MATHEMATICAL OPTIMIZATION METHODS FOR CLUSTERING AND CLASSIFICATION WITH BIOLOGICAL AND MEDICAL APPLICATIONS

BY CHUN-AN CHOU

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Industrial and Systems Engineering

Written under the direction of

Wanpracha Art Chaovalitwongse

and approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2011

ABSTRACT OF THE DISSERTATION

Mathematical Optimization Methods for Clustering and Classification with Biological and Medical Applications

by Chun-An Chou

Dissertation Director: Wanpracha Art Chaovalitwongse

The focus of the thesis is on the development of effective combinatorial optimization approaches for both large-scale clustering and classification problems in data mining with high computational complexity by massive biological and medical data.

In the first part, we study an important clustering problem in computational and population biology, namely sibling reconstruction problem. The problem is mathematically considered a special case of capacitated clustering problem. A mathematical optimization model is proposed to establish the sibling relationships (i.e., groups of siblings) based on the biological concept of combinatorial constraints and similarity likelihood of genetic data. Both exact and heuristic solution approaches are developed, which enable the problem to be solved comparably and outperform other existing combinatorial and statistical approaches significantly.

In the second part, we develop new combinatorial and pattern-based optimization approaches in the framework of Logical Analysis of Data (LAD) for binary classification. In the framework, while patterns are the building blocks for the LAD classification model, a new mathematical optimization model is proposed for generating decisive and high-quality patterns. Moreover, a column generation framework, where the proposed

pattern generation approach is employed, is developed to build an "optimal" LAD classifier such that the classification accuracy and computational efficiency are improved.

In the third part, we investigate feature selection that has two-fold advantages in classification problems with massive data: data reduction and noise reduction. First, we formulate a quadratic program by using statistical information (relevancy and redundancy) of features as inputs to select critical features that are favorable for classifiers. Second, we propose a new pattern-based optimization approach using a decomposed nearest neighbor rule for direct classification. The preliminary results show the potential for the improvement in data reduction and classification accuracy.

Acknowledgements

First of all, I love to thank Wanpracha Art Chaovalitwongse, Professor I admire the most, for inspiring, supporting, and advising me during the PhD study in the department of Industrial and Systems Engineering at Rutgers University. Because of his advice, I was able to fully dedicate myself to studying operations research in biological and medical applications. I am also obliged to Professors Hoang Pham, Endre Boros, Myong-K. Jeong, and Tanya Y. Berger-Wolf for being the committee members of my dissertation so as to make the dissertation complete.

During the years, when studying in the department of Industrial and Systems Engineering, I felt that I lived with a warming family where all faculties, staffs, and students are very kind and love to help others. I am very grateful to many people. in the department and RUTCOR. Special thanks to Professor Susan Albin, Professor Elsayed Elsayed, Cindy Ielmini, Helen Pirrello, Joseph Lippencott, Dr. Chungmok Lee, Liang Zhe, Ya-Ju Fan, and Shouyi Wang. Besides, many thanks to the sibling research group in the University of Illinois at Chicago, Professor Bhaskar DasGupta and Professor Mary V. Ashley, Saad Sheikh, and Isabel C. Caballero.

I also like to take the chance to thank all my dear friends in New Jersey, New York City, and Taiwan. They made my study life full of memorable happiness.

Finally, I love to thank my dearest parents who always fully support and encourage me whenever I need them. I also love to thank Tammy Chou for accompanying me during my Ph.D. Study.

This dissertation work has been supported by the National Science Foundation (IIS-0611998) and partially supported by the ISE department. I would like to thank them for their support.

Dedication

To my parents, Te-Jen Huang and Shu-Huei Chou

Table of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

Clustering and *classification* in data mining have frequently appeared in various practical areas, such as engineering, health care, biology, finance, etc. Data mining is a process of discovering useful knowledge from database to build a structure (i.e., model or pattern) that can meaningfully interpret the data. Clustering, an unsupervised learning, is to find cluster information without historical information such that members in each cluster are similar. Classification, a supervised learning, is to construct a prediction model for classifying future events in a way consistent with historical information.

When many statistical and machine learning methods have been developed, mathematical programming techniques, even combined with them, have started being successfully applied to clustering and classification problems. However, when real-life clustering and/or classification problems with massive data are formulated as mathematical programming models, there is a challenge to be faced because of highly computational complexity increased by the size and dimensionality of data. For this reason, there are a vast amount of investigations into developing new computational approaches to effectively and efficiently solve such large-scale and complex problems.

The focus of this dissertation is on developing effective mathematical optimization models and efficient computational algorithms for the problems in clustering, classification and feature selection. The presentation of the thesis is structured as follows.

In Chapter 2, we study an important problem recently arising in computational population biology, namely, sibling reconstruction problem. The goal of this study is to develop mathematical optimization models and computational algorithms based on the concepts of combinatorics and statistical likelihood to reconstruct the sibling

relationships. The input data sets used are multi-featured genetic markers of a single-generation population, which were sampled without parentage information. In this chapter, we propose a mathematical programming model based on the combinatorial concept from the inheritance rules (i.e., Mendel's Laws [30, 107]), and a more sophisticated mathematical programming model combining the similarity measure function. The studied problem also can be presented as a special version of the capacitated clustering problem. Due to the highly computational complexity, we develop an exact approach and greedy heuristic approaches to effectively and efficiently solve such a large-scale optimization problem. We present the experimental results on real biological and simulated data sets and show the comparable performance with the existing sibling reconstruction approaches.

In Chapter 3, for binary classification, we develop accurate prediction models based on the technique of Logical Analysis of Data (LAD) that particularly deal with binary inputs/outputs. A LAD classification model mainly consists of the sets of positive and negative patterns/rules from binarized features of known (numerical) data sets. To efficiently generate decisive patterns, we propose a new mathematical programming approach for pattern generation. Further, we develop a new column generation framework, where widely used objectives are considered in master problem and the proposed pattern generation approaches is employed in subproblem, to construct an accurate LAD classification model. We present the experimental results on various benchmark data sets in medical prognosis and diagnosis.

In Chapter 4, we attempt to develop new optimization-based feature selection approaches because the feature selection are shown to be beneficial for improving the performance in classification or clustering problems. The selected feature subset, instead of the full feature set, is used in classification models (or classifiers). First, we propose a new quadratic program incorporated with statistical information as inputs, which ultimately selects a compact subset of informative features. Second, we also propose a pattern-based optimization approach, called decomposed support feature machine, using a decomposed nearest neighbor rule, which is directly applied to classification. The preliminary results on a number of biomedical data sets show the potential for the

improvement of classification performance.

In Chapter 5, the dissertation is concluded.

Chapter 2

CLUSTERING STUDY IN COMPUTATIONAL BIOLOGY: SIBLING RECONSTRUCTION PROBLEM[1]

2.1 Introduction

In studies of many natural populations, it is more practical to sample genetic data from a cohort of individuals without parental information. The *sibling reconstruction* problem (SRP) is a problem of establishing sibling relationships (i.e., groups of individuals who are siblings) among individuals using their genetic markers that are sampled and genotyped from a single generation without parental information. Such problem has become increasingly more important to computational biologists and population biologists as the sibling relationships enable them to study several fundamental biological phenomena, including mating systems, ecological behaviors and evolution, and social organizations. *Microsatellites*, also known as simple sequence repeats (SSRs) or short tandem repeats (STRs), have been widely used as molecular markers in population genetics for reconstructing family relationships, including sibships. Microsatellites can be described as polymorphic simple sequence repeats genotyped in DNA (typically one to six base pairs). Because microsatellites often present high levels of inter- and intra-specific polymorphism, they are used to detect variation among individuals and populations in a particular segment of DNA [116].

---

[1]The chapter is part of two published papers [37, 42] and one submitted manuscript [44] in collaboration with Wanpracha Art Chaovalitwongse, Tanya Y. Berger-Wolf, Bhaskar DasGupta, and Mary V. Ashley.

2.1.1 Literature Reviews

Over the past decade, several computational methods have been developed for the sibling reconstruction problem from microsatellites. Those methods can be categorized into statistical and combinatorial approaches. The main principle of statistical approaches is based on the inference of possible sibling groups that can be derived from pairwise or group-based likelihood [113, 127, 24, 130, 33, 86, 135, 136]. Because such approaches estimate the likelihood of all possible pairs or group partitions, they all are very time consuming. Although the reconstructed solutions from those approaches are fairly accurate when dealing with genetic data that are error-free, almost all datasets contain genotyping errors or missing data. In a recent study, [135] proposes a new group-likelihood method that tolerates genotyping errors in genetic markers. There are also a few studies integrating optimization with statistical approaches such as graph-based approaches [13, 24, 12] and a simulated annealing approach [11]. Until recently, combinational optimization approaches have been developed with some degree of success [21, 22, 36]. The main principle of combinatorial approaches is based on the complex combinatorial constraints derived from the Mendel's laws. Those approaches enumerate all possible sibling groups, and solve the sibling reconstruction problem as a set covering problem. To overcome the inefficiency of complete enumeration, a greedy approach is proposed to solve an integrated optimization model subject to the combinatorial constraints of the Mendel's laws [37]. In addition, a related topic, called haplotype reconstruction, has been studied using combinatorial optimization approaches [53, 95, 93].

Although both statistical and combinatorial approaches individually have made great strides in extracting biological knowledge from microsatellite data, they are still faced with several challenges including computational complexity and inaccurate reconstruction due to incomplete or erroneous genetic data [78, 136]. Statistical approaches need to quantify comprehensive inference of all possible combinations of sibling groups from the entire population. This procedure is computationally expensive and the computational time grows drastically as the data set is expanded. Combinatorial approaches

also suffer from the need to enumerate good sibling group candidates [21, 22, 36] while optimizing an artificial objective based on the parsimony assumption, which often traps optimization algorithms in local solutions [37]. In addition, genotyping errors and missing data in sampled genetic markers are other key challenges. Statistical approaches attempt to infer the relatedness of individuals using similarity measures between genetic markers from individual pairs as the main objective while those genetic markers are susceptible to genotyping errors [136]. On the other hand, combinatorial approaches restrict themselves to hard combinatorial constraints of sibling groups based on the Mendel's laws [37]. Thus, even a few genotyping errors or missing data may make an actual sibling group violate those constraints, and the true sibling groups will not be considered as feasible groups.

2.1.2 Contributions

In this study, we first present a mathematical programming model to construct and assign individuals into sibling groups that satisfy the complex combinatorial constraints derived from Mendel's laws. The model is provably a true presentation of the sibling reconstruction problem under a parsimony assumption. We propose a new heuristic approach based on a well-known approximation algorithm to solve this model as we find that this model is a very large-scale mixed-integer programming model.

Tn order to efficiently provide a more accurate solution to the sibling reconstruction problem, we develop a new computational approach that incorporates both combinatorial and statistical concepts in a single optimization model. Specifically, based on a branch-and-price framework, we formulate a set covering problem to minimize the number of reconstructed sibling groups while using the column generation technique to generate high-quality sibling group candidates. As high-quality sibling groups are generated in the subproblem, we propose mixed-integer linear (and nonlinear) programming formulations to generate a sibling group with the maximum likelihood (statistical similarity measure) subject to the combinatorial constraints derived from Mendel's laws.

In addition, we show that the sibling reconstruction problem can be presented as

a special case of the well-known capacitated clustering problem (CCP). We propose a new heuristic optimization algorithm, which has similar concept to a greedy randomized adaptive search procedure (GRASP) [56] that integrates the combinatorial constraints and the concept of parsimony with a statistical similarity measure. The proposed framework involves the following phases: the construction of clusters and the enhancement of quality of clusters. In the first phase, an efficient greedy approach, proposed by [37], is employed repeatedly to construct a number of different possible partitions of (disjoint) sibling groups by introducing a randomized perturbation. Subsequently, among all possible partitions of sibling groups, a set covering problem (SCP) is solved to select the minimum set of sibling groups to cover the population. In the second phase, we propose a new two-stage local search with a memory function to improve the quality of sibling reconstruction based on the similarity of individuals in the sibling groups. Finally, a SCP is solved again to find the minimum number of sibling groups.

