cluster indicators capture information about the patients' conditions and associated comorbidity patterns that are relevant to patient health outcomes, thus resulting in a better prediction of patient health outcomes.

Having established that the preferred model is the one that includes the demographic variables, CCI, and the cluster indicators, we then examine the contribution of individual predictors when predicting hospital length of stay. The results of the corresponding logistic regression model are included in Table 2.5. These results show that patient demographic variables (race, age, gender, payer, and zip-income) are statically significant along with clinical conditions as indicated by the cluster indicators representing comorbidity patterns.

Table 2.5: Summary of Logistic Regression Model

		Intercept	Std.Error	p-value	
RACE	Reference: Caucasians				
	African American	-0.021	0.010	0.031	*
	Hispanics	0.045	0.012	0.000	***
	Others	0.049	0.013	0.000	***
AGE		0.014	0.000	0.000	***
GENDER	Reference: Male				
	Female	0.042	0.006	0.000	***
PAYER	Reference: Medicare				
	Medicaid	0.102	0.014	0.000	***
	Private including HMO	-0.131	0.009	0.000	***
	Self-pay	-0.212	0.015	0.000	***
	No Charge	-0.073	0.039	0.061	
	Others	-0.097	0.020	0.000	***
ZIP INCOME	Reference: $1 -$38,999				
	$39,000 - $47,999	-0.031	0.008	0.000	***
	$48,000 - $62,999	-0.014	0.009	0.101	
	$63,000 or more	-0.003	0.009	0.761	

		Intercept	Std.Error	p-value	
ADMISSION	Reference: Elective				
	Non-Elective	0.653	0.008	0.000	***
COMORBIDITY		0.270	0.004	0.000	***
CLUSTER #	Reference: 1				
	2	0.149	0.012	0.000	***
	3	0.796	0.011	0.000	***
	4	0.302	0.013	0.000	***
	5	0.433	0.010	0.000	***
	6	0.393	0.009	0.000	***
	7	-0.347	0.039	0.000	***
	8	-1.801	0.033	0.000	***
	9	-1.297	0.034	0.000	***
	10	-0.943	0.034	0.000	***
	11	-0.513	0.037	0.000	***
	12	0.414	0.046	0.000	***
	13	-0.040	0.015	0.008	**
	14	0.216	0.018	0.000	***
	15	0.080	0.016	0.000	***
	16	0.350	0.020	0.000	***
	17	-0.149	0.015	0.000	***
	18	-0.101	0.018	0.000	***

2.7 Conclusion

In this paper, we proposed a novel two-way clustering approach for characterizing and summarizing a patient's comorbid conditions. In contrast to the standard Charlson Comorbidty Index, our proposed comorbidity cluster indicators capture information about the patient's condition and associated comorbidity patterns that are relevant to patient health outcomes. We evaluated our proposed method using both empirical test data and asymptotic statistics. Our experimental results show that the predictive model which includes patient demographics, CCI, and cluster indicators achieves 69% accuracy in predicting hospital length of stay. Our future work will focus on further improvement in the achieved accuracy of the prediction model. A possible direction to pursue is to improve our knowledge about disease relationships by combining clinical data and genetic data (biological networks) to compare and contrast the disease-disease interactions and disease co-occurrence patterns, and systematically compare significant disease patterns in clinical data with disease pairs having significant genetic overlap.

Currently, we are in the process of having our code as an R package that will implement the proposed methodology, thus making it widely available for use by other researchers.

References

[1] Valderas, Jose M., Barbara Starfield, Bonnie Sibbald,Chris Salisbury, and Martin Roland *"Defining comorbidity: Implications for understanding health and health services"*. The Annals of Family Medicine, no. 4 (2009): 357-363.

[2] *"U.S. Department of Health and Human Services. Multiple Chronic Conditions—A Strategic Framework: Optimum Health and Quality of Life for Individuals with Multiple Chronic Conditions"* . Washington, DC. December 2010.

[3] Piette, John D., and Eve A. Kerr. *"The impact of comorbid chronic conditions on diabetes care"*. Diabetes Care, no. 3 (2006): 725-731.

[4] Kilbourne, Amy M., Jack R. Cornelius, Xiaoyan Han, Harold A. Pincus, Mujeeb Shad, Ihsan Salloum, Joseph Conigliaro, and Gretchen L. Haas *"Burden of general medical conditions among individuals with bipolar disorder"*. Bipolar disorders, no. 5 (2004): 368-373.

[5] Ng, Shu Kay, Libby Holden, and Jing Sun. *"Identifying comorbidity patterns of health conditions via cluster analysis of pairwise concordance statistics"*. Statistics in medicine, no. 27 (2012): 3393-3405.

[6] Lash, Timothy L., Vincent Mor, Darryl Wieland, Luigi Ferrucci, William Satariano, and Rebecca A. Silliman. *"Methodology, design, and analytic techniques to address measurement of comorbid disease"*. The Journals of Gerontology Series A: Biological Sciences and Medical Sciences, no. 3 (2007): 281-285.

[7] John E. Cornell, Jacqueline A. Pugh, John W. Williams, Lewis Kazis, Austin F.S. Lee, Michael L. Parchman, et.al. *"Multimorbidity Clusters: Clustering Binary Data From a Large Administrative Medical Database"*. Applied Multivariate Research, Volume 12, No. 3, 2007, 163-182.

[8] S.K. Ng *"A two-way clustering framework to identify disparities in multimorbidity patterns of mental and physical health conditions among Australians"*. Statistics in Medicine, 7 May 2015, DOI: 10.1002/sim.6542.

[9] Ghiassian, Susan Dina, Jörg Menche, and Albert-László Barabási. *"A DIseAse MOdule Detection (DIAMOnD) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome"*. PLoS Comput Biol, no. 4 (2015): e1004120

[10] Davis, Darcy A., and Nitesh V. Chawla. *"Exploring and exploiting disease interactions from multi-relational gene and phenotype networks"*. PloS one 6, no. 7 (2011): e22670.

[11] Barabási, Albert-László, Natali Gulbahce, and Joseph Loscalzo. *"Network medicine: a network-based approach to human disease"*. Nature Reviews Genetics 12, no. 1 (2011): 56-68.

[12] Sun, Kai, Joana P. Gonçalves, Chris Larminie, and Nataša Pržulj. *"Predicting disease associations via biological network analysis"*. BMC bioinformatics 15, no. 1 (2014): 1.

[13] Frank Emmert-Streib, Shailesh Tripathi, Ricardo de Matos Simoes, Ahmed F Hawwa, Matthias Dehmer. *"The human disease network"*. Systems Biomedicine, 1:1(2013), 20-28, DOI: 10.4161/sysb.22816

[14] Park, Juyong, Deok-Sun Lee, Nicholas A. Christakis, and Albert-László Barabási. *"The impact of cellular networks on disease comorbidity"*. Molecular systems biology 5, no. 1 (2009): 262.

[15] Vidal, Marc, Michael E. Cusick, and Albert-Laszlo Barabasi. *"Interactome networks and human disease"*. Cell 144, no. 6 (2011): 986-998.

[16] Menche, Jörg, Amitabh Sharma, Maksim Kitsak, Susan Dina Ghiassian, Marc Vidal, Joseph Loscalzo, and Albert-László Barabási. *"Uncovering disease-disease relationships through the incomplete interactome*". Science 347, no. 6224 (2015): 1257601.

[17] Lee, D-S., J. Park, K. A. Kay, N. A. Christakis, Z. N. Oltvai, and A-L. Barabási. *"The implications of human metabolic network topology for disease comorbidity"*. Proceedings of the National Academy of Sciences 105, no. 29 (2008): 9880-9885.

[18] Kannry, Joseph L., and Marc S. Williams. *"Integration of genomics into the electronic health record: mapping terra incognita"*. Genetics in Medicine 15, no. 10 (2013): 757-760.

[19] Hwang, TaeHyun, Gowtham Atluri, MaoQiang Xie, Sanjoy Dey, Changjin Hong, Vipin Kumar, and Rui Kuang. *"Co-clustering phenome–genome for phenotype classification and disease gene discovery"*. Nucleic acids research 40, no. 19 (2012): e146-e146.

[20] Chmiel, Anna, Peter Klimek, and Stefan Thurner. *"Spreading of diseases through comorbidity networks across life and gender"*. New Journal of Physics 16, no. 11 (2014): 115013.

[21] Dey, Sanjoy, György J. Simon, Bonnie L. Westra, Michael Steinbach, and Vipin Kumar. *"Mining Interpretable and Predictive Diagnosis Codes from Multi-source Electronic Health Records"*. In SDM, pp. 1055-1063. 2014.

[22] Liu, Chuanren, Fei Wang, Jianying Hu, and Hui Xiong. *"Temporal phenotyping from longitudinal electronic health records: A graph based framework"*.. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 705-714. ACM, 2015.

[23] Albright, K. (2012). *"Specifications for Weighting the 2011 1-year, 3-year, and 5-year American Community Survey Housing Unit Samples."* DSSD 2011 American Community Survey Memorandum Series ACS11-W-10.

[24] *https://www.hcup-us.ahrq.gov/toolssoftware/comorbidity/comorbidity.jsp*

[25] Benjamini, Yoav, and Yosef Hochberg. *"Controlling the false discovery rate: a practical and powerful approach to multiple testing"*. Journal of the Royal Statistical Society. Series B (Methodological) (1995): 289-300.

[26] Udall, Margarita, Jack Mardekian, and Javier Cabrera. *I"dentification of Patients with Painful Diabetic Peripheral Neuropathy Who Have a Favorable Cost Profile with Pregabalin Treatment"*. Pain Practice 13, no. 6 (2013): 476-484.

[27] Charlson, M. E., P. Pompei, K. L. Ales, and C. R. MacKenzie. *"A new method of classifying prognostic comorbidity in longitudinal studies: development and validation"*. J Chron Dis 1987; 40: 373–383.

[28] Debopriya Ghosh, Javier Cabrera, and Nabil R. Adam. *"Weighted K-means Clustering"*. IDSLA Technical Report 2015-030.

[29] Friedman, J., Hastie, T. and Tibshirani, R. *"Regularization Paths for Generalized Linear Models via Coordinate Descent"*. (2008)

[30] Chih-Chung Chang and Chih-Jen Lin. *"LIBSVM : A Library for Support Vector Machines"*. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011.

Chapter 3

Weighted Probabilistic Distance Clustering for Big Data

Debopriya Ghosh, Adi Ben-Israel, Michael N. Katehakis

Big Data introduce statistical and computational challenges. Traditional algorithms do not scale to massive datasets, rendering them unusable or greatly impeding their performance. Reducing large sample size can greatly improve the performance of these algorithms. In this paper, we develop a new probabilistic, iterative method for clustering weighted data, using soft assignments of points to clusters with membership probabilities depending on distances and cluster sizes. We refer to it as weighted probabilistic distance (w-PDQ) clustering, where Q stands for cluster size. The novel aspect of the proposed method is the ability to handle weighted reduced data, which makes it suitable for clustering large datasets. Experiments on simulated and real data demonstrate that the weighted probabilistic distance clustering approach performs favorably to other model-based clustering approaches. In addition, the approach is robust to outliers and computationally efficient as it does not require computing complex density functions. A R package on the new algorithm is developed for public access.

3.1 Introduction

The massive sample size and high dimensionality of Big Data introduce unique statistical and computational challenges. Many traditional algorithms that were designed for moderate sample size do not scale to massive datasets, rendering them unusable or greatly impeding their performance. For example, in many applications that involve internet-scale data, containing billions or trillions of data points, even a linear pass of the entire dataset becomes unaffordable. In such cases, reducing large sample size of the data to a manageable size can help to improve scalability and performance of the algorithms.

One common approach for reducing sample size is to use weights in the analysis of data. The reduced datasets contain observations that are assigned a weight. Lets say an observation has weight of two, which would mean that the observation counts as two almost identical observations in the dataset. These weights are referred to as design weights. It has been noted that as weights primarily adjust means and proportions, it also increases standard error of the estimates. Large weights (or very small ones) introduce instability in the data. This is why researchers often trim the weights to not allow extremely large weights that can increase instability of estimates. But trimming the weights reduces representativeness of the weighted data. Data analysis methods that allow effective use of observation weights could therefore handle large sample size data that are reduced with weights.

In this paper we present a novel approach of clustering weighted data. Clustering is defined as the task of partitioning the dataset into subsets (clusters), such that data in each subset are in some sense similar and dissimilar from other subsets. Clustering is applied in tremendously diverse areas for a multitude of purposes. The clustering algorithms can be broadly classified into two types – deterministic and probabilistic. Deterministic algorithms create groups based on measures between objects, or between objects and centroids. Deterministic clustering is suited for cohesive and well-separated groups, but fails when the clusters have different geometric forms and overlap. Probabilistic algorithms on the other hand cluster data points based on a probability model.

Data is assumed to arise from a mixture model, which means that it is viewed as coming from a finite number of populations, mixed in various proportions. Each population represents a cluster with its specific characteristics. Probabilistic clustering involves computing the membership probabilities for each clusters, given a data point. The cluster having the largest probability is chosen for that data point. In contrast to deterministic clustering, probabilistic clustering allows for various geometric properties through different parameterization of the distributions, or through completely different distributions among clusters. These methods are also suitable for modeling outliers.

However, with increasing applications of Big Data, traditional approaches of clustering are no longer effective on large datasets. Lot of researches now focus on clustering Big Data. In particular, researchers have proposed the idea of clustering weighted-data in the non-parametric clustering framework to handle large size data. The weighted data clustering methods first reduce the original dataset to a smaller one by assigning each selected datum a weight reflecting the number of nearby data, and then cluster the smaller weighted dataset.

Here, we develop a weighted clustering method called weighted probabilistic distance (w-PDQ) clustering which adjusts for cluster sizes and observation weights. Our work extends the previous work of Ben-Israel and Iyigun [1]-[2]. The novel aspect of our method is the ability to handle weighted data which makes it suitable for clustering Big Data. Compared to other probabilistic methods, this method is also computationally less expensive. Firstly, unlike model based clustering which uses expectation maximization(EM) technique, w-PDQ method makes no assumption on densities and does not require computation of complex density functions. Secondly, the proposed method require no switching and works well even with cold start, where as EM methods commonly use a preprocessor such as K-means, before starting the EM process to get closer to the centers.

The remainder of the paper is organized as follows. A discussion of the related work is presented in section 3.2. In section 3.3, we formulate the problem. The algorithms for solving the problem are discussed in section 3.4. Empirical evaluations are presented in Section 3.5. Results and directions for future work are provided in Section 3.6.

3.2 Related Work

Clustering problem has been studied for years. Earliest approaches were mostly based on heuristic or geometric procedures that relied on dissimilarity measures between pairs of observations. There is a vast literature on traditional clustering methods: see for example Sharma [3], Jain and Dubes [4], and Everitt et al. [5]. The two most popular traditional clustering methods are: (i) hierarchical clustering based on the distance between groups; and (ii) K-means based on iterative relocation.