2.1.3 Organization

The structure of this chapter is organized as follows. In Section 2.2, the biological background for the sibling reconstruction problem introduced. From genetic data, the functions of combinatorial implications (constraints) of Mendel's laws and statistical similarity measure are described in detail. In Section 2.3, a new mathematical program based on the 2-allele constraints and further a complete mathematical program integrated with the similarity measure function for the sibling reconstruction problem are presented. In Section 4.2.3.1, the background of real biological data sets used in this study is described. The simulated data sets are generated and used for the demonstration of the capability of the proposed approaches on larger and complex data sets. In Section 4.2.3.3, a widely used metrics to evaluate the reconstruction accuracy is presented. In Section 2.6, a greedy heuristic approach is proposed and the results are presented by testing on the real biological data sets and the synthetic data sets. In Section 2.7, a column generation framework with a branch-and-price approach is proposed to efficiently solve the sibling reconstruction problem. In Section 2.8, we describe how the sibling reconstruction problem can be solved as a capacitated clustering problem

by a proposed heuristic optimization algorithm. We conclude this chapter in Section 2.9.

2.2 Biological Background and Problem Definition

In this section, we introduce basic biological background from genetic data for the sibling reconstruction problem.

2.2.1 Basics of Genetic data

Microsatellites are repeating sequences of DNA, for example, $(AGC)_n$ or $(GT)_n$, where n is the length of repeated tandems. A different number of repeated tandems defines a distinct pattern of variable DNA sequences, which is called an *allele*. There are many microsatellite locations (called *loci*) that can be genotyped on the chromosome. There are often many alleles present at a microsatellite locus, which make them fully informative within pedigrees. Figure 2.1 illustrates a schematic example of microsatellites sampled from chromosome pairs of a cohort of individuals. From the figure, we assume that two microsatellite loci are genotyped from each individual. At locus 1, the two tandem repeats, $(CA)_2$ and $(CA)_3$, are encoded as alleles #1 and #2, respectively. At locus 2, the two tandem repeats, $(GA)_3$ and $(GA)_4$, are encoded as alleles #12 and #13, respectively. In diploid organisms, the genotype is determined by two homologous copies of each chromosome and two alleles. At each locus, *homozygous* alleles denote a pair of *identical* alleles, and *heterozygous* alleles denote a pair of *different* alleles. For example, *shrimp b* is heterozygous due to two distinct alleles #2 and #3 at locus 1 and homozygous due to a single allele #12 at locus 2.

In order to mathematically model microsatellites, we present the alleles in a multi-dimensional matrix form that encodes the allele information. We first define the following sets that will be used throughout this chapter: I is a set of individuals, L is a set of loci, and K_l is a set of alleles at locus $l \in L$. We define the matrix entry $a^l_{ik} \in \{0, 1, 2\}$ of individual $i \in I$ at locus $l \in L$, where $a^l_{ik} = 1$ when distinct allele $k \in K_l$ is present, $a^l_{ik} = 2$ when homozygous allele $k \in K_l$ is present, and 0 if allele k is not present.

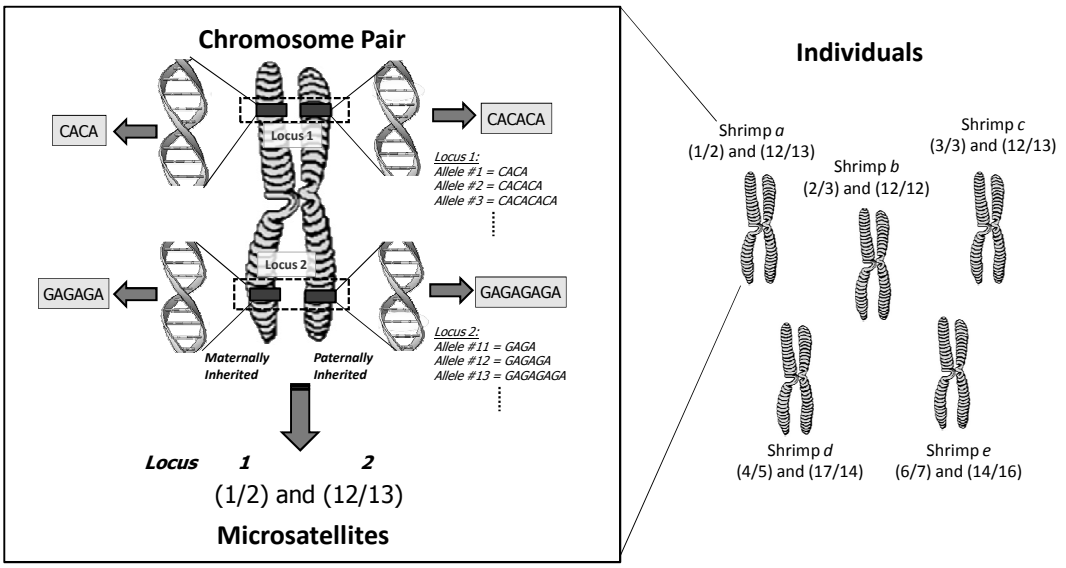

Figure 2.1: An illustration of microsatellites genotyped at two loci from a chromosome pair of diploid individuals. Individual genotypes are defined by a pair of co-dominant alleles at each locus.

Figure 2.2 illustrates an example of how to encode the allele information from a cohort of five individuals with two loci. For example, $a^2_{d,14} = 1$ indicates that *shrimp d* has a distinct allele #14 at locus 2, while $a^1_{c,3} = 2$ indicates that *shrimp c* has homozygous alleles #3 at locus 1. We note the case where two allele pairs (#1/#3) and (#3/#1), having the same alleles located at different sides, present the same genotype.

2.2.2 Combinatorial Implications of Mendel's Laws

To describe the genetical inheritance in diploid organisms, Mendel's laws (or Mendelian inheritance laws) [30, 107] laid down simple rules: *an offspring inherits one allele from each of its parents at each locus* and *the inheritance pattern of alleles at one locus is independent of the other loci*. A *sibling group* is defined as a set of individuals (i.e., siblings) that share common alleles from the same parents at each locus. Based on these rules, [21, 36] first introduced the *4-allele condition*, which is a necessary (but not sufficient) condition to ensure the sibling construction to be genetically consistent. Specifically, the 4-allele condition for any given valid sibling group constrains that *the number of distinct alleles at each locus is less than or equal to four*. Subsequently,

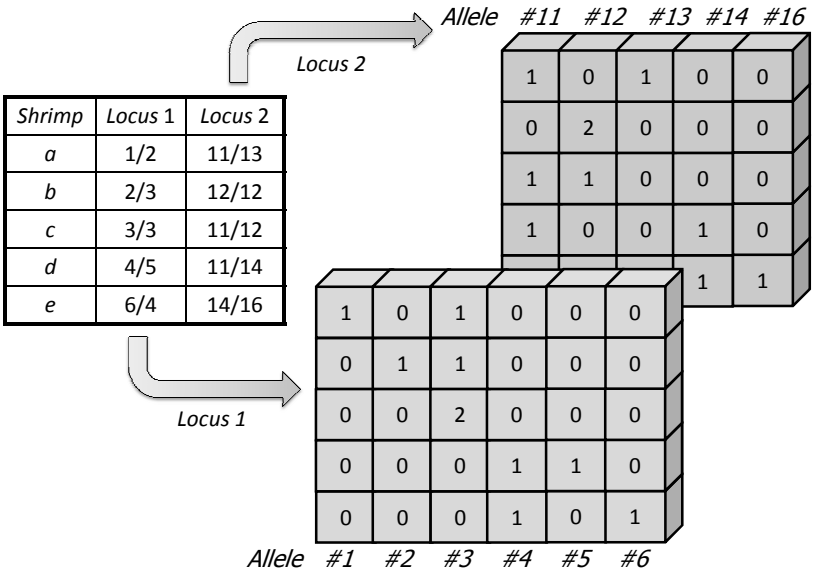

Figure 2.2: A multidimensional matrix presents microsatellites from a cohort of *shrimps* sampled at two loci. Note that each allele is represented by a number and same numbers represents the same alleles.

[22] proposed the *2-allele condition*, which is tighter and more restricted than the 4-allele condition. Specifically, the 2-allele condition for any given valid sibling group constrains that (1) *the number of distinct alleles plus the number of homozygous alleles at each locus is less than four*, and (2) *each allele cannot appear together with more than two other alleles at each locus.* [37] derived the mathematical constraints of the 2-allele condition by using the multi-dimensional matrix a^l_{ik} of microsatellites, which are expressed as follows:

Definition 1 (*2-allele constraints*)**.** *A sibling group of individuals $S \subseteq I$ satisfies the 2-allele condition if and only if they satisfies the following constraints:*

(a) at any locus $l \in L$, the sum of the numbers of distinct alleles and homozygous alleles is less than or equal to 4, i.e., $|\bigcup_{i\in S}\{k^1_{il}\}| + |\bigcup_{i\in S}\{k^2_{il}\}| \leq 4$, where k^1_{il} is an allele such that $a^l_{ik} \neq 0$ and k^2_{il} is an homozygous allele such that $a^l_{ik} = 2$, and

(b) at any locus $l \in L$, each allele k cannot appear together with more than two other alleles (excluding itself), i.e., $|\bigcup_{i\in S}\bigcup_{k'\in K_l\setminus k} k_{il} : \{k, k'\} \in K_l| \leq 2$.

For illustration, we shall show how the 2-allele constraints are applied to the reconstruction of sibling groups with the population shown in Figure 2.2. Shrimps a and b can be considered a valid sibling group because they satisfied both constraints *(a)* and *(b)*. Specifically, the group $\{a, b\}$ has distinct alleles $\{1, 2, 3\}$ at locus 1, and distinct alleles $\{11, 12, 13\}$ and homozygous alleles $\{12\}$ at locus 2. The sum of the numbers of distinct alleles and homozygous alleles is less than or equal to 4 for both loci 1 and 2. At both loci, each allele appears with at most two others alleles. On the other hand, shrimps b, c, and d are not a valid sibling group. The group $\{b, c, d\}$ fails to satisfy constraint *(a)* because the sum of the numbers of distinct alleles $\{2, 3, 4, 5\}$ and homozygous alleles $\{3\}$ exceeds 4 at locus 1 although both constraints *(a)* and *(b)* are satisfied at locus 2.

2.2.3 Similarity Measure from Genetic Data

Assume there are no typing errors or missing data, the similarity likelihood from genetic data can provide the direct inference of the sibling relationships when we do not have parentage information in advance. Here we propose a pairwise approach to score the similarities between the genotypes of all individual pairs. We define the similarity score $q^l_{ii'}$ in Equation (2.66) to describe the similarity degree of two individuals i and $i' \in I$ from the distance between both genetic markers, $|a^l_{ik} - a^l_{i'k}|$, for each locus $l \in L$.

$$q^l_{ii'} = \begin{cases} 1 & \text{if } \sum_{k \in K} |a^l_{ik} - a^l_{i'k}| = 0; \\ 0.5 & \text{if } \sum_{k \in K} |a^l_{ik} - a^l_{i'k}| = 2; \\ 0 & \text{if } \sum_{k \in K} |a^l_{ik} - a^l_{i'k}| = 4. \end{cases} \tag{2.1}$$

If two individuals share both common alleles at a locus, then $q^l_{ii'} = 1$, i.e., they are said to be identical. If two individuals share only one common allele at a locus, then $q^l_{ii'} = 0.5$. If two individuals share no common alleles at a locus, then $q^l_{ii'} = 0$. Subsequently, with the similarity score for each locus, we can determine the pairwise similarity score

for an individual pair over all loci, which is computed by

$$q_{ii'} = \sum_{l \in L} q^l_{ii'}. \tag{2.2}$$

The higher the degree, the more similar two individuals. Furthermore, we are able to obtain the total score of individuals in a sibling group $j \in J$, where J is a set of sibling groups. The group similarity is computed by

$$q^j = \sum_{(i,i') \in S} q_{ii'} \quad \forall j \in J. \tag{2.3}$$

Compare groups $\{a, b, c\}$, and $\{b, c, d\}$ in Figure 2.2. From their group similarity scores 2 and 1, we rather select the sibling group $\{a, b, c\}$ with higher similarity score for the reconstruction when both satisfy the 2allele constraints.