Clustering was also defined in a probabilistic framework, allowing to formalize the notion of clusters through their probability distribution. One of the main advantages of probabilistic approach is that it provides a principled statistical approach to clustering. The first works on finite mixture models were from Scott and Symons [6] and Duda et al. [7]. Since then, these models have been extensively studied, McLachlan and Basford [8], McLachlan and Peel [9], and Fraley and Raftery [10]. For comprehensive review see McNicholas [11].

Another relatively new approach is the Bayesian estimation for mixture models. the method was first studied by Gilks et al. [12], Gelman and King [13], Verdinelli and Wasserman [14], and Evans et al. [26]. Key papers in this area include Lavine and West[24], Diebolt and Robert [25], Escobar and West [15], and Bensmail et al. [16].

As pointed out previously, the clustering methods discussed above were primarily developed for moderate size data and face challenges while handling Big Data. A possible way to address the challenges, is to extend these existing methods so that they can cope with huge workloads of Big Data. Few recent papers indicate that most of the extensions rely on analyzing samples of Big Data, and vary in how the sample-based results are used to derive a partition for the overall data. For instance, in [17] Hathaway et al., developed a density-weighted c-means clustering approach for clustering a smaller, density-weighted dataset, by weighted reduction of the original data. Another similar study by Ghosh et al. [18], presented a weighted K-means algorithm for clustering reduced weighted data.

In this paper, we propose a novel clustering method for weighted data. The method

is referred to as weighted probabilistic distance (w-PDQ) clustering. It computes cluster membership probabilities based on the distance of data points to cluster centers, and directed by the cluster sizes. The proposed method is based on a measure called the joint distance function (JDF). JDF approximates the data points in its lowest contours and is harmonic mean of their distances from the cluster centers. Here, we present two different approaches: (i) when cluster sizes are unknown; and (ii) when cluster sizes are specified. The latter is known as capacitated problem. Another important feature is that in situations where the membership probabilities of a point are almost equal for the different clusters, w-PDQ method applies "power probabilities" to make the probabilities much sharper and eventually resulting in hard assignments. Weighted probabilistic distance clustering requires specifying the number of clusters. The method provides a intrinsic validation technique based on cluster uncertainty measure to determine the optimal number of clusters. To evaluate our method, we applied the w-PDQ clustering on various synthetic and real-world datasets. We also compared the performance of w-PDQ algorithm with other state-of-art clustering algorithms.

3.3 Problem Formulation

Let $\mathcal{D}$ be a data set containing $\mathcal{N}$ data points, $\mathcal{D} := \{(x_i, w_i) : \ x_i \in \mathbb{R}^n, \ w_i > 0, \ i = 1, \cdots, N\}$, where x_i is the feature vector of the data point, and w_i its weight. The objective is to partition the data set $\mathcal{D}$ into K clusters $\{\mathcal{C}_k : k = 1, \cdots, K\}$, such that points within a cluster are in some sense similar, and points in different clusters are dissimilar. The clusters $\mathcal{C}_k$ are typically disjoint sets, i.e.,

$$\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{C}_k \tag{3.1}$$

Each cluster has a representative point, or center $\mathbf{c_k}$, and distances to clusters are defined as distances to their centers $d_k(x, c_k)$. In general, the distance functions $d_k(.)$ are different for different clusters. For instance, $d_k(x, c_k) = \sqrt{\langle x - c_k, \Sigma_k^{-1} (x - c_k) \rangle}$, is the **Mahalanobis distance** corresponding to $\mathcal{C}_k$. In deterministic problems, center $\mathbf{c_k}$ of cluster $\mathcal{C}_k$ is the point $\mathbf{c}$ that minimizes the sum of its weighted distances from all points in $\mathcal{C}_k$,

$$\mathbf{c_k} := \arg\min_{\mathbf{c}} \sum_{x_i \in \mathcal{C}_k} w_i \, d_k(x_i, \mathbf{c}). \tag{3.2}$$

This calculation requires a **hard assignment** of points to clusters. In probabilistic assignment, $\mathbf{c_k}$ is computed as,

$$\mathbf{c_k} := \arg\min_{\mathbf{c}} \sum_{i=1}^{N} w_i \, \pi_k(x_i) \, d_k(x_i, \mathbf{c}) \tag{3.3}$$

where $\pi_k(x)$ is referred to as the " **assignment probability**", i.e., the **probability** of x "belonging" to $\mathcal{C}_k$, with (3.2) as a special case for hard assignment. This method allows **soft assignment** of points to clusters. A hard assignment is given by probabilities $p_k(x)$ that are all 0 or 1.

Probabilistic approximation of the clustering problem is formulated as follows:- For a given $\mathcal{D}$ and K, find centers $\{c_k : k = 1, \cdots, K\}$ so as to minimize,

$$\min_{c_1, \cdots, c_k; \pi_1, \cdot, \pi_k} \sum_{k=1}^{K} \sum_{i=1}^{N} w_i \, \pi_k(x_i) \, d_k(x_i, \mathbf{c}) \tag{3.4}$$

3.4 Algorithms

Probabilistic Clustering algorithms view the data as coming from a mixture model, where each distribution represents a cluster. The clusters have various geometric properties obtained through different parameterization of the distributions. In probabilistic clustering, given a data point, we compute its membership probabilities for each cluster. The point is assigned to the cluster having the largest probability. Below, we discuss two state-of-art algorithms for probabilistic clustering and present our proposed algorithm.

3.5 Model Based Clustering

Fraley and Raftery [10] presented the model based clustering method based on finite Gaussian mixture models. The method implements parameterized Gaussian hierarchical clustering algorithms and the EM algorithm for parameterized Gaussian mixture models with the possible addition of a Poisson noise. In the finite mixture model, each

cluster is represented by a Gaussian,

$$\phi_k(x|\mu_k, \Sigma_k) = (2\pi)^{-\frac{p}{2}} \|\Sigma_k\|^{-1/2} \exp\{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\} \tag{3.5}$$

where x represents the data, and k specifies a particular cluster. Clusters are ellipsoidal, centered at the means μ_k. The covariances Σ_k determine their geometric features.

The Gaussian finite mixture models are fitted via EM algorithm. EM algorithm iterates between an "E-step", which computes a matrix p such that p_{ik} is an estimate of the conditional probability that observation i belongs to group k given the current parameter estimates, and an "M-step", which computes maximum likelihood parameter estimates given p. In the limit, the parameters usually converge to the maximum likelihood values for the Gaussian mixture model

$$\prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \phi_k(x_i|\mu_k, \Sigma_k) \tag{3.6}$$

and the sums of the columns of p converge to N times the mixing proportions π_k, where N is the number of observations in the data. Here K is the number of groups in the data, which is assumed to be known for the purposes of the EM algorithm.

3.6 Bayesian Non-parametric Clustering

Bayesian non-parametric mixture models [19] induces a random partition model of the data points into clusters. The data is assumed to be conditionally i.i.d. with density,

$$f(x|P) = \int \phi(x|\theta) dP(\theta) \tag{3.7}$$

where $\phi(x|\theta)$ is a specified parametric density on the sample space with mixing parameter $\theta \in \Theta$. The model is completed with a prior of the unknown parameter, which in this case is the unknown mixing measure. In the general setting, parameter P can be any probability measure on Θ, requiring a non-parametric prior. Typically, the non-parametric prior has discrete realizations with

$$P = \sum_{j=1}^{\infty} w_j \delta_{\theta_j} \tag{3.8}$$

where the weights w_j and atoms θ_j are assumed to be independent and θ_j are i.i.d. from some base measure P_0. Thus, the density is modeled as,

$$f(x|P) = \sum_{j=1}^{\infty} w_j \phi(x|\theta) dP(\theta) \tag{3.9}$$

Since P is discrete, this model induces a latent partitioning $\mathcal{C}$ of the data where two data points belong to the same cluster if they are generated from the same mixture component. The partition can be represented as $\mathcal{C} = \{\mathcal{C}_k : k = 1, \cdots, K\}$. Let $\mathbf{x_j} = \{x_n\}_{n \in \mathcal{C}_j}$, the marginal likelihood of the data $\mathbb{D}$ given the partition is,

$$f(\mathbb{D}|\mathcal{C}) = \prod_{j=1}^{K} m(\mathbf{x_j}) = \prod_{j=1}^{K} \int \prod_{n \in \mathcal{C}_j} \phi(x_n|\theta) dP_0(\theta) \tag{3.10}$$

The posterior of the partition which reflects the belief and uncertainty in the clustering given the data, is simply proportional to the prior times the marginal likelihood.

$$p(\mathcal{C}|\mathbb{D}) \propto p(\mathcal{C}) \prod_{j=1}^{K} m(\mathbf{x_j}) \tag{3.11}$$

3.7 Weighted Probabilistic Distance (w-PDQ) Clustering

The basic principle of weighted probabilistic distance clustering is based on the following assumption. We assume for each $x_i \in \mathcal{D}$ and cluster $\mathcal{C}_k$, the probability that x belongs to $\mathcal{C}_k$ given by $p_k(x)$ satisfies,

$$\frac{p_k(x_i)\, d_k(x_i)}{q_k} = D(x_i), \;\; k = 1, \cdots, K, \tag{3.12}$$

where, $d_k(x)$ denotes $d_k(x, c_k)$, the distance of x to center $\mathbf{c_k}$ of the k-th cluster, and q_k is size of the cluster. The **cluster membership probabilities** $\{p_k(x) : k = 1, \cdots, K\}$ of a point x depend only on the **distances** and **cluster sizes**.

$$p(x) = f(d(x), q) \tag{3.13}$$

where $\mathbf{p}(\mathbf{x}) \in \mathbb{R}^K$ is the vector of probabilities $\{p_k(x)\}$, $\mathbf{d}(\mathbf{x})$ is the vector of distances $\{d_k(x)\}$, and $\mathbf{q}$ is the vector of cluster sizes $\{q_k\}$. From the basic principle in (3.12),

$$d_i(x) < d_j(x) \implies p_i(x) > p_j(x), \text{ and}$$
$$q_i > q_j \implies p_i(x) > p_j(x), \forall i, j \in 1, \cdots, K \tag{3.14}$$

The cluster membership is more probable when the data point is closer to the cluster center and larger the cluster. The cluster size q_k is absent in (3.12), if it is not relevant. The weight w_i in LHS (3.12) can be interpreted as an observation with weight w_i which is equivalent to w_i similar observations, each of weight 1, in the same location. The distance $d_k(x_i)$ in (3.12) can be replaced by an increasing function of itself, giving another principle.

3.7.1 Cluster Membership Probabilties

The **cluster membership probabilities**, $p_k(x) := \text{Prob}\{x \in \mathcal{C}_k\}, k \in 1, \cdots, K$ assumed to depend only on the distances $\{d_k(x) : k = 1, \cdots, K\}$ of the point x from the cluster centers and the cluster sizes $\{q_k\}$. From the above principle, and the fact that probabilities add to 1 we get,

Theorem: Given the cluster centers $\{c_1, \cdots, c_k\}$, and distances $\{d_k(x) : k = 1, \cdots, K\}$ of a data point x from the given centers, the membership probabilities of x are,

$$p_k(x) = \left(\prod_{j \neq k} \frac{d_j(x)}{q_j}\right) \left(\sum_{i=1}^{K} \prod_{j \neq i} \frac{d_j(x)}{q_j}\right)^{-1}, \ k = 1, \cdots, K, \tag{3.15}$$

Proof: Using (3.12) we write for i,k

$$p_i(x) = \left(\frac{p_k(x) d_k(x)}{q_k}\right) / \left(\frac{d_i(x)}{q_i}\right) \tag{3.16}$$

Since $\sum_{i=1}^{K} p_i(x) = 1$,

$$p_k(x) \sum_{i=1}^{K} \left(\frac{d_k(x)/q_k}{d_i(x)/q_i}\right) = 1$$

$$p_k(x) = \frac{1}{\sum_{i=1}^{K} \left(\frac{d_k(x)/q_k}{d_i(x)/q_i}\right)} = \frac{\prod_{j \neq k} d_j(x)/q_j}{\sum_{i=1}^{K} \prod_{j \neq i} d_j(x)/q_j} \tag{3.17}$$

The probabilities **do not depend on the weight** of x.

3.7.2 Power Probabilities

To make the cluster membership probabilities better approximate hard assignments, we replace (3.12) by

$$\frac{p_k(x_i)\, d_k^{\nu}(x_i)}{q_k} = D(x_i),\ k = 1, \cdots, K, \tag{3.18}$$

with exponent $\nu \geq 1$, and denote the resulting probabilities and JDF by $p_k^{(\nu)}(x)$ and $D^{(\nu)}(x)$ respectively. These are obtained from (3.15) and (3.24) by replacing every distance d by d^{ν}. The probabilities $p_k^{(\nu)}(x)$ can be computed by raising (3.17) to the power ν and normalizing,

$$p_k^{(\nu)}(x) = \frac{p_k^{\nu}(x)}{\sum\limits_{j=1}^{K} p_j^{\nu}(x)},\ k = 1, \cdots, K. \tag{3.19}$$

Hard assignments can be approximated by the probabilities $\{p_k^{(\nu)}\}$, for sufficiently high ν. Indeed if $d(x, c_k)$ is the unique minimal distance of x from all centers,

$$d(x, c_k) \stackrel{!}{=} \min\{d(x, c_j) :\ j = 1, \cdots, K\},$$

then $p_k(x)$ is the unique maximal cluster–membership probability,

$$p_k(x) \stackrel{!}{=} \max\{p_j(x) :\ j = 1, \cdots, K\},$$

and, by (3.19),

$$\lim_{\nu \to \infty} p_j^{(\nu)}(x) = \begin{cases} 1,\ j = k, \\ 0,\ j \neq k, \end{cases} \tag{3.20}$$

a hard assignment of x to the k^{th} cluster.

3.7.3 Updating the Exponent ν

If the assignment probabilities are the power probabilities (3.19), we update ν incrementally. As typically the case in gradient methods, the iterations (3.40) make big steps at first, approaching their fixed points, then the iterations slow down and movement in each iteration is small. The "fast" iterations are few in number, the number of "slow" iterations is determined by the stopping criterion.

Most progress towards identifying the cluster centers occur in the first few iterations. The slow iterations at the end deal mainly with the assignment problem. With a low value of ν, the distributions may be far from hard assignments, and require rounding to the nearest integer, 0 or 1. High values of the exponent ν produce $\{p_k^{(\nu)}(x)\}$ that are close to hard assignments. For this reason, high values of ν are useful in the slow iterations at the end. In contrast, using high values of ν at the beginning may cause premature convergence to a sub-optimal solution.

This suggests increasing the exponent ν at each iteration. We use a simple update here,

$$\nu^+ = \nu + \Delta \tag{3.21}$$

where $\Delta > 0$ is the increment per iteration. If ν_0 is the initial exponent, the k^{th} exponent is $\nu_0 + k\Delta$.