In most studies of natural populations, it is more practical to sample genetic data from a cohort of individuals without parentage information. Reconstructing sibling groups would mostly rely on the similarity likelihood from genetic data of individuals, while all sibling groups satisfy the 2-allele constraints. For this, in the next section, we propose a complete mathematical formulation that combines both combinatorial and statistical functions to find the sibling reconstruction with the objective of the minimal set of sibling groups as there is no real objective defined properly.

2.3 Mathematical Programs for the Sibling Reconstruction Problem

In this section, we present two mathematical optimization models for the sibling reconstruction problem using the biological concepts of the combinatorial constraints and the similarity measure.

We define the following notations that are used thorough this chapter. Consider a set of individuals $i \in I$ presented by a set of loci $i \in L$ of alleles $k \in K_l$, where $K_1 \cup K_2 ... \cup \ K_{|L|} = K$, in a set of sibling groups $j \in J$. The decision variables are defined as follows:

- $z_j \in \{0,1\}$: indicate if any individual is selected to be a member of sibling group j;
- $x_{ij} \in \{0,1\}$: indicate if individual i is selected to be a member of sibling group j;
- $y^l_{jk} \in \{0,1,2\}$: indicate if any members in sibling group j has distinct ($y^l_{jk} = 1$) or homozygous ($y^l_{jk} = 2$) allele(s) k at locus l;
- $v^l_{jkk'} \in \{0,1\}$: indicate if allele k appears with allele k' in sibling group j at locus l.

2.3.1 2-Allele Optimization Model

The first optimization model, called 2-allele optimization model (2AOM), is to find a minimum set of sibling groups subject to the 2-allele constraints alone. A mix-integer linear program is given by.

$$
\begin{aligned}
(\text{2AOM}) \quad \min \quad & \sum_{j\in J} z_j && (2.4)\\
\text{s.t.} \quad & \sum_{j\in J} x_{ij} \le z_j \quad \forall\, i \in I,\ j \in J && (2.5)\\
& \sum_{j\in J} x_{ij} \ge 1 \quad \forall\, i \in I, && (2.6)\\
& \sum_{i\in I_j} a^l_{ik} z_j \le y^l_{jk} \quad \forall\, j \in J,\ k \in K,\ l \in L, && (2.7)\\
& \sum_{k\in K} y^l_{jk} \le 4 \quad \forall\, j \in J,\ l \in L, && (2.8)\\
& \sum_{i\in I} a^l_{ik} a^l_{ik'} z_j \le M v^l_{jkk'} \quad \forall\, j \in J,\ k \in K,\ k' \in K\backslash k,\ l \in L, && (2.9)\\
& \sum_{k'\in K\backslash k} v^l_{jkk'} \le 2 \quad \forall\, j \in J,\ k \in K,\ l \in L, && (2.10)\\
& z_j,\ x_{ij},\ v^l_{jkk'} \in \{0,1\};\ y^l_{jk} \in \{0,1,2\} \\
& \qquad \forall\, j \in J,\ k \text{ and } k' \in K,\ l \in L. && (2.11)
\end{aligned}
$$

The objective in Equation (2.4) is to minimize the total number of sibling groups. Equation (2.5) ensures that the binary sibling group variables must be activated for

the assignment of any individual i to sibling group j. Equation (2.6) ensures that every individual i has to be assigned to at least one sibling group j. Constraint set in Equation (2.7) ensures that a group j is activated as the integer variable y_{jk}^{l} for distinct or homozygous indication must be activated for the existence of distinct or homozygous allele(s) at locus l in sibling group j. Constraint set in Equation (2.8) ensures that in a group j, the total number of distinct and homozygous alleles is not greater than 4. We relate these two constraint sets in Equation (2.7)-(2.8) to (a) in Definition 1. Constraint set in Equation (2.9) restricts that allele pair k and k' must be activated when individual i is assigned to group j. The big M is a large positive number, defined by $M = |I| + 1$. Constraint set in Equation (2.10) ensures that every allele in a sibling group does not appear with more than two other alleles (excluding itself). We relate these two constraint sets in Equation (2.9)-(2.10) to (b) in Definition 1.

The 2AOM problem is considered to be a generalization of the well-known set covering problem with additional constraints such as the 2-allele constraints. For computational complexity, the total number of discrete variables is $O(\max(|J|*|K|*|L|, |I|*|J|))$, and the total number of constraints is $O(|J|*|K|^2*|L|)$. It is easy to see that the 2AOM problem is a very large-scale complex problem and may not be easy to solve for large data sets.

An implementation issue is noted here. Solving the 2AOM problem requires an initialization in terms of the total number of sibling groups. If the initial number of sibling groups is too small, the problem will become infeasible. If the initial number of sibling groups is too large, we will have to introduce much more binary variables then we need to. The proposed heuristic approach discussed next can also be used to initialize the number of sibling groups as its solution can be theoretically shown to be an upper bound of the 2AOM problem.

2.3.2 Integrated 2-Allele Optimization Model with Similarity Measure

In the 2AOM, only the combinatorial constraints and the concept of parsimony, which is to minimize the number of sibling groups, were considered in the model. More importantly, statistical similarity measure from genetic features of individuals can provide direct information to benefit the sibling relationships, while the combinatorial constraints give the robustness of reconstructing sibling groups. Therefore, we further incorporate the similarity measure in the model. We slightly modify the sibling group set. Assume there is a completely enumerated set of sibling groups $j \in J$ and an assignment matrix is known as $\delta_{ij} \in \{0,1\}$ to indicate that individual $i \in I_j$ is assigned in group j. A modified mixed-integer linear program is given by

$$
\begin{aligned}
(\text{I2AOM}) \quad \min \quad & \sum_{j\in J}(1-\theta q^j)z_j && (2.12)\\
\text{s.t.} \quad & \sum_{j\in J} z_j \geq 1 \quad \forall\, i \in I, && (2.13)\\
& \sum_{i\in I_j} a^l_{ik}\delta_{ij}z_j \leq y^l_{jk} \quad \forall\, j \in J,\ k \in K,\ l \in L, && (2.14)\\
& \sum_{k\in K} y^l_{jk} \leq 4 \quad \forall\, j \in J,\ l \in L, && (2.15)\\
& \sum_{i\in I} a^l_{ik}a^l_{ik'}\delta_{ij}z_j \leq M v^l_{jkk'} \\
& \qquad \forall\, j \in J,\ k \in K,\ k' \in K\backslash k,\ l \in L, && (2.16)\\
& \sum_{k'\in K\backslash k} v^l_{jkk'} \leq 2 \quad \forall\, j \in J,\ k \in K,\ l \in L, && (2.17)\\
& z_j,\ v^l_{jkk'} \in \{0,1\};\ y^l_{jk} \in \{0,1,2\} \\
& \qquad \forall\, j \in J,\ k \text{ and } k' \in K,\ l \in L. && (2.18)
\end{aligned}
$$

The objective in Equation (2.12) includes the minimization of sibling groups and the maximization of similarity degrees of individuals in the same groups, where we introduce a parameter θ balancing two different scales. Equation (2.13) ensures that every individual i has to be assigned to at least one group j. Constraint set in Equation

(2.14) ensures that a group j is activated as the integer variable y_{jk}^{l} for distinct or homozygous indication must be activated for the existence of distinct or homozygous allele(s) at locus l in sibling group j. Constraint set in Equation (2.15) ensures that in a group j, the total number of distinct and homozygous alleles is not greater than 4. We relate these two constraint sets in Equation (2.14)-(2.15) to (a) in Definition 1. Constraint set in Equation (2.16) restricts that allele pair k and k' must be activated when individual i is assigned to group j. The big M in Equation (2.16) is a large positive number defined by $M = |I| + 1$. Constraint set in Equation (2.17) ensures that every allele in the group does not appear with more than two other alleles (excluding itself). We relate these two constraint sets in Equation (2.16)-(2.17) to (b) in Definition 1.

2.4 Description of Data Sets

In this section, we describe the real biological data sets and simulated data sets, used for testing the performances of our proposed approaches.

2.4.1 Real Data Sets

The real biological data sets are considered the benchmark data sets widely used in the literature because the true sibling relationships (ground truth) are known. The background of the data sets is described as follows:

Salmon: The Atlantic salmon *Salmo salar* data set comes from the genetic improvement program of the Atlantic Salmon Federation [76]. We use a truncated sample of microsatellite genotypes of 250 individuals from 5 families with 4 loci per individual. The data does not have missing alleles at any locus. This data set is a subset of one of the samples of genotyped individuals used in [13]. There are 2.66% alleles missing in the data set.

Radish: The wild radish *Raphanus raphanistrum* data set [45] consists of samples from 150 radishes from two families with 5 loci and 5 alleles per locus. There are 37 missing alleles among all the loci.

Shrimp: The tiger shrimp *Penaeus monodon* data set [80] consists of 59 individuals from 13 families with 7 loci. There are 8 pairs of missing alleles.

Fly: The *Scaptodrosophila hibisci* data set [139] consists of 190 individuals in the same generation from 6 families sampled at various number of loci with up to 8 alleles per locus. All individuals shared 2 sampled loci which were chosen for our study. Around 39% of the alleles are missing in the data set.

Ant: The *Leptothorax acervorum* data set [72] are haplodiploid species. This data set is a subset of one of the samples used in [135], which consists of 377 worker diploid ants. There are 9% of the alleles missing in the data set.

Turtle: Kemp's ridleys sea turtle data set, *Lepidochelys Kempi*, is polyandrous and sampled from 26 mothers and offspring groups at 3 loci [82]. There are 16.38% of the alleles missing in the data set. The other data set is a subset obtained from the original sampled data by eliminating most violated and indefinite sibling groups. There are still 12.12% missing alleles.

Characteristics of the data sets are summarized in Table 4.1. The numbers of individuals, actual sibling groups, loci, and different types of alleles genotyped at each locus, missing values in data are, in turn, reported. Based on our preliminary analysis of the genotypes, except salmon and turtle, there are no violations of the 2-allele condition (2-allele constraints) in the data sets of shrimp, fly, and ant. There are missing alleles in the data sets except salmon, especially fly and turtle data sets that contain larger portions. In our study, we do not leave them out and treat the missing alleles as a wild card that can represent any alleles in comparison with the others. Thus, a lower bound (the worst case) for the reconstruction is guaranteed.

2.4.2 Simulated Data Sets

To create a set of simulation data, we developed a random population generator, which works as follows. The generator first constructs a number of adults (parents) with the full genetic information. Based on this information, a single generation of sibling data was generated and the parentage information was retained so that the true sibling

Table 2.1: Characteristics of biological data sets

Species	No. of individuals	No. of groups	No. of loci	No. of types of alleles per locus	Missing alleles (%)
Salmon	351	6	4	(9, 11, 9, 7)	0.00
Radish	531	2	5	(3, 2, 4, 4, 2)	3.99
Shrimp	59	13	7	(20, 18, 12, 7, 23, 9, 16)	2.66
Fly	190	6	2	(7, 7)	37.89
Ant	377	10	6	(22, 16, 15, 3, 5, 8)	9.00
Turtle	175	26	3	(5, 13, 10)	16.38
Turtle-m[a]	55	9	3	(5, 9, 8)	12.12

[a] Turtle-m is the subset of turtle without most indefinite sibling groups.

groups are known. The sibling problem generator requires the following parameters: M is the number of adult males; F is the number of adult females; l is the number of sampled loci; a is the number of alleles per locus; j is the number of juveniles in the population per one adult female; o is the maximum number of offsprings per parent couple. Although the random problem generator is rather simplistic, it is consistent with the genetics of known parents and provides a baseline for the accuracy of the algorithm. The procedure of our random generator can be described in detail as follows:

Step 1. First, we generated the parent population of M males and F females with parents with l loci, each having a distinct alleles per locus.

Step 2. After the parents were generated, we created a population of their offsprings by randomly selecting j pairs of parents. A male and a female were chosen independently and uniformly at random from the parent population.

Step 3. For each of the chosen parent pairs, we generated a specified number of offsprings, o, each randomly receiving one allele from its mother and one from its father at each locus.

This population generator is a rather simplistic approach; however, it is consistent with the genetics of known parents and provides a baseline for testing the performance of the any solution approaches. To produce a simulated data set used in this study, we varied the parameters of the population generator as follows:

- The number of adult females (F) and the number of adult males (M) are set to 10, 30;
- The number of sampled loci (l) is set to 2, 3, 4, 6, 10;

- The number of alleles per locus (a) is set to 2, 5, 10, 20;
- The number of families (j) is 1, 2, 5, 10;
- The maximum number of offsprings per couple (o) is set to 2, 5, 10, 40, 50.