3.7.4 Joint Distance Function

From the basic principle of weighted probabilistic distance clustering we get,

$$p_k(x) = \frac{D(x)}{d_k(x)/q_k} \tag{3.22}$$

Here, $D(x)$ is a constant and is function of x. Since the probabilities add to 1, from (3.12)–(3.15) it follows that for any x,

$$D(x) = \left(\prod_{j=1}^{K} \frac{d_j(x)}{q_j}\right) \left(\sum_{\ell=1}^{K} \prod_{j\neq\ell} \frac{d_j(x)}{q_j}\right)^{-1}, \tag{3.23}$$

We call this constant $D(x)1$ as **JDF** (Joint Distance Function) at a point x. If the cluster sizes in (3.24)are not relevant we get,

$$\begin{aligned} D(x) &= \frac{1}{\sum_{k=1}^{K} \frac{1}{d_k(x)}} = \frac{\prod_{j=1}^{K} d_j(x)}{\sum_{\ell=1}^{K} \prod_{j\neq\ell} d_j(x)} \\ &= \frac{1}{K} H(d_1(x), d_2(x), \cdots, d_K(x)) \end{aligned} \tag{3.24}$$

where $H(\cdots)$ is the **harmonic mean** of its arguments. Extending to the whole data set, we obtain **JDF of the data set** by,

$$\sum_{i=1}^{N} D(x_i) \tag{3.25}$$

3.7.5 Probabilistic Assignments

We approximate the clustering problem using a probabilistic model (3.4), replacing hard assignments by probabilities, called **probabilistic** (or "soft") **assignment**, and denoted by $\pi_k(x)$, the probability that x is "assigned" to $\mathcal{C}_k$. **Assignment probabilities** include hard assignments, and the probabilities (3.17), (3.19) as special cases.

Probabilistic assignments $\{\pi_k(x) : k \in 1, \cdots, K\}$ are assumed to have the following property: If a point x coincides with center $\mathbf{c_k}$, i.e.,

$$\text{if} \quad d(x, c_k) = 0, \quad \text{then} \begin{cases} \pi_k(x) = 1, \\ \pi_j(x) = 0, \quad j \neq k \end{cases} \tag{3.26}$$

3.7.6 Extremal Principle for Probabilities

Equation (3.12) is the optimality condition of the following extremum problem, with the probabilities $\{p_k(x)\}$ as variables, the distances $\{d_k(x)\}$ and cluster sizes q_k assumed given. At any point x_i,

$$\begin{aligned} \min_{p_1(x_i), \cdots, p_K(x_i)} \quad & \sum_{k=1}^{K} \frac{p_k(x_i)^2 \, d_k(x_i)}{q_k} \\ \text{s.t.} \quad & \sum_{k=1}^{K} p_k(x_i) = 1. \end{aligned} \tag{3.27}$$

The Lagrangian of (3.27) is,

$$L(p, \lambda) = \sum_{k=1}^{K} \frac{p_k(x_i)^2 \, d_k(x_i)}{q_k} - \lambda \left(\sum_{k=1}^{K} p_k(x_i) - 1 \right) \tag{3.28}$$

Differentiating the Lagrangian w.r.t. $p_k(x_i)$ and zeroing the partial derivatives,

$$2 \frac{p_k(x_i) \, d_k(x_i)}{q_k} = \lambda, \ \forall k \tag{3.29}$$

which gives (3.12). We can therefore write (3.27) as

$$\begin{aligned}\min_{p_1(x_i),\cdots,p_K(x_i)} \quad & \sum_{k=1}^{K} \frac{p_k(x_i)^2\, d_k(x_i)}{q_k} \\ &= D(x_i) \sum_{k=1}^{K} p_k(x_i), \quad \text{using (3.12)}, \\ &= D(x_i)\end{aligned}$$

Therefore, problem (3.27) is equivalent to

$$\min\ D(x_i). \tag{3.30}$$

3.7.7 Extremal Principle for Cluster Sizes

If the cluster sizes need to be determined, they are the minimizers of the following problem, for the given distances $d_k(x_i, c_k)$ and probabilities $\pi_k(x_i)$, $i = 1, \cdots, N;\ k = 1, \cdots, K$.

$$\min_{q_1,\cdots,q_K} \quad \sum_{k=1}^{K} \sum_{i=1}^{N} w_i\, \frac{\pi_k(x_i)\, d_k(x_i)}{q_k} \tag{3.31}$$

$$\text{s.t.} \sum_{k=1}^{K} q_k = W, \tag{3.32}$$

The Lagrangian of (3.31) is

$$L(q, \lambda) = \sum_{k=1}^{K} \sum_{i=1}^{N} w_i\, \frac{\pi_k(x_i)\, d_k(x_i)}{q_k} + \lambda \left(\sum_{k=1}^{K} q_k - W \right) \tag{3.33}$$

Differentiating the Lagrangian w.r.t. q_k and zeroing the partial derivatives,

$$\begin{aligned}\sum_{i=1}^{N} w_i\, \frac{\pi_k(x_i)\, d_k(x_i)}{q_k^2} &= \lambda,\ \forall k \qquad (3.34)\\ \therefore\ q_k &= \sqrt{\sum_{i=1}^{N} \frac{w_i}{\lambda}\, \pi_k(x_i)\, d_k(x_i)}\end{aligned}$$

and finally, by (3.32),

$$q_k = \frac{\sqrt{\sum\limits_{i=1}^{N} w_i\, \pi_k(x_i)\, d_k(x_i)}}{\sum\limits_{j=1}^{K} \sqrt{\sum\limits_{i=1}^{N} w_i\, \pi_j(x_i)\, d_j(x_i)}}\, W \tag{3.35}$$

3.7.8 Updating Cluster Centers

As mentioned before, the distance of point x from the cluster $\mathcal{C}_k$ is its distance from the cluster center $\mathbf{c_k}$ which is denoted as $d_k(x, c)$,

$$d_k(x, c_k) = \sqrt{\langle x - c_k, \Sigma_k^{-1}\,(x - c_k)\rangle},\ k = 1, \cdots, K, \tag{3.36}$$

For simplicity we refer to it as $d_k(x)$. The gradient of (3.36) w.r.t. c_k is

$$\nabla_{c_k}\, d_k(x, c_k) = -\Sigma_k^{-1}\,\frac{x - c_k}{d_k(x, c_k)},\ k = 1, \cdots, K. \tag{3.37}$$

Given the distances $d_k(x_i, c_k)$ and assignment probabilities $\pi_k(x_i)$, the centers are the minimizers of the function,

$$\min_{c_1, \cdots, c_k} \sum_{k=1}^{K} \sum_{i=1}^{N} w_i\,\frac{\pi_k(x_i)\, d_k(x_i, c_k)}{q_k}, \tag{3.38}$$

which is the sum of the objectives (3.27), over all x_i. The gradient of the objective (3.38) w.r.t. c_k is, by (3.37),

$$\nabla_{c_k} \sum_{i=1}^{N} w_i\,\frac{\pi_k(x_i)\, d_k(x_i, c_k)}{q_k} = -\Sigma_k^{-1} \sum_{i=1}^{N} w_i\,\frac{\pi_k(x_i)}{q_k\, d_k(x_i, c_k)}\,(x_i - c_k),\ k = 1, \cdots, K. \tag{3.39}$$

Equating the gradient to zero, and simplifying, we get the new center $(c_k)^+$ as a convex combination of the N data points,

$$(c_k)^+ = \sum_{i=1}^{N} \lambda_{ki}\, x_i,\ k = 1, \cdots, K, \tag{3.40}$$

where the weights λ_{ki} are

$$\lambda_{ki} = \frac{w_i\,\dfrac{\pi_k(x_i)}{d_k(x_i, c_k)}}{\sum\limits_{j=1}^{N} w_j\,\dfrac{\pi_k(x_j)}{d_k(x_j, c_k)}},\ k = 1, \cdots, K;\ i = 1, \cdots, N. \tag{3.41}$$

The weights λ_{ki} in (3.41) depend on the old centers c_k, and therefore (3.40)–(3.41) give the new centers $(c_k)^+$ interms of the old ones. The λ_{ki} depend on the observation weights w_i. Suppose, x_i is an outlier, it is far from the centers, and therefore all its weights λ_{ki} are very small. From (3.40), we can say that the centers $(c_k)^+$ are **not sensitive to outliers**.

3.7.9 Updating Covariance Matrix

In case of Mahalanobis distance, $d(x, c_k) = \sqrt{(x - c_k)^T \Sigma_k^{-1} (x - c_k)}$, the covariance matrix Σ_k of the k^{th}–cluster is updated at each iteration by,

$$\Sigma_k = \frac{\sum_{i=1}^{N} \lambda_{ki}(x_i - c_k)(x_i - c_k)^T}{\sum_{i=1}^{N} \lambda_{ki}}, \tag{3.42}$$

3.7.10 Cluster Uncertainty

The JDF has the dimension of distance. Normalizing it we get a dimensionless function,

$$E(x) = KD(x)/\big(\prod_{j=1}^{K} d_j(x)\big)^{1/K} \tag{3.43}$$

with 0/0 interpreted as zero. $E(x)$ is the harmonic mean of the distances divided by their geometric mean. It follows that $0 \leq E(x) \leq 1$, with $E(x) = 0$ if any $d_j(x) = 0$, i.e., if x is the cluster center, and $E(x) = 1$ if and only if the distances $d_j(x)$ are all equal.

$E(x)$ can be written using as the geometric mean of the cluster membership probabilities (up to a constant),

$$E(x) = K\big(\prod_{j=1}^{K} p_j(x)\big)^{1/K} \tag{3.44}$$

The function $E(x)$ represents the uncertainty of classifying the point x. We call $E(x)$ as **cluster uncertainty measure**. The cluster uncertainty of the data set is defined as

$$E(\mathcal{D}) := \frac{1}{N} \sum_{i=1}^{N} E(x_i) \tag{3.45}$$

$E(\mathcal{D})$ is a monotone decreasing function of K. It decreases from $E(\mathcal{D} = 1)$ for $K = 1$ to $E(\mathcal{D}) = 0$ for $K = N$, the trivial case where every data point is in a separate cluster. An intrinsic criterion for determining the optimal k is provided by the rate of decrease of $E(\mathcal{D})$.

Initialization:	given data set $\mathcal{D}$ with N points,
	each with weight w_i and $\sum_{i=1}^{N} w_i = W$,
	$\mathcal{K}$ the number of clusters
	any $\mathcal{K}$ centers $\{c_k : k = 1, \cdots, K\}$,
	any $\mathcal{K}$ cluster sizes $\{q_k > 0 : \sum_{k=1}^{K} q_k = W\}$,
	$\epsilon > 0$
Iteration:	
Step 1	**compute** distances $\{d_k(x, c_k) : k = 1, \cdots, K\}$ for all $x \in \mathcal{D}$
Step 2	**update** the cluster sizes $\{q_k^+ : k = 1, \cdots, K\}$ (using (3.35))
Step 3	**update** the centers $\{c_k^+ : k = 1, \cdots, K\}$ (using (3.40)–(3.41))
Step 4	**update** the covariance matrix $\{\Sigma_k^+ : k = 1, \cdots, K\}$ (using (3.42))
Step 5	**if** $\sum_{k=1}^{K} \|c_k^+ - c_k\| < \epsilon$ **stop**
	return to Step 1

3.7.11 Weighted PDQ-Algorithm

A schematic description of the algorithm is provided below.

Notes:

(a) If the cluster sizes $\{q_k\}$ are known, they are used as the initial estimates and are not updated thereafter, in other words Step 2 is absent.

(d) The computations stop (in Step 4) when the centers stop moving, at which point the cluster membership probabilities may be computed by (3.15). These probabilities are used afterwards for classifying the data.

(e) Step 3 of the algorithm is a generalization of the Weiszfeld iteration, to several centers. As in the classical case, to establish convergence it is necessary to modify the gradient in question, if a center coincides with one of the data points, we apply a mechanical solution, see [Ref].

3.8 Binder's Loss

A loss function $L(\mathcal{C}, \hat{\mathcal{C}})$ measures the loss of estimating the true clustering $\mathcal{C}$ with $\hat{\mathcal{C}}$. Since, the true clustering is unknown, the loss is averaged across all possible true clusterings, where the loss associated to each potential true clustering is weighted by its posterior probability. The point estimate $\mathcal{C}^*$ corresponds to the estimate that minimizes

the posterior expected loss,

$$\mathcal{C}^* = \underset{\hat{\mathcal{C}}}{\operatorname{argmin}}\ \mathbb{E}[L(\mathcal{C},\hat{\mathcal{C}})|y_{1:N}] = \underset{\hat{\mathcal{C}}}{\operatorname{argmin}}\ \sum_{\mathcal{C}} L(\mathcal{C},\hat{\mathcal{C}})p(\mathcal{C}|y_{1:N}) \tag{3.46}$$

Let $n_{ij} = |\mathcal{C}_i \cap \hat{\mathcal{C}}_j|$ be the cardinality of the intersection between $\mathcal{C}_i$, the set of data points in cluster i under $\mathcal{C}$, and $\hat{\mathcal{C}}_j$, the set of data points in cluster j under $\hat{\mathcal{C}}$, where $i = 1, \cdots, k_N$, and $j = 1, \cdots, \hat{k}_N$. the notation k_N, $\hat{k}_N$ denotes the number of clusters in $\mathcal{C}$ and $\hat{\mathcal{C}}$ respectively. Binder's loss [20]is a quadratic function of the counts n_{ij}, which for all possible pairs of observations, penalizes the two errors: (i) allocating two observations to different clusters when they should be in the same cluster; and (ii) allocating them to the same cluster when they should be in different clusters. It is written as:

$$B(\mathcal{C},\hat{\mathcal{C}}) = \sum_{n<n'} l_1 \mathbf{1}(\mathcal{C}_n = \mathcal{C}_{n'})\mathbf{1}(\hat{\mathcal{C}}_n \neq \hat{\mathcal{C}}_{n'}) + l_2 \mathbf{1}(\mathcal{C}_n \neq \mathcal{C}_{n'})\mathbf{1}(\hat{\mathcal{C}}_n = \hat{\mathcal{C}}_{n'}) \tag{3.47}$$

If the two type of errors are penalized equally, $l_1 = l_2 = 1$, then

$$B(\mathcal{C},\hat{\mathcal{C}}) = \frac{1}{2}\Big(\sum_{i=1}^{k_N} n_{i+}^2 + \sum_{j=1}^{\hat{k}_N} n_{+j}^2 - 2\sum_{i=1}^{k_N}\sum_{j=1}^{\hat{k}_N} n_{ij}^2\Big), \tag{3.48}$$

where $n_{i+} = \sum_j n_{ij}$ and $n_{+j} = \sum_i n_{ij}$. Under Binder's loss with $l_1 = l_2$, the optimal partition $\mathcal{C}^*$ is the partition $\mathcal{C}$ which minimizes

$$\sum_{n<n'} |\mathbf{1}(\mathcal{C}_n = \mathcal{C}_{n'}) - p_{nn'}^2| \tag{3.49}$$

or equivalently, the partition c which minimizes

$$\sum_{n<n'} (\mathbf{1}(\mathcal{C}_n = \mathcal{C}_{n'}) - p_{nn'}^2)^2 \tag{3.50}$$

where $p_{nn'} = P(\mathcal{C}_n = \mathcal{C}_{n'}|y_{1:N})$ is the posterior probability that two observations are clustered together.

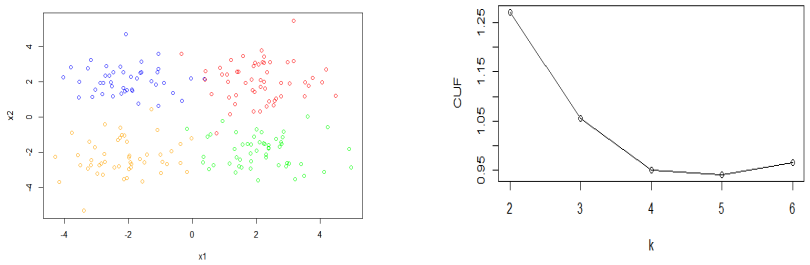
Binder's loss counts the total number of disagreements (D) in the $\binom{N}{2}$ possible pairs of observations. The Rand index $R(\mathcal{C},\hat{\mathcal{C}})$, which is a widely used cluster comparison criterion measures the number of agreements (A) in all possible pairs. since $D + A = \binom{N}{2}$, Binder's loss and rand Index are related as follows:

$$B(\mathcal{C},\hat{\mathcal{C}}) = (1 - R(\mathcal{C},\hat{\mathcal{C}}))\binom{N}{2} \tag{3.51}$$

3.9 Cluster Validation

A fundamental step for any unsupervised algorithm is to determine the "right" number of clusters into which the data may be clustered. In dichotomous situation the answer $K = 2$ is obvious, but in general the answer lies between two extremes $K = 1$ (one cluster fits all) and $K = N$ (each point is a cluster). **Elbow Method** is one of the most popular methods to determine this optimal value of K.

Here, we apply a similar approach on the dataset in example 4, to determine the value of K based on the CUF mentioned in. We iterate over a range of values of K and calculate the CUF for each value of K. As seen below**??**, the value of $E_K(\mathcal{D})$ decreases monotonically with K. The decrease of the uncertainty $E_K(\mathcal{D})$ is precipitous until reaching the "right" value of K and thereafter becomes almost flat.



(a) Data (b) Cluster validation

Figure 3.1: Finding optimal value of "k"

3.10 Empirical Evaluation

We evaluated our method using both simulated and real-world data sets. The purpose of using simulated datasets was to illustrate how well the algorithms could recover the parameters of underlying distributions. We constructed several examples with data simulated from different multivariate normal distributions.

Since our proposed method requires a random initialization, it can sometime lead to unstable solution. To avoid this, we apply multiple initial configurations. For determining the optimal partition, we used **Binder's loss** that is commonly used in Bayesian

clustering methods. The optimal partition thus obtained corresponds to the estimate which minimizes the posterior expected Binder's loss.

In order to compare different clustering methods, we used **KL-Divergence** and **adjusted Rand index** (ARI) [21]. KL divergence measures the distance between actual and estimated distributions of the clusters. A smaller value of KL Divergence indicates better approximation of the true clusters. Rand index compares predicted classifications with true classes. The ARI corrects Rand index for chance, its expected value under random classification is 0, and it takes a value of 1 when there is perfect class agreement.

All computations were performed using R and the CRAN packages *mclust*, *mcclust*, *mcclust.ext* [22]. The implementation of our w-PDQ algorithm is included in a newly developed package called *PDQClustering* made available on github.

3.10.1 Example 1

This dataset is constructed to illustrate the application of w-PDQ clustering on weighted data. Here (Figure 3.2), we generated containing 200 data points from two different normal distributions and assigned random weights to these points. Parameters of the distributions are as follows:

$\mu_1 = (0, 0), \Sigma_1 = \begin{pmatrix} 0.1 & 0 \\ 0 & 1 \end{pmatrix}$, and

$\mu_2 = (3, 0), \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0.1 \end{pmatrix}$

We initialized the w-PDQ algorithm with 100 different initial configurations, the value of ϵ was set to 0.0001 and the value of ν was set to 0.003. Table 3.1 shows the true and estimated values of the cluster parameters. Binder's loss for the estimated clustering was 0.26. Adjusted Rand index was 0.95. KL divergence for cluster 1 and 2 were 0.02 and 0.1 respectively.

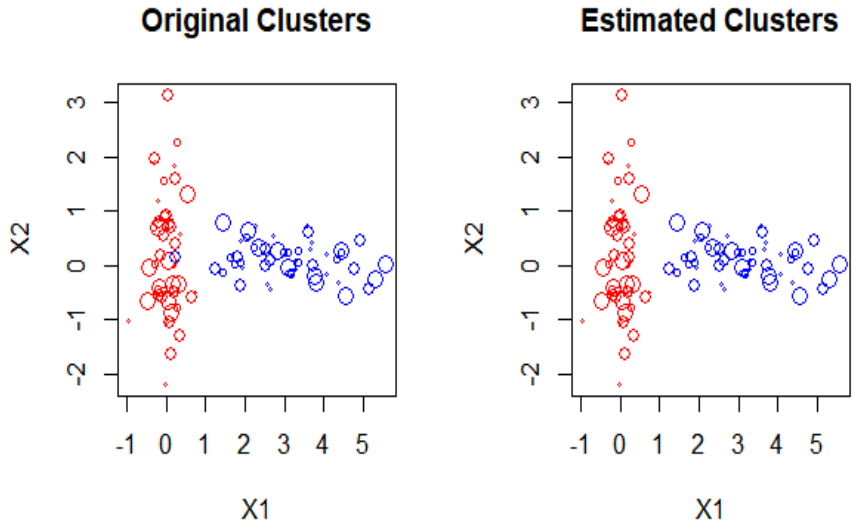


Figure 3.2: Gaussian mixtures with random weights

Table 3.1: True and estimated values of cluster parameters

	Parameters	True	Estimated
Cluster 1	μ_1	(0,0)	(0,0.1)
	Σ_1	(0.1,0,0,1)	(0.1,0,0,1.2)
Cluster 2	μ_2	(3,0)	(3.1,0.1)
	Σ_2	(1,0,0,0.1)	(1.2,0,0,0.1)

3.10.2 Example 2

This is an example where the level of differentiation among the true clusters is significantly low and the clusters tend to overlap. The dataset (Figure 3.3) containing 300 data points was simulated from three different multivariate normal distributions having different covariances.

$\mu_1 = (4,4), \Sigma_1 = \begin{pmatrix} 0.25 & 0.21 \\ 0.21 & 0.25 \end{pmatrix}$, $\mu_2 = (5,5), \Sigma_2 = \begin{pmatrix} 0.25 & -0.21 \\ -0.21 & 0.25 \end{pmatrix}$, and

$\mu_3 = (6.5,5), \Sigma_3 = \begin{pmatrix} 0.25 & 0.21 \\ 0.21 & 0.25 \end{pmatrix}$

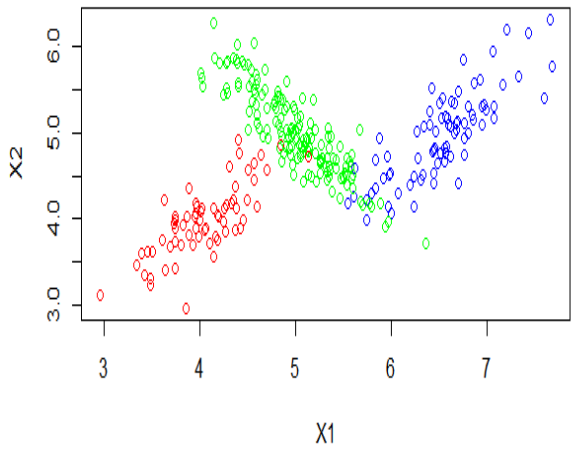
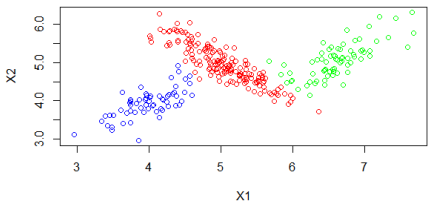
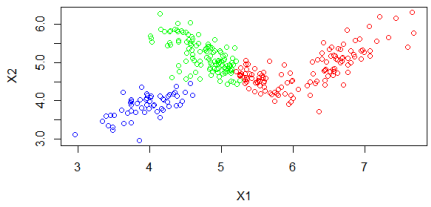


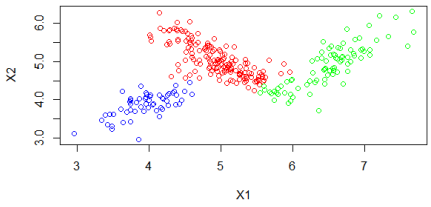
Figure 3.3: Overlapping Gaussians of equal sizes



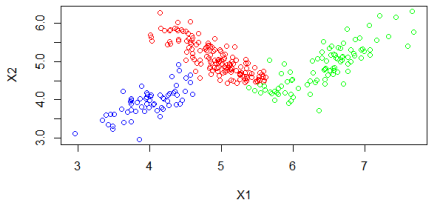
(a) $\nu = 0.2$



(b) $\nu = 0.3$



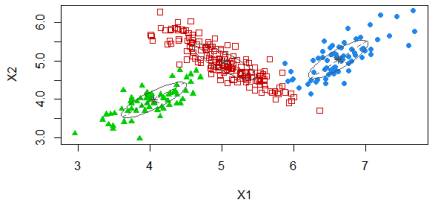
(c) $\nu = 0.37$



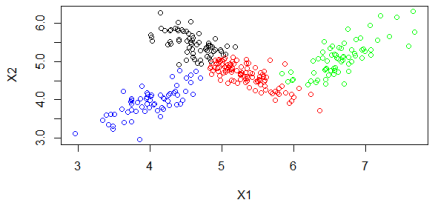
(d) $\nu = 0.38$

Figure 3.4: Weighted PDQ clusters for different power probabilities

In this example, a major challenge was to deal with data points that lie in overlapping regions of the clusters. These data points had nearly equal membership probabilities for each cluster. In order to make these probabilities much sharper and eventually



(a) EM Method



(b) Bayes Method

Figure 3.5: Results of EM and Bayesian clustering methods for overlapping clusters

Table 3.2: True and estimated values of cluster parameters

	Parameters	True	Estimated (w-PDQ)	Estimated (EM)
Cluster 1	μ_1	(4.0,4.0)	(4.0,3.93)	(4.05,3.98)
	Σ_1	(0.25,0.21,0.21,0.25)	(0.13,0.1,0.1,0.15)	(0.20,0.17,0.17,0.23)
Cluster 2	μ_2	(5.0,5.0)	(5.03,4.95)	(5.03,4.97)
	Σ_2	(0.25,-0.21,-0.21,0.25)	(0.21,-0.20,-0.20,0.25)	(0.2,-0.17,-0.17,0.22)
Cluster 3	μ_3	(6.5,5.0)	(6.62,5.06)	(6.62,5.04)
	Σ_3	(0.25,0.21,0.21,0.25)	(0.17,0.14,0.14,0.21)	(0.17,0.17,0.17,0.25)

to make hard assignments, we used the power probabilities mentioned above (3.19). (Figure 3.4) illustrates the clusters obtained for different values of ν. Number of initial configurations was set to 100 and the value of ϵ was set to 0.0001. We also applied model based clustering based on the EM method on this data (Figure 3.5).

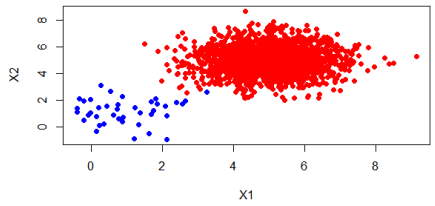
The ARIs were 0.91, 0.86 , 0.88, 0.88 respectively. ARI for the EM method was 0.86. KL divergences for the W-PDQ method were 0.18, 0.03, 0.01 for clusters 1 , 2 , and 3 respectively. For EM method, the KL divergences were 0.17, 0.02, 0.05 for clusters 1, 2, and 3 respectively. However, it was interesting to see the results of Bayesian method that detected four clusters instead of three based on minimum Binder's loss.

3.10.3 Example 3

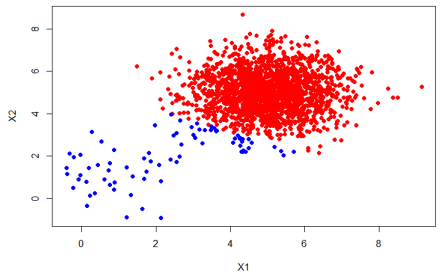
This dataset (Figure **??**) consists of two clusters of unequal sizes. The cluster on the left consists of 50 data points and the one on right contains 2000 data points. The points are generated from Normal distributions with parameters:

$$\mu_1 = (5,5), \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{and}$$

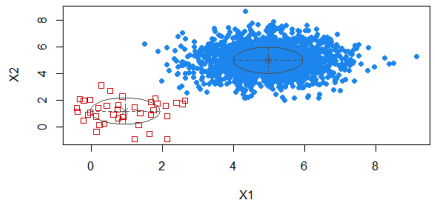
$$\mu_2 = (1,1), \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



(a) Original data



(b) w-PDQ Method



(c) EM Method

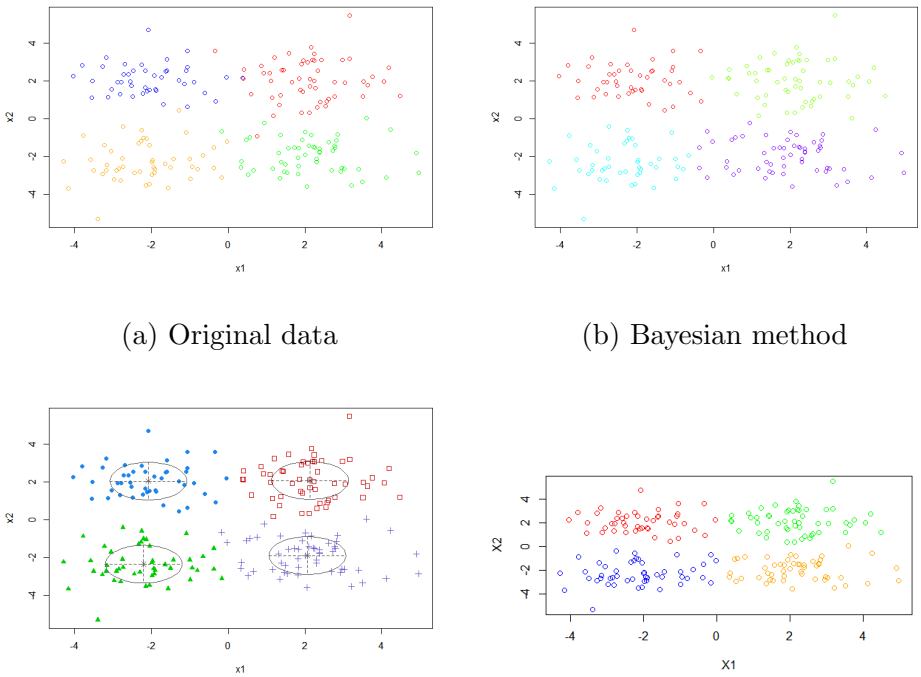
Figure 3.6: Weighted PDQ clusters for different power probabilities

Here, the clusters are highly imbalanced. The challenge is how well the clustering algorithms can estimate the true classes. Results of our experiments showed that deterministic and Bayesian algorithms either tries to equalize the cluster sizes or end up estimating all in one cluster. Both w-PDQ and EM methods could closely approximate

the true clusters. The ARI for EM method was 0.98 and that of w-PDQ method was 0.96. KL divergences for the EM method was 0.02, 0.01 and that for w-PDQ method was 0.02 and 0.018.