For each parameter setting, we obtained a set of offspring population with known parent pairs. In each population, there are $o \times j$ individuals in j known sibling groups.

2.5 Evaluation and Assessment

In our study, the ultimate goal of SRP is the accurate reconstruction of sibling relationships. However, in SRP, the objective of minimizing the number of sibling groups and maximizing the similarity likelihood is made for easy implementation, but cannot be used as a real objective for assessing the reconstruction results. Moveover, the real relationships (ground truth) are specifically known in all test data sets, so reconstruction accuracy is among the most used to evaluate the performance by measuring the percentage of individuals correctly assigned to the sibling groups in comparison with the actual sibling groups. The reconstruction accuracy can be calculated by quantifying the *error rate* from the minimum partition distance [63], which is equal to (1 - *error rate*). The minimum distance is equivalent to a maximum assignment linear problem (MALP) and known to be a maximum bipartite weighted matching problem. The MALP can be formulated as follows. Given two (non-) disjoint sets of sibling groups $A = \{a_1, a_2, ..., a_m\}$ and $B = \{b_1, b_2, ..., b_n\}$. We denote a $m \times n$ cost matrix C, where c_{ij} is the cost of assigning group a_i to b_j, which is the number of individuals correctly assigned. We define a binary decision variable: $x_{ij} = 1$ if group a_i is assigned to group b_j, and 0 otherwise. The mathematical programming formulation is given by

$$\text{(MALP)} \quad \max \quad \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \tag{2.19}$$

$$s.t. \quad \sum_{i \in I} x_{ij} \leq 1 \quad \forall\, j \in J, \tag{2.20}$$

$$\sum_{j \in J} x_{ij} \leq 1 \quad \forall\, i \in I, \tag{2.21}$$

$$x_{ij} \in \{0, 1\} \quad \forall\, i \in I,\ j \in J. \tag{2.22}$$

The objective in Equation (2.19) is to maximize the cost of assigning groups in A to groups in B. The constraints in Equations (2.20) and (2.21) ensure that each group in B (A, respectively) is assigned to at most one group in A (B, respectively). Note that the solution to MALP can be represented as the minimum number of individuals to be removed from the resulting sibling groups (i.e., *error rate*) so as to be identical to the actual sibling groups.

2.6 2AOM by Heuristic Approach: Iterative Maximum Covering Set

2.6.1 Iterative Maximum Covering Set

In the 2AOM problem, the parsimony assumption to minimize the number of sibling groups may not give the most accurate sibling reconstruction, which is the real objective of our sibling reconstruction problem. In addition, we can only say, that the optimal solution to 2AOM (the number of sibling groups) is biologically a true lower-bound of the real sibling groups. Therefore, to solve the 2AOM problem more efficiently, we herein propose a heuristic approach, namely *Iterative Maximum Covering Set* (IMCS), which is an iterative optimization approach motivated by the standard approximation algorithm of the set covering problem, i.e., a maximum coverage approach. The idea behind this approach is to construct one sibling group maximizing the individual cover in each iteration. Essentially, in each iteration, we solve a reduced problem of 2AOM. The objective of IMCS is to maximize the total number of individuals to be covered by a sibling group, which satisfies the 2-allele constraints. The IMCS problem can be formally defined as follows. We define the following decision variables:

- $x_i \in \{0, 1\}$: indicate if individual i is selected to be a member of the current sibling group;
- $y_k^l \in \{0, 1, 2\}$: indicate if any members in the current group has distinct ($y_{jk}^l = 1$) or homozygous ($y_{jk}^l = 2$) allele(s) k at locus l;
- $v_{kk'}^l \in \{0, 1\}$: indicate if allele k appears with allele k' in the current sibling group at locus l.

The mathematical formulation of IMCS problem is given by

$$
\begin{aligned}
(\mathsf{IMCS}) \quad \max \quad & \sum_{i \in I} x_i && (2.23)\\
\text{s.t.} \quad & a_{ik}^l x_i \le y_k^l \quad \forall\, i \in I,\ k \in K,\ l \in L, && (2.24)\\
& \sum_{k \in K} y_k^l \le 4 \quad \forall\, l \in L, && (2.25)\\
& \sum_{i \in I} a_{ik}^l a_{ik'}^l x_i \le M v_{kk'}^l \quad \forall\, k \in K,\ k' \in K \backslash k,\ l \in L, && (2.26)\\
& \sum_{k' \in K \backslash k} v_{kk'}^l \le 2 \quad \forall\, k \in K,\ l \in L, && (2.27)\\
& x_i,\ v_{kk'}^l \in \{0,1\};\ y_k^l \in \{0,1,2\} \\
& \qquad \forall\, i \in I,\ k \text{ and } k' \in K,\ l \in L. && (2.28)
\end{aligned}
$$

The objective in Equation (2.23) is to maximize the total number of individuals assigned to the current sibling group. Similarly, Constraint sets in Equations (2.24)-(2.27) ensure that all individuals assigned in current sibling group must satisfy the 2-allele constraints. The big M is a large positive number defined by $M = |I| + 1$.

The heuristic approach is to iteratively solve the IMCS problem. In each iteration, the solution of the IMCS problem is a set of individuals assigned to the current sibling group. Then we remove the assigned individual from the set I and repeat the procedure until all individuals are assigned. Note that the IMCS approach is viewed as a assignment problem where every individual belongs to only one sibling group, while the 2AOM problem is solved for non-disjoint sibling groups. The IMCS approach is fast and scalable for very large-scale sibling reconstruction problem because the problem size is significantly reduced as we remove the possible groups of larger individual assignment.

2.6.2 Reconstruction Results of 2AOM and IMCS

We present the reconstruction results of the 2AOM and IMCS approaches on real and simulated data sets. All programs were coded in MATLAB with synchronization of CPLEX version 10.0 in GAMS on the platform of an Intel Xeon Quad Core 3.0GHz processor workstation with 8 GB RAM memory. The computational times reported

Table 2.2: Performance characteristics of the 2AOM and IMCS approaches on real biological data sets.

		2AOM[a]				IMCS		
Species	Actual no. of groups	No. of groups	Accuracy (%)	Gap (%) in CPLEX	Time[b] (sec.)	No. of groups	Accuracy (%)	Time[b] (sec.)
Salmon	6	8	94.02	63	> 72000	7	98.29	130
Radish	2	3	51.98	0	75	3	52.54	26
Shrimp	13	14	96.61	67	> 72000	13	100.00	150
Fly	6	7	67.72	55	> 72000	8	53.80	23
Ant[c]	10	-	-	-	> 72000	11	93.10	506
Turtle	9	9	47.27	60	> 72000	8	61.82	12

[a] The initial number of sibling group is 30.
[b] Computational time limit is set to be 20 hours.
[c] No feasible solution to 2AOM was found within time limit.

were obtained from the desktop's internal timing calculations, which include time used for preprocessing and postprocessing. The computational time for each instance is 20 hours liimit.

Table 2.2 reports the accuracies and computational times for the real data sets. It is seen that the IMCS approach obtains the optimal solutions in all instances, whereas solving 2AOM problem directly obtains the optimal solution only for the radish data set. Specifically, solving 2AOM in CPLEX failed to obtain the optimal solution within 20 hour time limit for the salmon, shrimp, fly, ant, turtle data sets. Note that the reported results (i.e., the numbers of sibling groups) were based on the best integer feasible solutions.

We further show the capability of the 2AOM and IMCS by testing various simulated data sets generated by the population generator. We also compare the accuracies of both approaches to a set covering approach M4SCP proposed in [36]. The M4SCP approach is a similar combinatorial approach that involves enumerating all possible sibling groups and solving a set covering problem to find a minimum set of sibling groups among them. In Table 2.3, the results are reported for different parameter settings. For each instance, we fix one parameter at a time. We observed that the IMCS approach outperforms the 2AOM and M4SCP approaches on average. We note that all instances were solved to optimality except an instance with the setting of $l = 10$, $a = 10$, $j = 10$, and $o = 10$. In addition, in most cases, the 2AOM approaches could not obtain the optimal solutions except the instance with setting of $a = 2$ and $j = 2$.

Table 2.3: Accuracies of the 2AOM and IMCS approaches compared to the M4SCP approach [36] from simulated data sets.

Parameter	2AOM		IMCS		M4SCP	
Settings	Accuracy	Time	Accuracy	Time	Accuracy	Time
l=2	59.25%	2273.04	57.61%	2.28	54.18%	0.26
l=4	63.94%	2754.80	66.53%	8.28	52.71%	0.21
l=6	64.28%	3005.49	71.44%	28.96	54.78%	0.19
l=10	60.56%	3078.93	71.89%	239.21	55.28%	0.19
a=2	26.67%	0.56	26.67%	0.21	36.98%	0.16
a=5	69.42%	3679.45	72.19%	30.54	58.34%	0.16
a=10	71.81%	3699.62	81.83%	225.17	60.71%	0.39
a=20	80.14%	3732.64	86.78%	22.81	60.91%	0.19
j=2	76.67%	1.50	78.13%	0.72	62.88%	0.02
j=5	64.63%	3079.56	64.58%	3.65	49.56%	0.11
j=10	44.73%	5253.14	57.90%	204.68	34.00%	0.75
o=2	49.48%	1711.83	54.38%	2.67	18.19%	0.22
o=5	69.46%	3250.27	69.83%	14.41	36.66%	0.27
o=10	67.08%	3372.10	76.40%	191.97	53.98%	0.22

2.7 Column Generation Framework with a Branch-and-Price

In this section, we propose a column generation framework with a branch-and-price as an alternative to solve the sibling reconstruction problem.

Recall that the 2AOM and I2AOM are a generalization of the set covering problem and has been shown to be strongly NP- hard [15, 37]. The main challenge of solving set covering problems lies in the enumeration of all possible sets. In our case, explicitly enumerating the complete set of valid sibling groups is intractable and impractical. Our group has also shown that it is very hard to approximate as well [15]. In addition, one will have to fine tune and optimize the balancing parameter θ. For these reasons, we develop a column generation approach to efficiently generate high-quality (more probable) sibling groups. Here we use the terms *sibling group* and *column* interchangeably when mentioned throughout this section.

In our column generation approach, the master problem (MP) is formulated as a mini- mum set covering problem, and the pricing subproblem (SP) is formulated as a generalized knapsack problem where sibling groups that satisfy the 2-allele constraints and maximize the similarity scores are constructed. We solve the linear programming (LP) relaxation of the restricted master problem (RMP) with a limited set of valid sibling groups. A set of (optimal) dual variables is produced corresponding to the

constraint set of individual $i \in I$ and is passed to the pricing subproblem as a guide for generating sibling groups. The task of the subproblem (SP) is to price out improving sibling groups with respect to the dual variables. In particular, some of sibling groups have individuals hardly being grouped together with other individuals. New sibling groups are added into the RMP after checking the optimality. The RMP is updated and resolved. The procedure is iteratively performed until there is no new sibling groups to improve the solution, which implies that the current LP solution to the master problem is optimal. Note that we need to solve the original RMP for an integer programming (IP) solution (i.e., a final set of sibling groups) at the end of column generation iterations. Moreover, a branch rule is introduced to the column generation at each node in the branch and bound search to find a proven optimal solution.

2.7.1 Restricted Master Problem: Set Covering Model

Given a limited set $J_s \subset J$ of valid sibling groups, the RMP for the sibling reconstruction is formulated as a minimum set covering problem (Min-SCP) as follows. We define the binary variable: $\varepsilon_i = 1$ if any individual i is not assigned to any group and 0 otherwise. We denote an assignment matrix δ_{ij}, where $i \in I$ and $j \in J_s$. Each column δ_j of the assignment matrix represents a valid sibling group such that $\delta_{ij} = 1$ if individual i is assigned to the group j and 0 otherwise.

$$\text{(Min-SCP)} \quad \min \quad \sum_{j \in J_s} z_j + C \sum_{i \in I} \varepsilon_i \tag{2.29}$$

$$\text{s.t.} \quad \sum_{j \in J_s} \delta_{ij} z_j + \varepsilon_i \geq 1 \quad \forall\, i \in I, \tag{2.30}$$

$$z_j,\ \varepsilon_i \in \{0, 1\}, \quad \forall\, i \in I,\ j \in J_s. \tag{2.31}$$

The objective in Equation (2.29) is to minimize the number of sibling groups plus the penalty cost of individuals not assigned. Constraint set in Equation (2.30) ensures every individual has to belong to at least one sibling group. If an individual i is not assigned to any sibling groups, then penalized in the objective function with the cost $C = |I| + 1$. Equation (2.31) fixes binary variables.

It is noted that LP relaxation of Min-SCP is solved during column generation iterations and the IP formulation of Min-SCP is solved to obtain the exact IP solution at the last iteration since some columns are left out of the LP (i.e., fractional solutions). Next, we present the pricing subproblem to find new sibling groups to improve the objective function value of Min-SCP.

2.7.2 Subproblem: Generating Valid Sibling Groups

For the optimal reconstruction, we attempt to construct a sibling group with (maximum) negative reduced cost, which is able to improve the solution to the RMP. We obtain the optimal dual variables π_i associated to the constraint set in Equation (2.30). The reduced cost for a new sibling group j is computed by

$$c_j = 1 - \sum_{i \in I} \pi_i \delta_{ij} \quad \forall j \in J. \tag{2.32}$$

The new sibling group j is assumed not identical to any groups in the existing set J_s. Note that individuals need not be considered in computation if the associated dual variables are zero or extremely small numbers. Subsequently, we check the optimality condition:

$$\bar{c} := \min\{c_j = 1 - \sum_{i \in I} \pi_i \delta_{ij} \mid j \in J\}. \tag{2.33}$$

If $\bar{c} \geq 0$, no improving sibling groups are eligible to add into the RMP, which states that the optimal solution is found. Otherwise, any sibling group j with negative reduced cost is added into the RMP.

2.7.2.1 Weighted Maximization Problem

To construct a sibling group from the reduced cost in Equation (2.32), we propose a weighted maximization problem (WMP), which uses the dual variables π_i as weight coefficients of individuals. We define the decision variables as follows. $x_i = 1$ if individual i is selected to be a member of current sibling group and 0 otherwise. $y_k^l = 1$ if any

individual in current sibling group has distinct allele k at locus l, $y_k^l = 2$ if any member in current sibling group has homozygous allele k at locus l, and 0 otherwise. $v_{kk'}^l = 1$ if allele k appears with allele k' at locus l in current sibling group, and 0 otherwise. The mathematical programming formulation of WMP is given by

$$
\begin{aligned}
(\mathsf{WMP}) \quad \max \quad & \sum_{i \in I} \pi_i x_i && (2.34)\\
\text{s.t.} \quad & a_{ik}^l x_i \leq y_k^l \quad \forall\, i \in I,\ k \in K,\ l \in L, && (2.35)\\
& \sum_{k \in K} y_k^l \leq 4 \quad \forall\, l \in L, && (2.36)\\
& \sum_{i \in I} a_{ik}^l a_{ik'}^l x_i \leq M v_{kk'}^l \quad \forall\, k \in K,\ k' \in K \backslash k,\ l \in L, && (2.37)\\
& \sum_{k' \in K \backslash k} v_{kk'}^l \leq 2 \quad \forall\, k \in K,\ l \in L, && (2.38)\\
& x_i,\ v_{kk'}^l \in \{0,1\};\ y_k^l \in \{0,1,2\} \quad \forall\, i \in I,\ k \text{ and } k' \in K,\ l \in L. && (2.39)
\end{aligned}
$$

The objective in Equation (2.34) is to maximize the weighted sum of individuals assigned to the current sibling group. Similarly, Constraint sets in Equations (2.35)-(2.38) ensure that all individuals assigned in current sibling group must satisfy the 2-allele constraints. The big M is a large positive number defined by $M = |I| + 1$. Equation (2.39) fixes decision variables. The constructed sibling group is associated to a new column δ_j, where $j \in J$, that could be added to the RMP as a possible sibling group.

2.7.2.2 Similarity Maximization Problem

In the WMP, we only consider a sibling group constructed subject to 2-allele constraints. To integrate the genetic data, we further propose a similarity maximization problem (SMP) with the similarity measure function as follows. We first define a $|I| \times |I|$ symmetric matrix $\bar{Q} = \pi^T Q \pi$, where the vector π is the dual variables and $Q = (q_{ii'})$ is a $|I| \times |I|$ symmetric matrix, whose each element represents the pairwise similarity measure as computed in Equation (2.2).It is important to note that the SMP is a

nonlinear program. The mathematical programming formulation of SMP is given by

$$\text{(SMP)} \quad \max \quad \sum_{i\in I}\sum_{i'\in I} \bar{q}_{ii'} x_i x_{i'} \tag{2.40}$$

$$\text{s.t.} \quad a^l_{ik} x_i \le y^l_k \quad \forall\, i \in I,\ k \in K,\ l \in L, \tag{2.41}$$

$$\sum_{k\in K} y^l_k \le 4 \quad \forall\, l \in L, \tag{2.42}$$

$$\sum_{i\in I} a^l_{ik} a^l_{ik'} x_i \le M v^l_{kk'} \quad \forall\, k \in K,\ k' \in K\backslash k,\ l \in L, \tag{2.43}$$

$$\sum_{k'\in K\backslash k} v^l_{kk'} \le 2 \quad \forall\, k \in K,\ l \in L, \tag{2.44}$$

$$x_i,\ v^l_{kk'} \in \{0,1\};\ y^l_k \in \{0,1,2\} \quad \forall\, i \in I,\ k \text{ and } k' \in K,\ l \in L. \tag{2.45}$$

The only difference between WMP and SMP lies in the objective function. The objective function of WMP is linear whereas the objective function of SMP is quadratic. The quadratic objective in Equation (2.40) is to maximize the total similarity score of individuals with non-zero dual variables assigned in the current sibling group. Similarly, constraint sets in Equations (2.41)-(2.44) ensure that all individuals assigned in the current sibling group must satisfy the 2-allele constraints. To solve the quadratic SMP, we employ a linearization technique proposed in [38] to reformulate the quadratic program as a mixed integer linear program. We define $s_i \ge 0$ as the total pairwise similarity score for individual i and $r_i \ge 0$ as a surplus variable.The linearized mathematical

formulation of SMP is given by

$$\text{(L-SMP)}\max \quad \sum_{i\in I} s_i \tag{2.46}$$

$$\text{s.t.} \quad \sum_{i'\in I\setminus i} q_{ii'}x_{i'} - r_i - s_i = 0 \quad \forall\, i \in I, \tag{2.47}$$

$$s_i \le M_2 x_i \quad \forall\, i \in I. \tag{2.48}$$

$$a^l_{ik}x_i \le y^l_k \quad \forall\, i \in I,\ k \in K,\ l \in L, \tag{2.49}$$

$$\sum_{k\in K} y^l_k \le 4 \quad \forall\, l \in L, \tag{2.50}$$

$$\sum_{i\in I} a^l_{ik}a^l_{ik'}x_i \le M v^l_{kk'} \quad \forall\, k \in K,\ k' \in K\setminus k,\ l \in L, \tag{2.51}$$

$$\sum_{k'\in K\setminus k} v^l_{kk'} \le 2 \quad \forall\, k \in K,\ l \in L, \tag{2.52}$$

$$x_i,\ v^l_{kk'} \in \{0,1\};\ y^l_k \in \{0,1,2\};\ s_i,\ r_i \ge 0 \quad \forall\, i \in I,\ k \text{ and } k' \in K,\ l \in L \tag{2.53}$$

The objective in Equations (2.46) is still to maximize the total similarity score of individuals with non-zero dual variables assigned to the current sibling group. Constraint set in Equation (2.47) is to calculate the total pairwise similarity score of individuals grouped with individual i, excluding itself. Constraint set in Equation (2.48) ensures individual i is activated to select. The big M_2 is a large positive number. which can be set to $M = \sum_{i,i'\in I} ||q_{ii'}||$. Constraint sets in Equations (2.49)-(2.52) ensure that all individuals assigned in the current sibling group must satisfy the 2-allele constraints.

2.7.2.3 Greedy Generation Procedures

According to [132] and [19], solving the SP in the column generation is computationally intensive. However, in practice, it is not necessary to select the only column with the highest reduced cost i.e., any column with a negative reduced cost can be a good candidate. In addition, it is extremely hard to generate a sibling group containing all individuals with non-zero dual variables at one time due to the 2-allele constraints. To overcome these challenges, we develop two greedy sibling group generation procedures, and combine them with the WMP or SMP. The key idea of these greedy procedures is

to generate multiple "good" columns in each iteration of the column generation such that the overall number of iterations is reduced.

The first procedure, a greedy set partitioning procedure (GSPP), iteratively generates disjoint sibling groups by solving the WMP (or SMP) to obtain a group after removing individuals already assigned in the previous iteration. The procedure continues until all individuals are assigned. In this procedure, the number of individuals in a group decreases with the iterations. In the latter iterations, there may be one or a few individuals that are hard to assign to the same groups. The second iterative procedure, a greedy set covering procedure (GSCP), generates possibly non-disjoint sibling groups. We use individuals with non-zero dual variables as base individuals. In every iteration, we solve the WMP (or SMP) to obtain a group that contains other individuals with a selected base individual. in contrast with GSPP, we do not remove any individuals being assigned previously after solving the WMP (or SMP) in every iteration. We note that GSCP generally requires more computational time than GSPP because the number of iterations of the GSCP is only fixed with the size of individuals with non-zero dual variables.

2.7.3 Branching Rule

In the branch-and-bound search, we consider a branching rule (called a branch-on follow-on rule) [120, 19]. We determine, on the one hand, two individuals belong to the same sibling group and, on the other hand, to different sibling groups. In other words, on the one (right) branch, a binding rule $B(x_i, x_{i'})$ is defined that a sibling group is considered when it contains individuals x_i and $x_{i'}$; On the other (left) branch, a releasing rule $R(x_i, x_{i'})$ is defined that a sibling group is forbidden when it contains individuals x_i and $x_{i'}$. After determining the branch rule, at each node, the Min-SCP in the MP and the WMP (or SMP) in the SP need to be modified by adding a new constraint. For the Min-SCP, on the right branch, we add $z_j \leq 0$ for the binding rule so that the sibling group j containing individuals x_i and x_i' is forbidden to select, whereas, on the left branch, we do nothing for releasing rule because of $z_j \leq 1$ originally. For

the WMP (or SMP), on the right branch, we add a bound constraint for the binding rule that is given by

$$x_i - x'_i = 0, \tag{2.54}$$

and, on the left branch, for the releasing rule that is given by

$$x_i + x'_i < 1. \tag{2.55}$$

Usually, the determination of a beneficial branch rule (i.e., a pair of individuals) at a node would reduce the branch-and-bound search procedure. We here consider a pair selection that determines a pair of two individuals x_i and $x_{i'}$ having the highest probability of being together among all groups based on the LP solution to the RMP. We define $p(x_i, x_{i'})$ as a probability of x_i and $x_{i'}$ being together. The pair selection rule is given by

$$h(x_i, x_{i'}) = \arg \max_{(i,i') \in I} \{p(x_i, x_{i'}) = \frac{\sum_{j \mid i \text{ and } i' \in I_j} z_j}{\sum_{j \mid i \text{ or } i' \in I_j} z_j}\}. \tag{2.56}$$

For example, we assume there are three sibling groups $g_1 = \{1, 2, 3, 4\}$, $g_2 = \{1, 2, 3\}$ and $g_3 = \{3, 4\}$ with LP solutions $lp_1 = 0.5$, $lp_2 = 0.5$, and $lp_3 = 0.5$. As a consequence, we have several options for the pair x_i and $x_{i'}$, such as (x_1, x_2), (x_2, x_3), (x_3, x_4) and so on. By using the rule in Equation (2.56), between two pairs (x_1, x_2) with the probability $p(x_1, x_2) = (0.5 + 0.5) \ / \ (0.5 + 0.5) = 1$ and (x_2, x_3) with $p(x_2, x_3) = (0.5 + 0.5) \ / \ (0.5 + 0.5 + 0.5) = 2/3$, we determine the pair (x_1, x_2) to branch. Note that if the pair has been selected at previous nodes, we skip to the next highest one and so on.