3.10.4 Example 4

This is a simulated dataset containing 200 points from a mixture of four normals. True clusters are located at $(+/-2, +/-2)$ with a standard deviation of 1. The dataset is provided in the package *mcclust*. We applied the three methods – EM, Bayesian, and w-PDQ on the data.



(a) Original data

(b) Bayesian method

(c) EM method

(d) w-PDQ method

Figure 3.7: Weighted PDQ clusters for different power probabilities

Results are shown in Figure 3.7. We obtained an ARI of 0.86 for Bayesian method, 0.93 for EM method, and 0.93 for w-PDQ method. The Kl divergences for clusters 1, 2, 3, and 4 were, (i) EM method: 0.01, 0.008, 0.01, 0.005; (ii) w-PDQ method: 0.007,0.01,0.012,0.01; (iii) Bayesian method: 0.02, 0.12, 0.1, 0.17.

3.11 Capacitated Clustering Problem

The capacitated clustering problem (CCP), is a well-known NP-hard combinatorial optimization problem that partitions a group of $\mathcal{N}$ items into K clusters by imposing constraint on cluster sizes. This class of problems are also referred to as capacitated facility location problem. The recent work on facility location problems are either based on linear programming (LP) or local search-based algorithms. Mostly, they asume hard assignments as opposed to the probailistic assignment to facilities. See[23], where the authors have used standard LP relaxation to approximate the optimal solution of the NP hard problem. In order to solve the CCP we have adopted similar objective function, but applied probabilistic decoposition instead.

Given the dataset $\mathcal{D} := \{(x_i, w_i) : \ x_i \in \mathbb{R}^n, \ w_i > 0, \ i = 1, \cdots, N\}$, containing $\mathcal{N}$ points, our objective is to partition the data set $\mathcal{D}$ into K clusters $\{\mathcal{C}_k : k = 1, \cdots, K\}$ where each cluster has limited capacity $\mathcal{Q} = \{q_1, q_2, \cdots, q_k\}$. This means that there are constraints on the cluster sizes,

$$\sum_{x_i \in \mathbb{C}_k} w_i = q_k, \ k = 1, \cdots, K, \tag{3.52}$$

where,

$$\sum_{k=1}^{K} q_k = W \tag{3.53}$$

Since the cluster sizes are given and remain constant, the approach defined in the previous sections won't work. We formulate the problem as follows:

$$\min \quad \sum_{k=1}^{K} y_k + \sum_{i=1}^{N} \sum_{k=1}^{K} w_i d_k(x_i) \pi_k(x_i) \tag{3.54}$$

$$\text{s.t.} \quad \sum_{k} \pi_k(x_i) = 1, \quad \forall i \tag{3.55}$$

$$\pi_k(x_i) \leq y_k, \quad \forall i, k \tag{3.56}$$

$$\sum_{i=1}^{N} w_i \, \pi_k(x_i) \leq q_k y_k, \quad \forall k \tag{3.57}$$

$$y_k \leq 1, \quad \forall k \tag{3.58}$$

$$\pi_k(x_i), y_k \geq 0, \quad \forall i, k \tag{3.59}$$

Variable y_i indicates whether the cluster is available and $\pi_k(x_i)$ indicates the probability of point x_i being assigned to the k^{th} cluster. The first constraint says that for a given point, the cluster membership probabilities should add up to one. The second constraint says that if point i is assigned to cluster j then it must be available, and constraint 3 indicates that at most q_k demand may be assigned to cluster k.

Solving for a given point x, the object function 3.57 simplifies to,

$$\min \quad \sum_k v_i + \sum_k w d_k(x)\pi_k(x) \tag{3.60}$$

$$\text{s.t.} \quad \sum_k w\pi_k(x) = w \tag{3.61}$$

$$w\pi_k(x) \leq q_k v_k, \quad \forall k \tag{3.62}$$

$$v_k \leq 1, \quad \forall k \tag{3.63}$$

$$\pi_k(x), v_k \geq 0, \quad \forall k \tag{3.64}$$

Here w is the total demand (weight) of x, $w\pi_k(x)$ is the total demand assigned to cluster k, and v_i indicates if the cluster k is available. At any time the cluster can be fractionally available that is $0 < v_k < 1$. From constraint (56), we can write,

$$v_k = \frac{w\pi_k(x)}{q_k} \tag{3.65}$$

Now, substituting the variable v_i in 3.60 we get,

$$\min \sum_k \left(\frac{1}{q_k} + d_k(x)\right) w\pi_k(x) \tag{3.66}$$

and substitute the constraints (3.62) and (3.63) by $w\pi_k(x) \leq q_k$ for each k. clearly this is equivalent to the other formulation. To enforce the cluster size constraints we take the assignment probabilities as,

$$\pi_k(x) = \frac{\dfrac{q_k}{1 + q_k d_k(x)}}{\sum\limits_{j=1}^{K} \dfrac{q_j}{1 + q_j d_j(x)}} \tag{3.67}$$

We compute the centers as a convex combination of the $\mathcal{N}$ data points,

$$(c_k)^+ = \sum_{i=1}^{N} \lambda_{ki} x_i \quad k = 1, \cdots, K \tag{3.68}$$

where the weights λ_{ki} are

$$\lambda_{ki} = \frac{w_i \pi_k(x_i)}{q_k \sum_{j=1}^{N} w_j \pi_k(x_j)}, \; k = 1, \cdots, K; \; i = 1, \cdots, N \tag{3.69}$$

This probabilistic decomposition of the capacitated problem ensures that the total capacity of the clusters are fully utilized. The idea of the probabilistic decomposition is that every point belongs to every cluster with a certain probability. This allows the demand of each point to be distributed among the different cluster according to the membership probabilities. From (3.68) and (3.69) it is expected that the cluster centers will be pulled towards the points with higher fractional demand.

3.12 Determining the Spatial Clusters of COVID-19 Cases

At the time of this analysis, there were more than 1,50,000 confirmed cases of COVID-19 in the United States. New York State was the epicenter of the outbreak. We have gathered seven days of data from the daily reports of COVID-19 cases made available by Johns Hopkins University Center for Systems Science and Engineering. For our analysis we just selected the daily records for New York State. The data consists of daily counts for the 62 counties in New York State.

The goal was to find a given number of clusters, where each cluster was assigned a fixed capacity. The capacities here indicate the available hospital beds and the cluster centers could be the location of these hospitals. This data was aggregated to the level of counties with 62 data points (counties), each assigned a weight that represent the count of confirmed cases in that county.

We specified the number of clusters to be 5, and set the capacities to be of proportions 0.2, 0.3, 0.2, 0.2, 0.1 respectively. There were 59,648 confirmed cases. Therefore, the actual capacities rounded to the next integer were 11930, 17895, 11930, 11930 , 5965. We used Euclidean distance for this problem, since our data were spatial coordinates. Unlike, hard capacitated facility problems, which assigns an observation to only one cluster our method perform probabilistic decomposition, and thus allow the demand of an observation to be distributed across different clusters proportionately

with its membership probabilities.

The probabilistic cluster assignments by our methods resulted in cluster sizes 11930, 17895, 11929, 11930, and 5965. The proportion of confirmed cases was highly skewed with long tailed distribution . Most of the confirmed cases were reported in neighboring counties. As such the centers were greatly pulled towards these hot spots.

3.13 Discussion

The results indicate that overall weighted probabilistic distance clustering compares favorably to the other methods. In example 1, we illustrated how well the proposed algorithm could approximate the true clusters when applied on weighted data. Both KL divergences and ARI indicated that the estimated clusters were in fact very close to the true clusters. Example 2 was a special case when there is some amount of correlation in the data and also the clusters tend to overlap. Results showed that EM and weighted PDQ methods could detect the clusters with very high accuracy. Bayesian method failed to identify the clusters in this case. In example 3, where the dataset consisted imbalanced clusters, both w-PDQ and EM methods could closely approximate clusters. The ARIs for both these methods were very close to one. Similar favorable results were obtained in example 4.

In the capacitated clustering problem, our results verified the assumptions and constraints specified. From the data we observe counties such as New York City, Westchester, Nassau, Suffolk, Rockland, and Orange were hard hit by the outbreak of COVID-19 pandemic. As expected the cluster centers were largely pulled towards these dense hotspots. To validate our assumptions, we also applied the method on the same dataset but assigning each counties somewhat comparable number of cases. In this case we observed the centers to be uniformly distributed across the entire state. Our future research would focus on expanding this work on much larger dataset, including the entire tri-state region as more data are available.

When compared to the EM algorithm, w-PDQ algorithm is computationally less expensive. EM algorithm is based on maximum likelihood, and depends on the density

functions in the mixture. Weighted PDQ algorithm makes no assumptions about the densities and avoids the overhead of computing complex density functions. On the other hand each EM iteration requires $K \times N$ function evaluations to evaluate the density functions, where K is the number of components in the mixture. Because, EM iterations are computationally expensive, it is common to use another method, e.g., K-means, as pre-processor, to get closer to the centers before starting EM. The proposed method require no such switch and works well even with cold start.

In Bayesian framework, an important factor is the choice of priors. A great deal of work is required for coming up with a prior that's well reasoned and for appropriately summarizing the prior. For models involving many variables and when the data cannot readily be thrown onto a cluster, the Bayesian method could be prohibitively intensive. To this end, weighted PDQ algorithm is computationally more efficient and require no such priors. Although Bayesian methods also perform soft-clustering and allow overlapping clusters, an important limitation is, as the level of differentiation among the true clusters decreases, the performance of Bayesian clustering methods also decrease. Weighted PDQ algorithm uses the power probabilities to address such situations. Power probabilities tend to push the probabilities to the extreme and avoid getting equal probabilities.

Lastly, we also point out that for many clustering techniques, the objective function is not convex or quasi-convex, and may have other stationary points. In w-PDQ clustering however, the JDF is a montonically decreasing function that guarantees convergence to a minimum, though not necessarily a global minimum.

3.14 Conclusion

In this paper, we proposed a novel approach called weighted probabilistic distance clustering (w-PDQ) for weighted-data. The method is a based on joint distance function, which is harmonic mean of distances between a point and the different clusters. For a given point, the probability of belonging to a cluster depends on its distance to the center and size of the cluster. Unlike other probabilistic clustering methods, w-PDQ

clustering is non-parametric and model free. This method is suited for large datasets due to its ability to handle weighted-data. In order to evaluate the proposed method, we performed several experiments on simulated data and compared its performance with other methods. Overall, the results indicate our method performs favorably to other methods. Our future work will focus on extending the method to an incremental setting such that it could also be applied on streaming data. We also plan to publish our R package on CRAN repository thus making it widely available for use by other researchers.

References

[1] C. Iyigun and A. Ben-Israel, *"Probabilistic distance clustering adjusted for cluster size,"* Probability in the Engineering and Informational Sciences, vol. 22, no. 4, pp. 603–621, 2008.

[2] C. Iyigun, *"Probabilistic distance clustering,"* Wiley Encyclopedia of Operations Research and Management Science, 2010.

[3] S Sharma, *Chapter 7 clustering algorithms. applied multivariate techniques,* 1996.

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

[5] B. Everitt and S. L. Landau, *"M. 2001. cluster analysis,"* Arnold, London, 2001.

[6] A. Scott and M. Symons, *"Clustering methods based on maximum likelihood,"* Biometrics, vol. 27, no. 2, 1971.

[7] R. Duda, P. Hart, and D. Stork, *"Pattern classification. 2nd edn wiley,"* New York, vol. 153, 2000.

[8] G. J. McLachlan and K. E. Basford, *Mixture models: Inference and applications to clustering.* M. Dekker New York, 1988, vol. 38.

[9] D. Peel and G. J. McLachlan, *"Robust mixture modelling using the t distribution,"* Statistics and computing, vol. 10, no. 4, pp. 339–348, 2000.

[10] C. Fraley and A. E. Raftery, *"Model-based clustering, discriminant analysis, and density estimation,"* Journal of the American statistical Association, vol. 97, no. 458, pp. 611–631, 2002.

[11] P. D. McNicholas, *"Model-based clustering,"* Journal of Classification, vol. 33, no. 3, pp. 331–373, 2016.

[12] H Stein, R Schwarting, G Niedobitek, and F Dallenbach, *"Cluster report: Cdw70,"* Oxford University Press, New York, 1989.

[13] A. Gelman and G. King, *"Estimating the electoral consequences of legislative redistricting,"* Journal of the American statistical Association, vol. 85, no. 410, pp. 274–282, 1990

[14] I. Verdinelli and L. Wasserman, *"Bayesian analysis of outlier problems using the gibbs sampler,"* Statistics and Computing, vol. 1, no. 2, pp. 105–117, 1991.

[15] M. D. Escobar and M. West, *"Bayesian density estimation and inference using mixtures,"* Journal of the american statistical association, vol. 90, no. 430, pp. 577–588, 1995.

[16] H. Bensmail, G. Celeux, A. E. Raftery, and C. P. Robert, *"Inference in model-based cluster analysis,"* Statistics and Computing, vol. 7, no. 1, pp. 1–10, 1997.

[17] R. J. Hathaway and Y. Hu, *"Density-weighted fuzzy c-means clustering,"* IEEE Transactions on Fuzzy Systems, vol. 17, no. 1, pp. 243–252, 2008.

[18] D. Ghosh, J. Cabrera, T. N. Adam, P. Levounis, and N. R. Adam, *"Comorbidity patterns and its impact on health outcomes: Two-way clustering analysis,"* IEEE Transactions on Big Data, 2016.

[19] F. A. Quintana, *"A predictive view of bayesian clustering,"* Journal of Statistical Planning and Inference, vol. 136, no. 8, pp. 2407–2429, 2006.

[20] D. A. Binder, *"Bayesian cluster analysis,"* Biometrika, vol. 65, no. 1, pp. 31–38, 1978.

[21] D. Steinley, *"Properties of the hubert-arable adjusted rand index.,"* Psychological methods, vol. 9, no. 3, p. 386, 2004.

[22] S. Wade and M. S. Wade, *"Package 'mcclust. ext',"* Journal of Computational and Graphical Statistics, vol. 16, pp. 526–558, 2015.

[23] R. Levi, D. B. Shmoys, and C. Swamy, *"Lp-based approximation algorithms for capacitated facility location,"* in International Conference on Integer Programming and Combinatorial Optimization, Springer, 2004, pp. 206–218.

[24] M. Lavine and M. West, *"A bayesian method for classification and discrimination,"* Canadian Journal of Statistics, vol. 20, no. 4, pp. 451–461, 1992.