According to the determined branching rules at each node, additional constraints to the master problem and the subproblem would leave a number of variables from consideration. We only eliminate the infeasible columns in the master problem and avoid to generate columns forbidden in the subproblem. However, in the branch-and-bound search, the problem structures in column generation procedure at each node remain unchanged.

2.7.4 Implementation Settings

Before we present the implementation results, there are some specific settings in our experiments needed to address ahead when applying a column generation framework to the large-scale combinatorial problems such as SRP. First, the column generation procedure starts with an initial (valid) solution. To obtain a good initial solution that can reduce the number of iterations of procedure, we directly employ the proposed greedy approaches GSPP and GSCP to generate a set of sibling groups. The solution is shown to be good enough and can provide a lower bound for comparison with final solutions. Secondly, it has been mentioned that there may be very similar (degenerate) solutions for large-scale problems [100]. To prevent the degeneracy, a perturbation is introduced and may result in different dual variables so as to obtain different combinations of solutions (e.g., sibling groups). We here introduce a perturbation to the coefficients in the objective function of Min-SCP. We rewrite the Equation (2.29) to become $\sum_{j\in J_s}\sigma_j z_j + C\sum_{i\in I}\varepsilon_i$, where σ_j is a random variable uniformly ranging between $[1-\epsilon, 1+\epsilon]$ and ϵ is a small positive number. The perturbation is only executed during the iterations where there is no improvement on the objective function value. Thirdly, poor convergence towards final optimal solution (tailing-off effect) frequently appears in implementing column generation for large-scale problems [19, 100]. In our experiments, we propose the following termination criteria to prevent a possible long tail. (1) The optimality condition in Equation (3.26) is met. We also consider the situation, i.e., $c_j = 0$, where different sibling groups may be priced out after a certain number of column generation iterations. (2) There is no improvement in the objective function value of the RMP after a maximum number of iterations (e.g., 50). Within the period, a maximum number of perturbation iterations is also considered (e.g., 10). (3) In every node in the branch-and-bound search, the computational time $T_{run}(L)$, where L is the level of a node in the search tree, starts with $T_{run}(0) = 5$ hours at root node, decreases geometrically by $T_{run}(L) = (1/2)^{(L-1)}T_{run}(0)$, and becomes 0.25 hour by $\max\{0.25, T_{run}(L)\}$ in a long run. The depth-first strategy is applied for favorable improvement. The procedure terminates when whichever criterion is reached first within a computational time limit of 20 hours.

Table 2.4: Configuration of experiments.

	Approach in subproblem	
Subproblem	GSPP	GSCP
WMP	GSPP-WMP	GSCP-WMP
SMP	GSPP-SMP	GSCP-SMP
HYBRID	GSPP-HYBRID	GSCP-HYBRID

As formulated in column generation, the (dual) SP is mathematically associated to the (primal) MP and the objective of SP is usually formulated based on the reduced cost derived from the dual variables in the MP. Columns are then generated in the SP and checked with the optimality according to the associated reduced costs. Although the proposed SMP in the SP can generate "good" columns according to our experiences, the quadratic objective function in Equation (2.40) of SMP is not directly derived from the reduced cost in Equation (2.32) and not intuitively associated to the MP. The eligibility of generated columns still cannot be guaranteed by checking the optimality condition directly. Therefore, we propose a hybrid approach consisting of the WMP and the SMP. In every column generation iteration, the SMP is first solved to generate high-quality columns and some of columns with negative reduced costs are possibly added in the RMP. If the optimality condition is met, we then solve the WMP with the same dual variables to make sure if the optimality is met again.

Experiments of performing all the proposed approaches are summarized in Table 2.4. Programs were coded in MATLAB with synchronization of CPLEX version 10.0 in GAMS on the platform of an Intel Xeon Quad Core 3.0GHz processor workstation with 8 GB RAM memory. The computational times reported were obtained from the desktop's internal timing calculations, which include time used for preprocessing and postprocessing. Note that the LP relaxation solution to the RMP in each column generation iteration was obtained by using the barrier LP solver in CPLEX in order to reduce the heading-in and the tailing-off effects.

2.7.5 Reconstruction Results

The results of our proposed approaches with the above-mentioned implementation settings for all instances are presented in Table 2.5. The LP and IP solutions, accuracies, computational times, and total numbers of new columns added in the column generation procedure at the root node are reported in the first part (on the left). The total numbers of visited nodes and computational times of the branch-and-bound search are reported in the second part (on the right). Overall, when the column generation is carried out along the root node of branch-and-bound search, we observe that very good solutions are obtained in most instances in terms of the IP solutions and accuracies in comparison with actual solutions (ground truth). Particularly, we obtain the best solution among all instances from GSCP-SMP and obtain the exact sibling reconstruction with 100% accuracy in some instances of the shrimp and ant data sets. As for the computational efficiency, there are relatively small quantities of columns (sibling groups) generated to the RMP of Min-SCP when compared to a brute enumeration. All experiments are terminated based on the preset stopping criteria within about 5 hours. In addition, it is noted that an instance of implementing GSCP-WMP on the fly data set is shown to be optimal. However, the sibling reconstruction is not 100% accurate. It might be caused by high percentage of missing alleles in the data set.

Because the solutions in most instances are not proven to be optimal in the column generation, we further employed a branch-and-price approach with the proposed branching rule. The stopping criteria are set as follows: (1) IP solution obtained at current (descendent) node is close to the LP solution from the root node (the global lower bound); (2) LP solution obtained at current (descendent) node is larger than the best IP solution on record; (3) there are no new columns generated at current (descendent) node; and (4) the total computational time is limited to 20 hours. As seen in the second part of Table 2.5, the nodes are pruned only after one branching (i.e. 3 nodes visited in total) in most instances based on the above stopping criteria. For the instances (especially the turtle data set) where the branch-and-bound search is not pruned within 20 hours, we also report the comparison results from the root node and

best node on record in Table 2.6. It shows that already obtained solutions at the root node are good enough compared to the IP and LP solutions reliably provided from the best node.

The effectiveness of incorporating the similarity measure are proved in our experiments. For clarity and briefness, we only plot a representative (the ant set) of all instances by the behaviors of IP/LP solutions and accuracies obtained by implementing GSCP-WMP (on the top) and GSCP-SMP (on the bottom) in Figure 2.3. Compared to GSCP-WMP, GSCP-SMP with the similarity measure results in relatively stable and better solutions fast in a short period of iterations.

In addition, we here remark two causes of undesirable reconstruction. As mentioned previously, the resultant reconstruction is susceptible to missing alleles in data sets. In our study, however, we do not leave them aside by using a wild card that can represent any alleles when compared to the other alleles in a group. When constructing in a group, the worst-case reconstruction is then promised. Surprisingly, we obtain a good reconstruction with higher accuracy for the fly data sets (shown in Table 2.10) even if there are a lot of missing alleles. In addition, we found the violations of the 2-allele constraints appearing in the actual data sets. This indeed causes the wrong base comparison although the 2-allele constraints is robust to accurate reconstruction. For instance, 98.29% is the best accuracy we obtained for the salmon data set. However, after reassignment of wrong sibling groups manually, we obtain the exact reconstruction result with 100% accuracy.

2.7.6 Comparison with Existing Approaches

Next, we compared the reconstruction solutions obtained by our proposed approaches with other existing methods, including 2AOM and IMCS [37], BMG [22], A&F [13], B&M [24], KINGROUP [86], and COLONY [135]. The 2AOM is a simple version of SRP without considering the similarity measure function and solved by the greedy heuristic approach (IMCS). Both 2AOM and IMCS approaches generated sibling groups only based on the 2-allele constraints. Their experimental results showed that the 2AOM could not be

Table 2.5: The results of our proposed approaches on real biological data sets. The first part (on the left) reports the characterization of results obtained from the last column generation iteration at root node. The second part (on the right) reports the numbers of visited nodes and computational times of the branch-and-bound search. The best results among all experiments are highlighted in bold-face in terms of the numbers of IP solution and accuracy. The total computational time is limited to 20 hours.

		Column Generation at root node						Branch-and-Price	
Configuration	Species	No. of groups	LP Solution	IP Solution	Accuracy (%)	Time (sec.)	No. of new columns	No. of nodes visited	Time (sec.)
GSCP-WMP	Salmon	6	7.00	7	98.29	807	140	3	4900
	Shrimp	13	13.00	13	100	3539	392	3	11966
	Fly	6	5.78*	7	75.26	11413	2558	3	11624
	Ant	10	10.00	10	99.73	3890	543	3	13958
	Turtle	26	15.61	17	47.43	18079	2716	43	>72000
	Turtle-m	9	6.75	7	67.27	602	615	3	2298
GSPP-WMP	Salmon	6	7.00	7	98.29	366	79	3	1622
	Shrimp	13	13.00	13	100	3147	163	3	15129
	Fly	6	7.50	8	47.89	94	64	66	>72000
	Ant	10	11.00	11	93.1	1496	109	3	5017
	Turtle	26	15.55	18	54.29	18026	1150	43	>72000
	Turtle-m	9	7.14	8	76.36	293	123	264	>72000
GSCP-SMP	Salmon	6	7.00	**7**	**98.29**	2091	3	3	8382
	Shrimp	13	13.00	**13**	**100**	1262	2	3	4367
	Fly	6	7.00	**7**	**84.74**	649	93	3	2648
	Ant	10	10.00	**10**	**100**	2998	12	3	14667
	Turtle	26	30.00	**30**	**70.29**	5212	245	3	16105
	Turtle-m	9	10.00	**10**	**83.64**	689	36	3	2111
GSPP-SMP	Salmon	6	7.00	7	98.01	11860	19	3	29516
	Shrimp	13	13.00	13	100	4420	1	3	12657
	Fly	6	6.40	7	69.47	890	66	3	2084
	Ant	10	10.00	10	97.61	15558	58	3	34597
	Turtle	26	16.19	18	53.14	18728	185	32	>72000
	Turtle-m	9	6.80	7	69.09	357	43	3	1460
GSCP-HYBRID	Salmon	6	7.00	7	98.01	3477	13	3	>72000
	Shrimp	13	13.00	13	100.00	1549	15	3	4946
	Fly	6	7.00	7	75.26	650	83	3	2588
	Ant	10	10.00	10	100.00	4849	23	3	18098
	Turtle	26	30.00	30	70.29	5475	236	3	16434
	Turtle-m	9	10.00	10	81.82	369	30	3	1163
GSPP-HYBRID	Salmon	6	7.00	7	98.29	11565	26	3	30274
	Shrimp	13	13.00	13	100	6275	12	3	20009
	Fly	6	6.40	7	66.84	1125	63	3	3395
	Ant	10	10.00	10	98.41	15999	56	3	34982
	Turtle	26	16.19	18	53.14	18739	185	32	>72000
	Turtle-m	9	6.80	7	67.27	781	57	3	1944

* The optimal solution is proved.

Table 2.6: Comparison results of the fly, turtle, and turtle-m data sets obtained from the root node and the best node on record in the branch-and-bound search.