[25] J. Diebolt and C. P. Robert, *"Estimation of finite mixture distributions through bayesian sampling,"* Journal of the Royal Statistical Society: Series B (Methodological), vol. 56, no. 2, pp. 363–375, 1994.

[26] M. Evans, I. Guttman, and I. Olkin, *"Numerical aspects in estimating the parameters of a mixture of normal distributions,"* Journal of Computational and Graphical Statistics, vol. 1, no. 4, pp. 351–365, 1992.

Part III:

Classification on High-Dimensional Data

Chapter 4

Enriched Random Forest for High Dimensional Genomic Data

A paper sumitted to IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2019

Debopriya Ghosh, Javier Cabrera

Abstract

Ensemble methods such as random forest works well on high-dimensional datasets. However, when the number of features is extremely large compared to the number of samples and the percentage of truly informative feature is very small, performance of traditional random forest decline significantly. To this end, we develop a novel approach that enhance the performance of traditional random forest by reducing the contribution of trees whose nodes are populated with less informative features. The proposed method selects eligible subsets at each node by weighted random sampling as opposed to simple random sampling in traditional random forest. We refer to this modified random forest algorithm as "Enriched Random Forest". Using several high-dimensional micro-array datasets, we evaluate the performance of our approach in both regression and classification settings. In addition, we also demonstrate the effectiveness of balanced leave-one-out cross-validation to reduce computational load and decrease sample size while computing feature weights. Overall, the results indicate that enriched random forest improves the prediction accuracy of traditional random forest, especially when relevant features are very few.

4.1 Introduction

In recent years unprecedented increase in structural and functional analysis of human genome have presented enormous opportunities and challenges for machine learning researchers. High-throughput genomic technologies, including gene expression microarray, single nucleotide polymorphism(SNP) array, microRNA array, RNA-seq, ChIP-seq, and whole genome sequencing enabled us to detect variations that are associated with risk of diseases with finer resolution than before. In genomic applications, features usually correspond to genes, protiens (sequences), or single motifs. Let, n denote the number of training data samples, p the original feature dimension, the raw feature can be expressed as a set p-dimensional vectors: $x(t) = [x_1(t), x_2(t), ..., x_p(t)]^T, t = 1, 2, ..., n$. The feature dimension (p) can be extremely high, where as the sample size (n), is often severely limited. For example, in gene expression microarray data, features represent gene expression coefficients corresponding to the abundance of mRNA in a sample, for a number of patients. Usually, there are very few samples (often less than 100 patients) and the number of feature for each sample ranges from 6000 to 60,000. In this extreme of very few observations on very many features, classical regression framework is no longer applicable. Firstly, due to the small sample size over-fitting will be induced if all the features are used in classification/regression model. Secondly, the highly correlated structure of genomic data violates the independent assumption of traditional statistical models. Moreover, many biological mechanisms involve gene-gene interactions or gene networks. In high-dimensional setting, it is not realistic to prespecify such interaction effects in statistical models, especially high-order interactions. Generally, a small portion of genomic markers are associated with the phenotypes, and performing feature selection for high-dimensional, correlated, and interactive genomic data require sophisticated methodology. This leads to the challenge of "large p, small n" paradigm in biological big data which cannot be addressed by the widely used strategies such as deep learning employed in other big data areas.

With vast body of feature selection techniques, the need arises to determine which

technique to use in a given situation. Based on the evaluation criteria, feature selection algorithms are classified into three categories: 1) filter approaches; 2) wrapper approaches; and 3) embedded approaches. Wrapper approaches include a learning algorithm in the feature subset evaluation step. The learning algorithm is used as a "black box" by a wrapper to evaluate the goodness of the selected features. Given a classifier C, and given a set of features F, a wrapper method searches in the space of subsets of F, using cross-validation to compare the performance of the trained classifier C on each tested subset. A filter method, on the other hand, is independent of any learning algorithm. It does not make use of C, but rather attempts to find predictive subsets of features using simple statistics from the empirical distribution. For example, an algorithm that ranks features based on mutual information between the features and the class label. Filter algorithms are computationally less expensive and more general than wrapper algorithms. However, filters ignore the performance of the selected features on a learning algorithm. Wrapper algorithms achieve better performance than filter algorithms, but they may require orders of magnitude more computation time. In addition, in wrapper methods, repeated use of cross-validation on a single dataset can lead to uncontrolled growth in the probability of finding a feature subset that performs well on the validation data by chance alone. Embedded methods combine feature selection as well as classifier learning into a single process. Some embedded methods perform feature weighting based on regularization models with objective functions that minimize fitting errors and in the mean time force the feature coefficients to be small or exactly zero. Methods such as penalized regression, tree-based approaches, and boosting have been applied to handle high-dimensional problems.

As pointed out in literature, an ideal feature selection algorithm should achieve an optimal trade-off between *predictive performance*, i.e., the capacity of identifying the most relevant/predictive features, and *stability*, i.e., the robustness of results with respect to changes in dataset composition. In a problem with over 7000 features, filtering methods have significantly smaller computational complexity compared to wrapper methods. Previous studies that have analyzed microarray data have used filtering methods. However, it is also possible to exploit prediction-error-oriented wrapper methods

in context of large feature space. Wrapper methods has the risk of overfitting due to the reduced number of instances of microarray data and the small ratio between number of samples and number of features. Regularization methods trim the hypothesis space (i.e., the combinatorial space of feature subsets) by constraining the magnitude of parameters.

In this paper, we address the limitations of traditional RFs in high-dimensional setting, specifically, "large p small n" paradigm. We propose a novel method called Enriched Random Forest (ERF), that enhances traditional random forest by applying weighted random sampling, so that the chances of selecting less informative features are reduced. Odds of trees containing more informative features being included in the forest increases. Based on our proposed approach, we obtain a higher number of better base learners, and thus resulting in better fit. Another novel aspect of our approach is the effectiveness of balanced leave-one-out cross validation to reduce computational load as well as decrease the sample size while computing feature weights. This work extends our preliminary work [1], and addresses the future research goals set forth therein.

The remainder of the paper is organized as follows. A discussion of the related work is presented in section 4.2. In section 4.3, we discuss the details of the proposed approach. The experimental evaluation of the proposed approach and discussion of the results are included in sections 4.4 and 4.5. The conclusion and future work are presented in section 4.6.

4.2 Related Work

Feature selection is extremely important to address the large number of input features in high-dimensional supervised learning. It aims at selecting a subset of the original features, eliminating irrelevant and redundant features while achieving the best for a predetermined objective – the highest prediction accuracy. Feature selection is a difficult task mainly due to a large search space. For a dataset with p features, total number of possible solutions is 2^p. The task becomes more challenging as p becomes large and increases complexity of the problems. An exhaustive search for the best

feature subset of a given dataset is practically impossible in most situations. Another important challenge of feature selection is to account for feature interaction problems. There can be two-way, three-way, or complex multi-way interactions among features. A feature, which is weakly relevant to the target concept by itself, could significantly improve the accuracy if it is used together with some complementary features. In contrast, an individually relevant feature may become redundant when used together with other features. The principal reasons for feature selection in genomics are: (i) finding co-expressed genes to build metabolic pathways; (ii) biological relevance of individual genes for clinical diagnosis; and (iii) enhancement of classifier performance. In addition, feature selection also help data visualization, reduction of measurements, storage requirements, as well as reduction of data processing time.

Feature selection methods have received much attention in the classification literature. Xing et al. [19], reported the application of feature selection methods to classification problem using microarray data. Their approach was a hybrid of filter and wrapper approaches. The authors applied a sequence of simple filters called Markov Blanket Filter, to identify feature subsets for each subset cardinality. Cross validation was performed to compare between the resulting subset cardinalities. All of the classifiers that were studied – generative Gaussian classifier, discriminative logistic regression classifier, k-NN classifier, performed significantly better in the reduced feature space than in the original feature space. The proposed method explicitly eliminated redundant features. The study also compared feature selection to regularization methods. Results showed that explicit feature selection yields classifiers that perform better than regularization methods. Feature selection and regularization are not mutually exclusive and it would be worth considering their combinations.

Genomic data sets contain highly correlated variables, many of them being irrelevant for classification purpose. Although feature selection methods identify these noisy variables, it is to be noted that the term relevant is meaningful only in context of the objective function of the applied classifier. In addition, these data sets present challenge due to a large number of gene expression values per experiment and a relatively small

number of experiments. Czekaj et al. [7], demonstrated that the selected subsets of significant genes can vary in cardinality, and due to the redundancy (correlation) of genes, it is possible to select different minimal subsets of genes, necessary for classification. However, their interpretation ought to be made cautiously.

Guyon et al. [13], addressed the problem of selection of small subsets of genes from broad patterns of gene expression data. They used backward elimination procedure in linear Support Vector Machines (SVM), and referred to as SVM recursive feature elimination (SVM-RFE). Compared to other wrapper methods, SVM-RFE was scalable and efficient. Nested subsets of features were selected through sequential backward elimination, starting with all the feature variables and removing one feature at a time. At each step, the coefficients of the weight vector w of a linear SVM were used to compute the feature ranking score.The feature with the smallest ranking score was eliminated. The method was evaluated on two different cancer databases. Significant improvements were obtained over the baseline methods. The genes found by SVMs were biologically relevant in contrast to other methods that select genes correlated with the separation at hand and not relevant to the phenotype. Another similar study [14] provided an overview of the state-of-art feature selection methods. Sample applications for genomic signal processing were highlighted. The authors described the notion of self-supervision and developed a method called vector index adaptative SVM (VIA-SVM) for selection of features under self-supervision scenario. VIA-SVM was superior to SVM-RFE in two aspects: (i) it outperformed SVM-RFE at feature selection in low dimensions; and (ii) it automatically bounded the features within a smaller range. In addition, VIA-SVM was insensitive to the penalty factor in SVM training and avoids the need for a cut-off point to stop the feature selection process. Based on several experiments on microarray and SNPs data, VIA-SVM when combined with some filter provided substantial dimension reduction with significantly small decline in accuracy.

Multi-classifier systems exploit the strengths of diverse classifier models to obtain enhanced performance by their combination. This approach is referred as ensemble learning paradigm and has been extensively covered in pattern recognition and machine learning literature. In recent years, significant research efforts have explored the

extension of this paradigm to the feature selection process. Pes et al. [17], studied the effects and potential benefits of ensemble feature selection in the context of biomarker discovery from high-dimensional genomic data. They evaluated the effects of a specific ensemble approach, namely data perturbation. Data perturbation combines multiple selectors that exploit the same core algorithm but are trained on different perturbed versions of the original data. In this study, authors showed how the ensemble implementation improves the overall performance of the selection process, in terms of predictive accuracy and stability. Their results indicated that the beneficial impact of the ensemble approach is inversely proportional to the strength of the method. Only the least stable/effective methods take advantage of computationally expensive ensemble setting. They also measured the impact of the ensemble approach on final outcome, i.e., composition of the selected feature subsets. It turned out that different methods, when used in the ensemble version, tend to produce similar subsets. However, this does not explain the fact that their accuracy/stability patterns become almost coincident.

In [4], authors developed a framework for feature selection consisting of ensemble of filters and classifiers. Five filters based on different metrics were used. Each filter selected a different subset of features which is used to train and test a specific classifier. The outputs of these classifiers are then combined by simple voting. In this study, three well known classifiers were used for the classification task: C4.5, naive-Bayes, and instance based learner (IBL). The idea to use ensemble was to reduce the variability of selected features by using filters in different classification domains. The proposed method was evaluated using ten microarray data sets. The results obtained by the ensemble method achieved the lowest average error for each of the classifiers tested, showing the adequacy of the ensemble. In some specific cases, there was a filter that outperformed the ensemble. However, there was no better filter in general and the ensemble seemed to be the most reliable alternative for feature selection. The ensemble achieved best average error for the two classifiers C4.5 and IBL. IBL obtained the best error rates for 7 out of 10 data sets. For naive Bayes classifier, the results obtained by the ensemble in terms of average error was very close to the one obtained by best incremental ranked subset (BIRS), a wrapper method with the disadvantage of higher

computational cost.

Anaissi et al. [3], introduced ensemble SVM-Recursive Feature Elimination (ESVM-RFE) for gene selection that employ the concepts of ensemble and bagging used in random forest. The algorithm adopts backward elimination strategy to recursively eliminate features. The rationale for building ensemble SVM models using randomly drawn bootstrap samples from training set was to produce different feature rankings which would be subsequently aggregated as one feature ranking. Features were eliminated based upon the ranking of multiple SVM models instead of one particular model. The proposed approach also addressed the problem of imbalanced datasets by constructing nearly balanced bootstrap sample. The results of this study showed that ESVM-RFE increased the classification performance on five microarray datasets compared to state-of-art methods. When applied on the childhood leukaemia dataset, ESVM-RFE obtained average 9% better accuracy than SVM-RFE, and 5% over random forest approach. The genes selected by ESVM-RFE were further explored with Singular Value Decomposition (SVD) and significant clusters were found with the selected data. Another similar approach has been applied by Duan et al. [11] called multiple SVM-RFE. Unlike, SVM-RFE method, at each step, the method computes the feature ranking score from statistical analysis of weight vectors of multiple linear SVMs trained on sub-samples of training data. The results showed that the method selected better gene subsets than SVM-RFE and improved classification accuracy.

Random forests (RF) is a popular tree-based ensemble learning method that is highly adaptive to the characteristics of the data and applies to "large p, small n" problems. RFs also account for correlation as well as interactions among features. Chen et al. [6], reviewed the applications and progresses of RF for genomic data, including prediction, classification, variable selection, pathway analysis, genetic association, and unsupervised learning. The authors pointed out that a rigorous theoretical work of RF is needed. Its effectiveness in the non-standard small sample size and large feature space setting is not fully explored. Theoretical analysis should focus on asymptotic rates of convergence and answer questions, such as determining optimal values for RF parameters, *mtry* and *nodesize*, and provide ways to modify forests for improved

prediction performance. Also, trees and forests capture a lot of information about the data not typically available with other methods. Proximity can be used to quantify nearness of data points in high dimensions. Interactions between variables can be examined by studying the splitting behavior of the variables. This study discussed in detail the ways to utilize RFs for successful application to genomic data analysis.

Uriarte et al. [8], investigated the use of RF for classification of microarray data, including multi-class problems. They developed a new method of gene selection based on RF. The study used simulated and nine microarray datasets to compare the performance of RF to other classification methods, such as diagonal linear discriminant analysis (DLDA), k-NN, and SVM. The goal of the method was to yield smaller subsets of non-redundant genes while preserving predictive accuracy. The proposed method selected genes by iteratively fitting RFs, and at each iteration building a new forest after discarding the genes with smallest variable importance. The selected set of genes is the one that produced smallest error rate. The method used bootstrap technique to assess prediction error rates. Authors did not recalculate variable importance at each step because it could result in severe over-fitting. After fitting all forests, the out-of-bag (OOB) error rates of these forests were compared. The method chose the solution with smallest number of genes whose error rate is within u standard errors of the minimum error rate of all forests. When $u = 0$, it selected the genes that lead to the smallest error rate, and when $u = 1$, it was similar to "1 s.e. rule" used in classification trees. The results showed that this method returned small sets of genes compared to alternative variable selection methods when applied on simulated and real microarray datasets. The method did not return sets of genes that are highly correlated. It helped identify which genes have the largest signal to noise ratio and can be used as surrogates for complex processes involving many correlated genes. This study also examined the effects of changes in the parameters of random forest and the variable selection algorithm. A similar approach called guided regularized random forest (GRRF) proposed by Deng et al. [9] performed feature selection based on the importance score from a RF built on the complete training data complemented with the information gain in a local node. The trees in GRRF can be highly correlated and cannot be built in parallel. The guided

random forest (GRF) [10] addressed this limitation by using the importance scores from an RF and by having each tree built independently of one another.

In another similar study, Nguyen [16] used a two-stage quality based sampling method in RF for SNP subspace selection in Genome-wide association studies. The method applied p-value assessment to determine a cut-off point that separated the informative and non-informative SNPs in two groups. The informative SNPs were further subdivided into two groups: highly informative and weak informative SNPs. When sampling the SNP subspace for building trees for the forest, only those SNPs from the two subgroups were considered. The feature subspace always contained highly informative SNPs when used to split a node at a tree. The authors performed extensive experiments on two genome-wide SNP datasets and 10 gene datasets to demonstrate the effectiveness of their proposed method. Results indicated that the proposed method significantly reduced prediction errors and outperformed most state-of-art variants of RF. The approach enabled to generate more accurate trees with lower prediction error and also avoid over-fitting.

Ge et al. [12], developed a feature selection algorithm based on correlation measurement, Maximal Information Coefficient (MIC). This method selected features associated with phenotypes independently of each other and used nearest neighbor classification algorithm. Comparative study based on 17 datasets indicated that the method performed as well or better than existing methods, and significantly reduced the number of selected features. The selected features also appeared to have biomedical relevance to the phenotypes in the literature.

In this paper, we propose a novel method called Enriched Random Forest (ERF), that performs feature selection by sampling the variables used to partition each node according to a given set of weights assigned to each variable. As pointed out previously, in traditional RF, simple random sampling is used for selecting the subset of eligible features at each node, thus almost all these subsets are likely to contain a preponderance of non-informative features. To overcome this limitation of traditional RF, ERF applies weighted random sampling, assigning lower weight to the less informative features. If the weights of all the variables are set to one then the algorithm becomes standard

random forest and if the weight of a variable is set to zero then the variable will be excluded from the training data. To evaluate our method, we applied ERF to various gene expression dataset and compared its performance to that of traditional RF.

4.3 Enriched Random Forest

4.3.1 Background

Random Forest (RF) proposed by Breiman (2001) adds an additional layer of randomness to bagging that builds on large collection of de-correlated trees, and then average them. In addition to using different bootstrap samples for constructing each tree, in RF each node is split using the best split among all variables. The performance of RF is similar to boosting as well as they are simple to train and tune. The essential idea is to average many noisy but approximately unbiased models, and hence reduce the variance. Trees are used as the base learner in bagging, since they can capture complex interaction structures in the data, and if grown sufficiently deep, have relatively low bias. Also, trees are inherently noisy, so they benefit greatly from averaging.

In bagging, successive trees do not depend on earlier trees. Each tree is constructed independently using a bootstrap sample of the training data and is identically distributed (i.d.). Thus, the expectation of an average of B trees is the same as expectation of any one of them. This means the bias of bagged trees is the same as that of individual trees, and the only improvement can be achieved through variance reduction. An average of B i.i.d. random variables, each with variance σ^2 , has variance $\frac{1}{B}\sigma^2$. If the variables are simply i.d. with positive correlation ρ, the variance of the average is:

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{4.1}$$

As B increases, the second term diminishes, and the size of the correlation between pairs of bagged trees limits the benefits of averaging. RF improves variance reduction by reducing the correlation between the trees without increasing the variance too much. This is achieved in the tree growing process through random sampling of the predictor variables. When growing a tree on bootstrapped dataset, before each split, $m \leq p$ of

the predictor variables are selected at random as candidate for splitting. For regression, the default value for m is $\lfloor \frac{p}{3} \rfloor$ and the minimum node size is five. For classification, the default value for m is $\lfloor \sqrt{p} \rfloor$ and the minimum node size is one. After B such trees are grown, the RF predictor (regression) is given by:

$$\hat{f}_{rf}^{B}(x) = \frac{1}{B} \sum_{b=1}^{B} T(x; \Theta_b) \tag{4.2}$$

When used for classification, random forest obtain a class vote from each tree, and then classifies using majority vote.

4.3.2 Out of Bag Samples

An important feature of RFs is its OOB samples. For each observation $z_i = (x_i, y_i)$, random forest predictor is constructed by averaging only those trees corresponding to bootstrap samples in which z_i did not appear. An OOB error estimate is identical to that obtained by N-fold cross validation. Hence, unlike many other nonlinear estimators, RF can be fit in one sequence, with cross validation being performed on the way. Once, the OOB error stabilizes, the training can be terminated.

4.3.3 Variable Importance

RF also use the OOB samples to construct variable importance measure, to measure the prediction strength of each variable. When the b^{th} tree is grown, the OOB samples are passed down the tree, and the prediction accuracy is recorded. Then, the values for the j^{th} variable are randomly permuted in the OOB samples, and the accuracy is again computed. The decrease in accuracy as a result of this permuting is averaged over all trees, and is used as a measure of the importance of variable j in the RF.

4.3.4 Limitations of Random Forest

Although traditional RF works well in datasets with many features (large p), when the percentage of truly informative features is small, such as with DNA microarray data, its performance tends to decline significantly. In previous studies, Moechars et al.

[15], and Raghavan et al. [18], illustrated this point using an experiment conducted to study whether mice whose Slc17A5 gene has been knocked out could be distinguished from wild type mice at the gene expression level. Gene expression measurements were taken on newborn (0-day-old) mice as well as on 18-day-old mice. At day 0, there were no obvious occurrence of any phenotypic variations in the knockout mice but subtle effects would have already begun at the cellular level. By day 18 phenotypic variations in the knockout mice are evident with observable morphological alterations such as defects in myelination. The separation of the 18-day-old mice is straightforward both physiologically and with gene expression data. On applying traditional RF, an out-of-bag error rate of less than 10% is obtained. On the other hand, it is a challenge to seprate the newborn mice, not only physiologically, but even with gene expression data; the out-of-bag error rate for RF is over 50%.

Let us consider a situation with p features, of which only H are informative. Then, if at any node m features are selected by resampling randomly with equal weights, the probability distribution of the number of informative features selected is binomial with m trials and probability $\pi = \frac{H}{p}$. The mean number of informative features selected at each iteration is $\mu = \pi m$. Since π is typically very small, so will μ be. For example, if $H = 100, p = 10,000$ and $m = p^{1/2} = 100$, the resulting μ is only one informative feature per node. The trees built using such nodes will have low accuracy and overall performance of the ensemble will suffer. Thus, in situation like this, traditional RF algorithm can be considerably enhanced by reducing the contribution of trees whose nodes are populated by less informative features. To some extent, this can be achieved by pre-filtering, but here we develop a novel adjustment that has demonstrated superior performance when applied on high dimensional genomic datasets with too few truly informative features. We choose eligible subsets for splitting at each node by weighted random sampling instead of simple random sampling, with the weights tilted in favor of the informative features. This results in Enriched Random Forest.

4.3.5 Enriched Random Forest Algorithm

Enriched Random Forest enhances the performance of basic Random Forest method by reducing the contribution of trees whose nodes are populated by less informative features. ERF uses weighted random sampling instead of simple random sampling, so that less informative features are less likely to get selected and the odds of trees containing more informative features being included in the forest increases. Consequently, the ERF comprises of a higher number of better base learners, resulting in a better fit. ERF algorithm samples the variables used to partition each node according to a set of given weights assigned to each variable. If the weight of a variable is zero then the variable is excluded from the training.

Given a training set X consisting of n observations, an outcome variable Y, and p features, a tree is constructed as: a feature x and a threshold t that splits X into two subsets that are maximally distinct according to a specified criterion are selected from all features of X and all possible values of t. The training set is then split into the two buckets X_L and X_R depending on whether or not $x < t$. This procedure is repeated with each of X_L and X_R using another (x, t) combination until no further splitting is possible. In a random forest, a tree, rather than being trained on the entirety of the training set, is trained only on a sample of n observations drawn at random with replacement from the complete set of n observations. Additionally, when determining which feature to split on at each node, only a subset of m of the p features (usually $m = p^{1/2}$) are considered eligible; this subset is drawn at random with out replacement independently for each node from the complete set of p features. A RF is an ensemble of R number of such trees, where each tree is called a base learner. For classification, classes are assigned to test cases by majority vote: when given a test case, each tree assigns it a class according to its classifier; this information is collated and overall the forest assigns it the majority class. For regression, the outcome for a test case is predicted as the average of the values predicted by each tree. ERF uses weighted random sampling instead of simple random sampling. Weighting is done by scoring each feature based on its ability to separate the groups, e.g. via a t-test or chi-square test,

and using these scores to assign weights, w_i, so that the features that most separate the groups are assigned higher weights. Once the weights are determined, at any node, the subset of m eligible features is selected from the p features using weighted random sampling with weights w_i rather than simple random sampling. Below is an overview of our approach, followed by a detailed discussion of the feature scoring technique.

1. We split the given n observations with p variables randomly into two samples: in-bag samples (68% of n) and out-of-bag samples (32% of n).

2. Next, build a tree on the in-bag sample using the Classification and Regression Trees (CART) algorithm (or use any alternative splitting criterion) with two modifications.

   (a) To perform the split at each node, we use "mtry" variables (usually $\sqrt{p}$ or $\frac{p}{3}$) selected using the weight vector of probabilities W.

   (b) The complete tree is built without pruning.

3. We use the tree built using the in-bag samples to predict the outcome variable for the out-of-bag samples.

4. Steps 1-3 are repeated at least N = 1000 times and the out-of-bag predictions are stored in a matrix of dimension $n \times N$ where the entries for the in-bag observations of each column are missing values. If the response is categorical, we calculate for each row the most frequent prediction and assign that prediction to the observation of that row. In case of continuous response, the predicted value for each observation/row equals average of that row.

4.3.6 Weighting the Features

The key to ERF is to score each feature based on how well it separates the groups. Such score is generated by computing the correlation between the predictor variable and the response when both are of continuous numeric type. If the response is a binary variable and the predictor is continuous, we test each feature for a group mean effect using two sample t-test and one-way anova. When both response and predictor are

categorical, we perform chi-square independence test to determine significant difference between the expected frequencies and the observed frequencies in one or more categories. Next, we obtain a p-value from these significance tests, small p-value indicates greater separation and large p-values indicate less separation. However, to weight using the p-values themselves would fail to take into account: (i) the multiplicity of tests being performed; and (ii) the small sample sizes typical of microarray experiments. To adjust for the multiplicity problem, we compute the weights based on q-values, which are calculated from the p-values as: $q_i = min_{k \geq 1}\{min((p/k)pval(k), 1)\}$, where $p_{(i)}$ and $q_{(i)}$ are the p-value and q-value associated with the feature with *i-th* smallest p-value. The q-values provide false discovery rate (FDR) adjusted measures of significance for the features and are in the same order as the p-values. In addition, the use of q-values instead of p-values help lessen the likelihood of overfitting in situations with no separation of the data into groups. If p-value based weights were used, some genes by chance would have small p-values and would wrongly be assigned higher weights. This would result in ERF mistakenly implying a separation. If q-value based weights were used, all genes would be assigned equal weights and ERF would not find a separation. The standard way to compute weights of the predictor variable was by computing the logarithm of q-values. For applying a steeper transformation, we could apply $W = 1/Q$. Based on this weighting, features with less separability will get zero weight and features with high separability will get large weights. To adjust for (ii), we used Conditional t-test (Ct) [2] instead of usual t-test since it is likely to generate a better ranking of features. The usual t-test has low power and thus low discriminatory ability when the sample size is small.

Error rates could be underestimated if the weights are calculated just once based on all the samples than if they were determined separately for each tree based on only the in-bag samples. But, this would increase computational burden and render the weights less well determined than if they had been calculated outside the loop using all the samples. Here, we also implement another variant of ERF called ERF-CV that perform balanced leave-one-out cross-validation instead of bagging to lighten the computational load and to decrease the sample size when determining weights. Let

Algorithm 2 Algorithm for Enriched Random Forest

Input: A training set $S = (x_1, y_1), ..., (x_N, y_N)$, features F, and the number of trees in forest B

Output: The learned forest H

function *Enriched Random Forest* (S,F)
Initialisation :
$H \longleftarrow \phi$
LOOP Process:
for $i = 1$ to B **do**
  $S^{(i)} \longleftarrow$ A bootstrap sample S
  $h_i \longleftarrow$ *Randomized Tree Learn* ($S^{(i)}$,F)
  $H \longleftarrow H \cup \{h_i\}$
end for
return H
end function
function *Randomized Tree Learn* ($S^{(i)}$, F)
At each node
$W \longleftarrow$ *Compute Weight* ($S^{(i)}$, F)
$f \longleftarrow$ Subset of F using weighted random sampling
Split on best feature in F
return The learned tree h
end function

$J = R/N$. In ERF-CV, in J of the R trees,one observation is set aside as out-of-bag test set, the weights are calculated based on the $N-1$ in-bag cases and a tree is derived based on these in-bag cases. The prediction is done on the OOB case. This is repeated with each of the other cases. Less computation is required for ERF-CV than for ERF since weights are calculated only N times rather R times.

4.4 Experimental Evaluation

We implemented the proposed approach on different microarray datasets to evaluate its effectiveness in both regression and classification setting as compared to the traditional RF.

4.4.1 Regresion

Dataset 1: RNA Data

This dataset contain gene expression of 25000 genes for 100 observations. In such

high-dimensional datasets, it is supposed that a lot of variables are non-informative and that there exist unknown groups of highly correlated predictors. Applying the ERF algorithm, we perform feature selection in way such that the subset of eligible features at each node contain a preponderance of truly informative features. We split the data into train and test sets based on i the suggested train and test indices included in the data file. Here, we compute "pseudo R-squared" as indicated by Breiman (2001) [5]. Generally, explained variance (R^2) is defined as: $R^2 = 1 - \sum((\hat{y} - \bar{y})^2) / \sum((\bar{y} - y)^2)$, and takes value between 0 and 1. On the other hand "pseudo R-squared" is defined as: $R^2 = 1 - (Mean\ Squared\ Error)/var(y)$, which, mathematically can produce negative values. A simple interpretation of negative R^2, is that we are better off predicting any given sample as equal to overall estimated mean, indicating very poor model performance.