		Root node			Best node		
Configuration	Species	LP	IP	Accuracy	LP	IP	Accuracy
GSCP-WMP	Turtle	15.61	17	47.43	15.55	17	53.71
GSPP-WMP	Fly	7.50	8	47.89	5.50	7	82.11
GSPP-WMP	Turtle-m	7.14	8	76.36	7.00	7	74.55
GSPP-WMP	Turtle	15.55	18	54.29	15.53	17	48.57
GSPP-SMP	Turtle	16.19	18	53.14	16.18	18	55.43
GSPP-HYBRID	Turtle	16.19	18	53.14	16.18	18	55.43

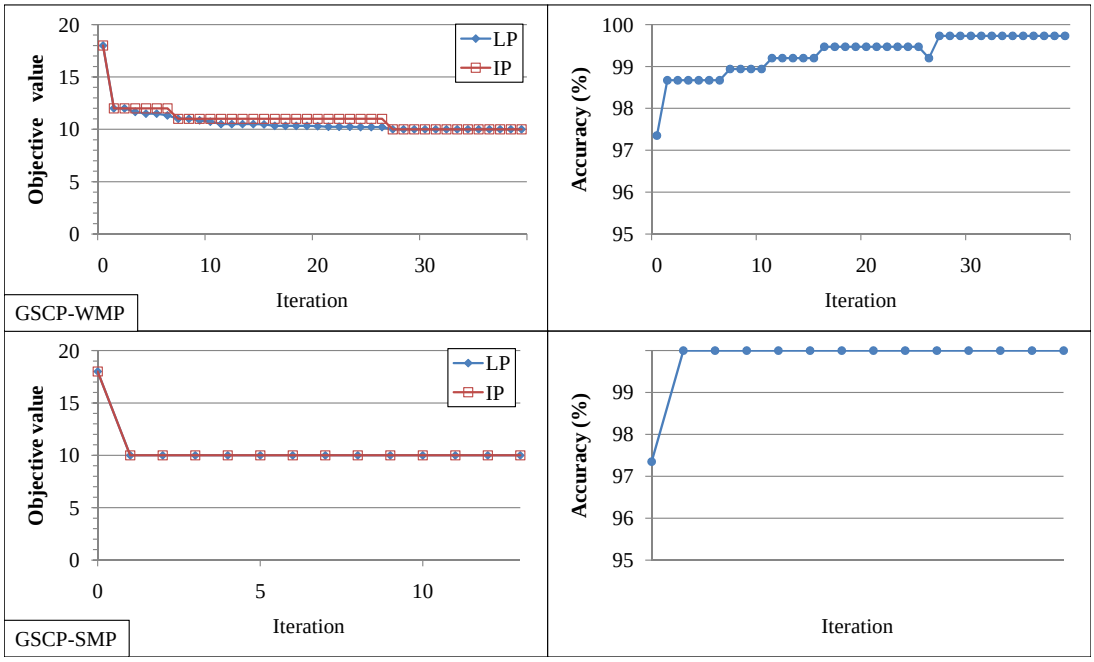

Figure 2.3: Display of the behaviors of the objective values (IP/LP solutions) and accuracies over the column generation iterations by performing the GSCP-WMP (on the top) and GSCP-SMP (on the bottom) for the ant data set.

solved in CPLEX to find an optimal solution within 20 hours. There are the gaps between IP and LP solutions such as 63%, 67%, and 57% for the salmon, shrimp, and fly data sets, and it failed to obtain a feasible solution for the ant data set. On the other hand, although the IMCS could result in better accuracies for all the data sets in a relatively short run when compared to the 2AOM, the optimal solution still could not be guaranteed. The BMG algorithm in is a 2-allele set construction version of the set covering model proposed by [36]. The procedure includes enumerating all maximal sibling groups subject to the 2-allele condition and then solve a set covering problem to find a minimum set of sibling groups. Without considering computational complexity, this approach could guarantee the best reconstruction (see accuracy shown in Table 2.10 except the turtle data set). The A&F algorithm is a combinatorial approach to exhaustively enumerate all possible sibling groups satisfying the 2-allele condition (although the authors did not explicitly state the condition) and to obtain a maximal, not necessarily optimal, collection of sibling groups. The B&M algorithm is an approach based on a mixture of likelihood and combinatorial techniques to construct a graph

with individuals as the nodes and the edges weighted by the pairwise likelihood (relatedness) ratio. The algorithm identifies potential sibling groups by finding the connected components in the graph. The KINGROUP (KG) algorithm is an approach based on the likelihood estimates of partitions of individuals into sibling groups by comparing, for every individual, the likelihood of being part of any existing sibling group with the likelihood of starting its own group. The COLONY approach uses the maximum likelihood method to assign sibship and parentage jointly.

In Table 2.10, we only report among the best reconstruction accuracy from our experiments in comparison with the above-mentioned combinatorial and statistical approaches. Because these approaches were performed on different computing platforms, the computational times are not reported here. From the results, we observed that the proposed approach outperforms the other approaches. Compared to the combinatorial approaches such as 2AOM, IMCS, BMG, A&F, and B&M, our approach combining the similarity measure function with the 2-allele constraints can give a better reconstruction although the 2-allele constraints have provided a robust base for sibling reconstruction. On the other hand, compared to the statistical approaches KG and COLONY, our approach still leads to a very competitive reconstruction. It is worth noting that our approach does not generate all possible sibling groups, whereas most of listed approaches are based on an enumeration. Computational complexity increases drastically as the data size increases. For instance, the results of A&F on most instances except the shrimp and fly data sets were not obtained due to computational resource limitations. In addition, for the data sets (e.g., the fly and turtle data sets) with large percentages of missing alleles, our approach obviously can handle the uncertainty from missing data to achieve higher reconstruction accuracy.

2.8 Randomized Greedy Optimization Algorithm for Capacitated Clustering Model

The capacitated clustering problem (CCP) has been one of the most challenging problems in clustering research. Several variants of CCP have been studied in the literature

Table 2.7: Recovery values of true full sibling groups (accuracy) when comparing our method with other existing approaches in five different species.

Species	GSCP-SMP[a]	2AOM	IMCS	BWG	A&F	B&M	KG	COLONY
Salmon	98.29	94.02	98.29	98.29	–[b]	98.29	94.60	56.70
Shrimp	100.00	96.61	100.00	100.00	67.80	100.00	77.97	100.00
Fly	84.74	66.84	47.37	100.00	31.05	19.62	54.73	–[d]
Ant	100.00	–[c]	93.10	100.00	–[d]	97.61	97.10	100.00
Turtle	70.29	–[c]	40.00	48.00	–[d]	38.18	39.40	40.00
Turtle-m	83.64	47.27	61.82	–[d]	–[d]	–[d]	–[d]	–[d]

[a] We report the best accuracy among all experiments.
[b] A&F ran out of 4GB memory as it enumerates all possible sibling groups.
[c] There are no results acquired within computational time limit.
[d] There are no results available.

including a capacitated centred clustering problem (CCCP) as well as a capacitated p-median problem (CPMP). The CCP can be formally defined as follows. Given a set of data points with associated weights (or features), the CCP is to partition the data points into clusters such that the total weight of data points in each cluster does not exceed the capacity limit of the cluster. In general, the objective of CCP is to maximize the homogeneity (similarity) of the data points in each cluster or to maximize the separation (dissimilarity) among different clusters [74]. Although clustering techniques have been essential tools to solve many practical problems, previous studies on the CCP are mostly applied to facility location problems and they often focus on the development of solution algorithms.

In this section, we present the sibling reconstruction problem to be formulated as a special version of the CCP. We propose a new heuristic optimization algorithm, which has similar concept to a greedy randomized adaptive search procedure (GRASP) [56], that integrates the combinatorial constraints and the concept of parsimony with a statistical similarity measure. The proposed framework involves the following phases: the construction of clusters and the enhancement of quality of clusters. In the first phase, an efficient greedy approach IMCS is employed repeatedly to construct a number of different possible partitions of (disjoint) sibling groups by introducing a randomized perturbation. Subsequently, among all possible partitions of sibling groups, a set covering problem (SCP) is solved to select the minimum set of sibling groups to cover the

population. In the second phase, we propose a new two-stage local search with a memory function to improve the quality of sibling reconstruction based on the similarity of individuals in the sibling groups. Finally, a SCP is solved again to find the minimum number of sibling groups.

2.8.1 Capacitated Clustering Problem

The mathematical model of the CCP was first proposed by [109] and its variants were used to study several practical problems in diverse applications. Here we consider one of the most common variants of CCP. Given a set of data points $i \in I$ with associated positive weights π_i and resources c_i, and a set of edges $(i, i') \in E$ with associated positive weights (e.g., similarities) $w_{ii'}$, where $i \neq i'$. Assume that there is a set of clusters $j \in J$ used to cover (represent) all data points. Let p be a predefined number of clusters. There is a resource limitation W_j on each cluster j. The objective of CCP is to find a set of clusters with the maximum weight (or similarity) per cluster subject to a resource capacity.

Define x_{ij} and z_j as binary variables, where $x_{ij} = 1$ if data point i is assigned to cluster j, and $x_{ij} = 0$ otherwise; $z_j = 1$ if cluster j is selected, and $z_j = 0$ otherwise. The formulation of CCP is given in Equations (2.57)-(2.63). The objective in Equation (2.57) is to maximize the total weight of all selected clusters. The constraint set in Equation (2.58) calculates the total weight of data points assigned to cluster j. The constraint set in Equation (2.59) ensures that every data point is assigned to one cluster, while the constraint set in Equation (2.60) guarantees that a cluster must be selected if there is any data point assigned to it. The constraint set in Equation (2.61) ensures that only p clusters are selected. The constraint set in Equation (2.62) is a knapsack constraint ensuring that the total resource of data points assigned to a cluster does not violate its capacity.

$$\text{(CCP)} \quad \max \quad \sum_{j \in J} W_j z_j \tag{2.57}$$

$$\text{s.t.} \quad W_j = \sum_{i \in I} \pi_i x_{ij} + \sum_{(i,\, i') \in E} w_{ii'} x_{ij} x_{i'j} \quad \forall\, j \in J \tag{2.58}$$

$$\sum_{i \in I} x_{ij} \;=\; 1 \quad \forall\, j \in J \tag{2.59}$$

$$x_{ij} \;\leq\; z_j \quad \forall\, i \in I,\; j \in J \tag{2.60}$$

$$\sum_{j \in J} z_j \;=\; p \tag{2.61}$$

$$\sum_{i \in I} c_i x_{ij} \;\leq\; C_j \quad \forall\, j \in J \tag{2.62}$$

$$x_{ij},\; z_j \;\in\; \{0, 1\}. \tag{2.63}$$

In the literature, exact solution methods have been proposed to solve different versions of CCP. [106] used a column generation with a specialized branching technique and solved a maximum weighted cluster problem (MWCP) in the subproblem. [18] presented a new exact algorithm by modeling the capacity location problem as a set partitioning problem with cluster-feasibility constraints. [99] proposed an approach that integrates the column generation and Lagrangean/surrogate relaxation techniques to solve capacitated p-median problems. More recently, [34] proposed a computational framework based on column generation and branch-and-price approaches to solve the capacitated network problems. Due to the computational complexity of real-life CCPs, a large number of heuristic approaches have been developed. Those include classical sub-gradient heuristics [109, 87], simulated annealing and tabu search [58, 111], bionomic approach [104], cluster search [41], GRASP-based algorithms [124, 50], and other heuristics [112, 123, 110, 17].

2.8.2 Capacitated Clustering Model for Sibling Reconstruction Problem

2.8.2.1 Capacitated Clustering Model

We formulate the SRP as a CCP by using the statistical likelihood measure as the objective function subject to the Mendelian combinatorial constraints. We note that this is the first mathematical model that integrates both statistical and combinatorial concepts to reconstruct the sibling relationship. We shall mathematically define our integrated problem as follows.

Given a set of individuals $i \in I$ with associated weights π_i and a set of edges $(i,\ i') \in E$ with associated similarity measures $w^l_{ii'}$ over all loci $l \in L$, where $i \neq i'$. Assume that there is a set of sibling groups $j \in J$ to represent the relationship of the given population. Because there is no prior parental information, the number of sibling groups is not known and will have to be determined by the model. Next we define the following decision variables.

- $z_j \in \{0,\ 1\}$: indicates if there is individual(s) assigned to be a member of sibling group j;
- $x_{ij} \in \{0,\ 1\}$: indicates if individual i is assigned to be a member of sibling group j;
- $y^l_{jk} \in \{0,\ 1,\ 2\}$: indicates if any member in sibling group j has distinct ($y^l_{jk} = 1$) or homozygous ($y^l_{jk} = 2$) allele(s) k at locus l;
- $v^l_{jkk'} \in \{0\ 1\}$: indicates if allele k appears with allele k' in sibling group j at locus l.