In random forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally. Each tree is constructed using a different bootstrap sample from the original data. About one-third of the cases are left out of the bootstrap sample and not used in the construction of the k^{th} tree. Each case left out in the construction of the k^{th} tree is used to estimate the error. This are called out-of-bag samples. However, our implementation also provides the flexibility to carry out cross-validation applying hold-out approach. We compared our proposed method to traditional random forest using out-of-bag samples as well as hold-out approach. Table 1. illustrates the performance of enriched random forest in contrast to traditional RF when applied to the *rnadata*. **Dataset 2: Toxicity Data**

Table 4.1: Predictive Performance of ERF and RF on RNA Data

	OUT-OF-BAG		HOLD-OUT SET	
Methods	MSE	R^2	MSE	R^2
Enriched Random Forest	3.87	0.15	3.46	0.13
Traditional Random Forest	4.70	-0.08	3.86	-0.12

Next, we also applied the method on another similar gene expression data, *liver.toxicity*, available in the R package *mixOmics*. This is a real dataset from a study by Heinloth et al.(2004), in which four male rats of the inbred strain Fisher 344 were exposed to different doses of acetaminophen (non toxic dose 50 or 100mg/kg), moderate toxic dose

(100mg/kg), and severe toxic dose (2000 mg/kg) in a controlled experiment. Necropsies were performed at different hours after exposure (6,18,24, and 48 hours) and the mRNA from the liver was extracted. In the original study, 10 clinical variables containing markers of liver injury were measured. However, the dataset used in our analysis contains: (i) a data frame, called *gene*, of size 64 rows representing the subjects and 3116 columns representing explanatory variables which are gene expression levels; and (ii) a vector, called *clinic*, with 64 rows and 1 column, one of the 10 clinical variables for the same 64 subjects: more precisely, the variable named $ALB.g.dL$, which corresponds to the albumin level. Table 2. illustrates the performance of enriched random forest in contrast to traditional RF when applied to the *liver.toxicity* data.

Table 4.2: Predictive Performance of ERF and RF on Liver Toxicity Data

	OUT-OF-BAG		HOLD-OUT SET	
Methods	MSE	R^2	MSE	R^2
Enriched Random Forest	0.04	0.4	0.02	0.7
Traditional Random Forest	0.05	0.24	0.02	0.62

4.4.2 Classification

Dataset 3: Slc17A5 Data

For classification task, we use the *Slc17A5 Day 0*, *Slc17A5 Day 10*, and *Slc17A5 Day 18* data. These datasets capture gene expression measurements of 45,101 genes for 12 samples belonging to two separate classes taken on newborn, 10-day-old, and 18-day-old mice respectively. *Slc17A5 Day 0* dataset is the primary dataset for our evaluation. The *Slc17A5 Day 18* dataset, which has unequivocal separation of classes, is used to assess the performance of ERF when there is strong signal. The *Slc17A5 Day 10* dataset captures an intermediate stage. In addition, we also created artificial datasets by random permutation of the *Slc17A5 Day 0*, *Slc17A5 Day 10*, and *Slc17A5 Day 18* datasets. These datasets were used to verify that the method is not overfitting. If the weighting is not done carefully, it is possible to find spurious classifications in datasets that have no true separation.

In classification, the out-of-bag data is used to get a running unbiased estimate

of the classification error as trees are added to the forest. Each case left out in the construction of the k^{th} tree is included in the out-of-bag data to get a classification for the k^{th} tree . In this way, a test set classification is obtained for each case in about one-third of the trees. At the end, take the class that got most of the votes every time case n was in out-of-bag data. The proportion of times the predicted class is not equal to the true class of n averaged over all cases is the out-of-bag error estimate. Table 3. display the results of enriched random forest and traditional RF when applied to the *Slc17A5* gene expression measured at day 0, day 10, and day 18.

Table 4.3: Predictive Performance of ERF and RF on *Slc17A5* Gene Expression Data

	Day 0	*Day 10*	*Day 18*
Methods	OOB Err. Rate	OOB Err. Rate	OOB Err. Rate
ERF	0.08	0	0
Traditional RF	0.58	0.47	0

The results on the permuted datasets are presented in Table 4.

Table 4.4: Predictive Performance of ERF and RF on *Slc17A5* Gene Expression Data

	Day 0	*Day 10*	*Day 18*
Methods	OOB Err. Rate	OOB Err. Rate	OOB Err. Rate
ERF	0.75	0.73	0.75
Traditional RF	1	0.8	0.83

Dataset 4: SRBCT Data

Our method is also applicable when the response variable has multiple groups. Here, we applied our proposed method on the *SRBCT* data available in the R package *mixOmics*. This real classification dataset is a small version of the small round blue cell tumors of childhood data and contains the expression measure of genes measured on 63 samples. The dataset is composed of: (i) a data frame, called gene, of size 63×2308 which contains the 2308 gene expressions; and (ii) a response factor of length 63, called class, indicating the class of each sample (4 classes in total). To verify that our method is not overfitting we performed y-randomization test. The values of response variable (class) are randomly ascribed (scrambled) to different samples, while the descriptors values

(genes) are left intact. Scrambled data are then used for training the model. The test indicate the quality of obtained models in comparison to chance models derived from random data. The results are displayed in Table 5.

Table 4.5: Predictive Performance of ERF and RF on *SRBCT* Gene Expression Data

	Original Data	*Scramled Data*
Methods	OOB Err. Rate	OOB Err. Rate
ERF	0.01	0.67
Traditional RF	0.01	0.75

We also performed similar experiments to compare the performances of ERF-CV and ERF on original and scrambled *Slc17A5 Day 0* and *Slc17A5 Day 18* data. ERF obtained error rates of 0.17 and 0.00 on original *Slc17A5 Day 0* and *Slc17A5 Day 18* datasets respectively. ERF-CV obtained 0.08 and 0.00 on original day 0 and day 18 data. On the other hand, on scrambled datasets ERF achieved an error rate of 0.83 and 0.68, while ERF-CV obtained 0.75 and 0.42 on day 0 and day 18 datasets respectively.

4.5 Discussion

Enriched Random Forest works best when applied to datasets that have subtle signal. If the signal were strong or non-existent, both ERF and RF would produce essentially the same result. Table 1 and 2, display the results of ERF when applied to two such datasets *RNA data* and *liver.toxicity data*. They show that ERF outperforms traditional RF in terms of Mean Square Error (MSE) and R^2. When applied to the *RNA data*, ERF achieves out-of-bag MSE of 3.87 in contrast to traditional RF which achieves out-of-bag MSE of 4.70. The pseudo-R^2 of ERF was found to be 0.15 whereas for traditional RF it was -0.08. In hold-out set approach, ERF and traditional RF obtained MSE of 3.46 and 3.86 respectively. The pseudo-R^2 for ERF and RF were 0.13 and -0.12 respectively. As explained previously, negative value of R^2 indicate that we are better off predicting any given sample as equal to overall estimated mean, indicating very poor model performance. Therefore, ERF performs well in comparison to traditional random forest when the percentage of truly informative feature is very small (i.e., the

signal is subtle). Traditional RF have little or no predictive power in such situation. Similarly, when applied on *liver.toxicity* data ERF obtained out-of-bag MSE of 0.04 and pseudo-R^2 of 0.4. Traditional RF, on the other hand, obtained out-of-bag MSE of 0.05 and pseudo-R^2 of 0.24. In hold-out set approach, MSE for both ERF and RF was found to be 0.02, and pesudo-R^2 of ERF and RF was found to be 0.7 and 0.6 respectively. When there is true signal in the data, enriched random forest performs consistently equally or better than standard random forest.

For classification task, we compare the out-of-bag error rates of ERF and traditional RF on three separate microarray datasets – *Slc17A5 Day 0*, *Slc17A5 Day 10*, and *Slc17A5 Day 18*. A good classifier should have low out-of-bag error rates for original datasets and high out-of-bag error rate for scrambled datasets. Table 3, display the results of ERF and traditional RF on the original *Slc17A5* datasets. The out-of-bag error rates for ERF were 0.08, 0, and 0 when applied to Day 0, Day 10, Day 18 gene expression measurement data. Traditional RF obtained error rate of 0.58 on Day 0, 0.47 on Day 10, and 0 for Day 18 measurements. At day 0, the phenotypic variations in the knockout mice were very subtle and mostly at the cellular level. By day 18 phenotypic variations are evident with observable morphological alterations. The separation of the 18-day-old mice is therefore more straightforward as the genes are fully expressed. Day 10 is an intermediate stage in the development process. Table 4, illustrates the performance of ERF compared to traditional RF on scrambled datasets. The out-of-bag error rates for ERF were 0.75, 0.73, and 0.75 when applied to day 0, day 10, day 18 measurements data. Traditional RF had error rate of 1, 0.8, and 0.83 on day 0, day 10, and day 18 data. These high out-of-bag error rates validate that ERF does not overfit unlike many other classifiers. In case of multiple groups, we evaluated our proposed approach using the *SRBCT* gene expression dataset. Our results indicate that both ERF and traditional RF perform equally well on this dataset, achieving an out-of-bag error rate of 0.01. To test for overfitting, we performed y-randomization test and found that the out-of-bag error rate increased significantly – ERF (error rate = 0.67) and RF (error rate = 0.74).

Our experiments also confirmed that ERF-CV performed equally compared to ERF

and were both significant improvements over traditional RF. By large, the ERF and ERF-CV error rates were similar to each other. Thus, ERF-CV could be more useful in practice since it is less computationally intensive and less prone to small sample sizes.

4.6 Conclusion

In this paper, we proposed a novel approach to enhance the traditional random forest algorithm to better perform in "large p, small n" paradigm. In contrast to the traditional RF, our proposed ERF method uses weighted random sampling to select subsets that has preponderance of informative features for splitting at each node. We extensively evaluated the effectiveness of our approach using several high-dimensional genomic datasets. Our main contribution is two-fold: (i) We applied weighted random sampling instead of simple random sampling, so that chances of selecting less informative features are reduced and odds of tree containing more informative features being included in the forest increases. Overall, our results indicate that ERF outperformed traditional RF when the signal is subtle. This means that only a small fraction of the features are truly informative. In case where the signal is strong and the data is easily separable ERF performed consistently equally and better than traditional RF. (ii) We also demonstrated how ERF-CV perform balanced leave-one-out cross-validation instead of bagging to lighten the computational load and decrease the sample size when determining weights.

We have extended the work of Amartunga et. al [1] which discussed ERF only in the two-group classification context. Here, we have proposed an extension to the case of multiple groups. In addition, we incorporated the idea of applying ERF in regression setting. Our implementation also addresses the challenge associated with variables that have mixed data types. We have applied appropriate statistical significance tests based on the data type of the predictor and response variables. Our future work will focus on further improvement in the achieved accuracy of the prediction model. In multinomial classification, complexity grows as the features that separate any two groups could differ substantially from the features that separate any two other groups. A possible direction to pursue is to possibly involve collation of multiple pairwise analyses.

We conjecture that this idea could be incorporated into other ensemble and machine learning techniques such as linear discriminant analysis, logistic regression, and SVM.

Currently, we are in the process of having our code as an R package that will implement the proposed methodology, thus making it widely available for use by other researchers.

References

[1] Amaratunga, Dhammika and Cabrera, Javier and Lee, Yung-Seop, *"Enriched random forests"*, Bioinformatics, 24, 18, 2010-2014.

[2] Amaratunga, Dhammika and Cabrera, Javier, *"A conditional t suite of tests for identifying differentially expressed genes in a DNA microarray experiment with little replication"*, Statistics in Biopharmaceutical Research, 1, 1, 26-38.

[3] Anaissi, Ali and Goyal, Madhu and Catchpoole, Daniel R. and Braytee, Ali and Kennedy, Paul J., *"Ensemble feature learning of genomic data using support vector machine"*, PloS one, 11,6, e0157330, 2016.

[4] Bolón-Canedo, Verónica and Sánchez-Maroño, Noelia and Alonso-Betanzos, Amparo, *"An ensemble of filters and classifiers for microarray data classification"*, Pattern Recognition, 45, 1,531-539.

[5] Breiman, Leo, *"Random forests"*,Machine learning, 45, 1,5-32. Springer, 2001.

[6] Chen, Xi and Ishwaran, Hemant, *"Random forests for genomic data analysis"*, Genomics, 99, 6,323-329, 2012.

[7] Czekaj, Tomasz and Wu, Wen and Walczak, Beata, *"Classification of genomic data: Some aspects of feature selection"*, Talanta, 76, 3,564-574, 2008.

[8] Díaz-Uriarte, Ramón and De Andres, Sara Alvarez, *"Gene selection and classification of microarray data using random forest"*, BMC bioinformatics, 7, 1, 3, 2006.

[9] Deng, Houtao and Runger, George, *"Gene selection with guided regularized random forest"*, Pattern Recognition, 46, 12,3483-3489, 2013.

[10] Deng, Houtao, *"Guided random forest in the RRF package"*, arXiv preprint arXiv:1306.0237. 2013.

[11] Duan, Kai-Bo and Rajapakse, Jagath C and Wang, Haiying and Azuaje, Francisco, *"Multiple SVM-RFE for gene selection in cancer classification with expression data"*, IEEE transactions on nanobioscience, 4, 3,228-234, 2005.

[12] Ge, Ruiquan and Zhou, Manli and Luo, Youxi and Meng, Qinghan and Mai, Guoqin and Ma, Dongli and Wang, Guoqing and Zhou, Fengfeng, *"McTwo: a two-step feature selection algorithm based on maximal information coefficient"*, BMC bioinformatics, 17, 142, 2016.

[13] Guyon, Isabelle and Weston, Jason and Barnhill, Stephen and Vapnik, Vladimir, *"Gene selection for cancer classification using support vector machines"*, Machine learning, 46, 1-3, 389-422. 2002.

[14] Kung, Sun-Yuan and Luo, Yuhui and Mak, Man-Wai, *"Feature selection for genomic signal processing: Unsupervised, supervised, and self-supervised scenarios"*, Journal of Signal Processing Systems, 61, 1, 3-20, 2010.

[15] Moechars, D and Van Acker, N and Cryns, K and Andries, L and Mancini, G and Verheijen, F., *"Sialin-deficient mice: a novel animal model for infantile free sialic acid storage disease (ISSD)"*. Society for Neuroscience 35th Annual Meeting, Washington, USA, 2005.

[16] Nguyen, Thanh-Tung and Huang, Joshua Zhexue and Wu, Qingyao and Nguyen, Thuy Thi and Li, Mark Junjie, *"Genome-wide association data classification and SNPs selection using two-stage quality-based Random Forests"*, BMC genomics, 16, 2, 55, 2015.

[17] Pes, Barbara and Dessì, Nicoletta and Angioni, Marta, *"Exploiting the ensemble paradigm for stable feature selection: a case study on high-dimensional genomic data"*, Information fusion, 35, 1, 132-147, 2017.

[18] Raghavan, Nandini and De Bondt, An MIM and Talloen, Willem and Moechars, Dieder and Göhlmann, Hinrich WH and Amaratunga, Dhammika, *"The high-level similarity of some disparate gene expression measures"*, Bioinformatics, 23, 22, 3032-3038.Oxford University Press, 2007.

[19] Xing, Eric P. and Jordan, Michael I. and Karp, Richard M., *"Feature selection for high-dimensional genomic microarray data"*, ICML, 1, 601-608, 2001.