Statistical Similarity Measure as Objective Function

The overall objective here is to reconstruct a set of sibling groups such that the total similarity degree and weight of individuals assigned to the selected sibling groups

is maximized. The objective function is given by

$$\max \quad \sum_{j \in J} W_j z_j, \tag{2.64}$$

where W_j is the sum of weight and similarity score for a sibling group j, which can be calculated by

$$W_j = \sum_{i \in I} \pi_i x_{ij} + \sum_{(i,\, i') \in E} (\sum_{i \in L} w^l_{ii'}) x_{ij} x_{ij} \quad \forall\, j \in J. \tag{2.65}$$

The above equation takes into account not only the weights of individuals assigned to the sibling group j but also the pairwise similarity measures over all loci. The weight of each individual can be estimated from the prior information; however, in our case all individuals are equally weighed because of the small sample size. To calculate the pairwise similarity score, we apply a simple pairwise approach to score the similarity based on genetic features at loci between a pair of individuals. The pairwise score can be calculated by

$$w^l_{ii'} := \begin{cases} 1 & \text{if } \sum\limits_{k \in K} |a^l_{ik} - a^l_{i'k}| = 0; \\ 0.5 & \text{if } \sum\limits_{k \in K} |a^l_{ik} - a^l_{i'k}| = 2; \\ 0 & \text{if } \sum\limits_{k \in K} |a^l_{ik} - a^l_{i'k}| = 4. \end{cases} \tag{2.66}$$

The sum of similarity score $\sum\limits_{l \in L} w_{ii'}$ over all loci represents the degree of similarity for a pair of individuals i and i'. The higher the degree, the more similar two individuals.

Capacity Constraints: Combinatorial Rules from Mendel's Laws

The capacity constraints of SRP are more complex than those of simple CCP's because the capacity constraints are multi-dimensional. That is, each capacity constraint must be satisfied for individual independent locus of a sibling group.

In [22], the 4-allele and 2-allele properties were first proposed based on the Mendel's laws. [37] augmented 2-allele property with a tighter constraint. For mathematical representation, we formulate combinatorial constraints from the modified 2-allele property by employing an indication matrix, $a^l_{ik} \in \{0, 1, 2\}$. From the first rule of the Mendel's

laws, the combinatorial constraints are given in Equations (2.67)-(2.68). Equation (2.67) ensures that the integer variable y_{jk}^{l} for distinct or homozygous indication must be activated for the existence of distinct or homozygous allele(s) at locus l in sibling group j. Equation (2.68) ensures that the number of distinct allele and the number of homozygous alleles is less than or equal to four.

$$a_{ik}^{l} x_{ij} \leq y_{jk}^{l} \quad \forall\, j \in J,\ k \in K,\ l \in L, \tag{2.67}$$

$$\sum_{k \in K} y_{jk}^{l} \leq 4 \quad \forall\, j \in J,\ l \in L. \tag{2.68}$$

From the second rule of the Mendel's laws, the combinatorial constraints are given in Equations (2.69)-(2.70). Equation (2.69) restricts that the binary variable for allele pair indication $v_{jkk'}^{l}$ must be activated for any assignment of individual i to sibling group j. Equation (2.70) ensures that every allele in the group does not appear with more than two other alleles (excluding itself). A big M number is defined by $M = |I| + 1$.

$$\sum_{i \in I} a_{ik}^{l} a_{ik'}^{l} x_{ij} \leq M v_{jkk'}^{l} \quad \forall\, j \in J,\ k \in K,\ k' \in K \backslash k,\ l \in L, \tag{2.69}$$

$$\sum_{k' \in K \backslash k} v_{jkk'}^{l} \leq 2 \quad \forall\, j \in J,\ k \in K,\ l \in L. \tag{2.70}$$

For the rest of the section, a so-called "feasible sibling group (or cluster)" is a set of individuals that satisfies the capacity constraints in Equations (2.67)-(2.70) at every locus.

Covering constraints

For certain species in natural populations that do not belong to the monogamous mating system, the overlapping situation where any individual can be assigned to more than one sibling group are commonly seen. We therefore consider the covering constraint set instead of the partitioning constraint set in Equations (2.60)-(2.61). The

covering constraint sets are given by

$$\sum_{i \in I} x_{ij} \geq 1 \quad \forall\, j \in J, \tag{2.71}$$

$$x_{ij} \leq z_j \quad \forall\, i \in I,\ j \in J. \tag{2.72}$$

Equation (2.71) ensures that every individual is assigned to at least one sibling group. Equation (2.72) ensures that the binary sibling group variable must be activated for the assignment of any individual i to sibling group j.

It is noted that because the actual number of sibling groups is not known in general, in this study, we therefore employ the parsimony assumption to find the minimum number of sibling groups instead of using the constraint set in Equation (2.61). For this purpose, sibling group selection can be formulated as a set covering problem (SCP) that incorporates the covering constraints.

2.8.2.2 Preliminaries of Solving CCP for SRP

According to the formulation in the previous subsection, the CCP for SRP can be considered as a complete optimization model (CCP-SRP) shown in Equations (2.73)-(2.78). The objective of CCP-SRP in Equation (2.73) integrates the minimization of sibling groups and the maximization of similarity degrees of individuals in the same sibling groups, where a balancing parameter θ is introduced between the two terms. The constraint sets in Equations (2.74)-(2.78) follow the same definitions described in the previous section.

$$\text{(CCP-SRP)} \quad \max \quad \sum_{j\in J}(\theta W_j - 1)z_j \tag{2.73}$$

$$\text{s.t.} \quad W_j = \sum_{i\in I}\pi_i x_{ij} + \sum_{(i,\,i')\in E}(\sum_{i\in L} w^l_{ii'})x_{ij}x_{ij} \quad \forall\, j \in J \tag{2.74}$$

$$\sum_{i\in I} x_{ij} \geq 1 \quad \forall\, j \in J \tag{2.75}$$

$$x_{ij} \leq z_j \quad \forall\, i \in I,\ j \in J \tag{2.76}$$

$$a^l_{ik}x_{ij} \leq y^l_{jk} \quad \forall\, j \in J,\ k \in K,\ l \in L \tag{2.77}$$

$$\sum_{k\in K} y^l_{jk} \leq 4 \quad \forall\, j \in J,\ l \in L \tag{2.78}$$

$$\sum_{i\in I} a^l_{ik}a^l_{ik'}x_{ij} \leq M v^l_{jkk'} \quad \forall\, j \in J,\ k \in K,\ k' \in K\backslash k,\ l \in L \tag{2.79}$$

$$\sum_{k'\in K\backslash k} v^l_{jkk'} \leq 2 \quad \forall\, j \in J,\ k \in K,\ l \in L. \tag{2.80}$$

The CCP-SRP is a mixed-integer nonlinear programming (MINLP) problem, which is viewed as a generalization of 2AOM. To solve the CCP-SRP, there are issues encountered such as highly computational complexity and the calibration of the parameter θ. Firstly, let us look back on the optimization model 2AOM in [37], which is to find a minimum number of sibling groups subject to capacity constraints and without the integration of statistical similarity measure. The 2AOM has been proved to be an *NP-hard* problem with many discrete variables and many constraints. It is hard to solve directly to obtain an optimal solution. According to our computational experiments, we failed to find a feasible solution to 2AOM in CPLEX after 20 hours of run. Consequently, it is not easy to calibrate the balancing parameter at a precise level, which plays a role in solving the SCP-SRP, when the value of similarity varies with assignments of individuals into different sibling groups. These observations and experiences have motivated us to develop an efficient heuristic method to solve this problem. In the next section, we thus propose a new greedy optimization heuristic to solve the decomposed CCP-SRP model in two phases.

2.8.3 Randomized Greedy Optimization Algorithm

we develop a new randomized greedy optimization algorithm (RGOA) to solve the CCP of SRP. The underlying concept behind the RGOA is motivated by the Greedy Randomized Adaptive Search Procedure (GRASP) [56]. The RGOA is divided into two phases: construction and enhancement phases. Recall that the objective of CCP in Equation (2.64) and its total weight in Equation (2.65) contains two terms, the individual weight and the pairwise similarity, to be maximized. The individual weight of sibling group assignment is maximized in the construction phase while the pairwise similarity is maximized in the enhancement phase.

The flowchart of our RGOA is shown in Figure 2.4 and the associated pseudo-code is presented in Algorithm 1. In the construction phase, we modify an efficient approach, called IMCS, for the SRP [37] by introducing a randomized perturbation on the individual weight. The function of randomized perturbation is added into IMCS to construct diverse, yet high-quality feasible, partitions of (disjoint) sibling groups. A number of diverse partitions of sibling groups are accumulated over a number of iterations in the construction phase, where a parameter max_t is predetermined for limiting the maximum number of iterations. Subsequently, we perform cluster selection by solving a SCP to find the minimum set of sibling groups, which will be an initial solution for the next phase. In the enhancement phase, we propose a new local search with a memory function in two scales, cluster-based and individual-based neighborhoods, to improve the solution quality with respect to the pairwise similarity degree. In order to explore more high-quality solutions, we implement the RGOA procedure repeatedly to obtained a number of (high-quality) elite sets of sibling groups, where a parameter max_r is predetermined for limiting the maximum number of replications. Finally, among all (elite) solutions, the cluster selection is again performed by solving a SCP to obtain the final minimum set of sibling groups.

Construction Phase: Finding good and feasible sibling groups

The goal of the construction phase is to construct high-quality partitions of feasible sibling groups, each maximizing the total weight of individuals assigned to it. In

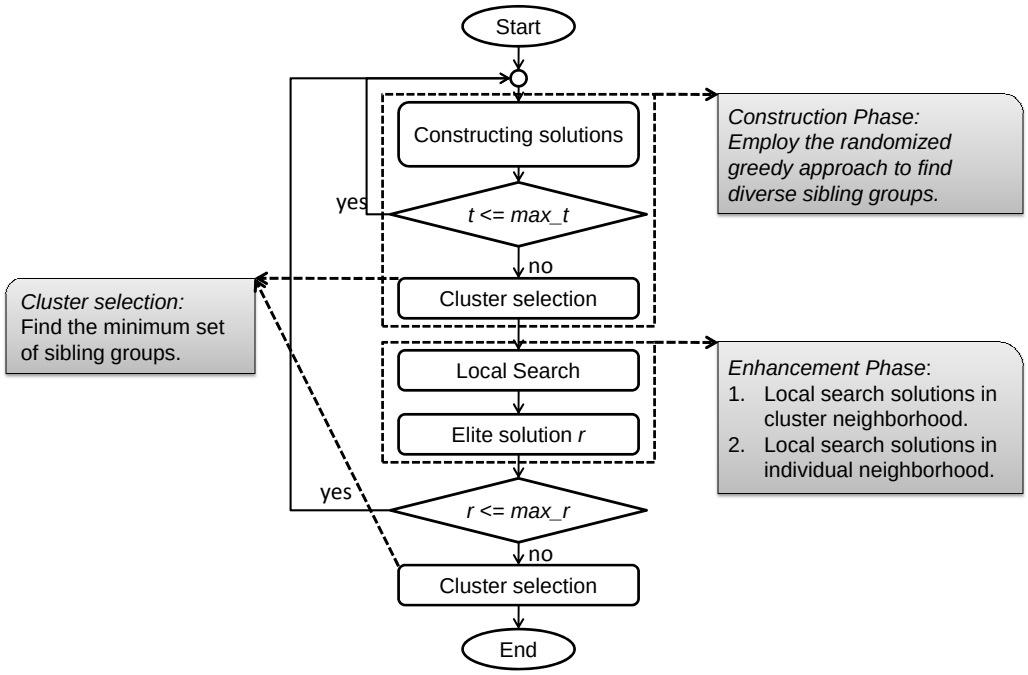

Figure 2.4: Flow diagram of randomized greedy optimization algorithm. *Construction phase* is to construct a set of sibling groups with the randomized perturbations. *Enhancement phase* is to employ the two-stage local search to improve the solution quality. *Cluster selection* is to solve a set covering problem (SCP) to obtain the minimum set of sibling groups. A solution is defined a set of sibling groups (clusters).

Algorithm 1 Randomized greedy optimization algorithm

```
Input: a set of individuals with genetic data
Output: a minimum set of sibling groups

procedure RANDOMIZED_GREEDY_OPTIMIZATION_ALGORITHM(input)
    repeat
        initialization: solution← apply IMCS
        repeat
            solution ← solve IMCSP
            solution ← Update(solution)                    ▷ accumulate solution
        until t > max_t
        solution ← ClusterSelection(solution)              ▷ solve a SCP
        solution ← LocalSearch_Cluster(solution)
        solution ← LocalSearch_Individual(solution)
        solution ← Update(solution)                        ▷ accumulate solution
    until r > max_r
    solution ← ClusterSelection(solution)                  ▷ solve a SCP
return output
end procedure
```

this study, the greedy IMCS approach is employed and generalized by adding a new randomized weight perturbation to it. The idea behind the IMCS procedure is to iteratively construct a sibling group that covers the maximum number of individuals until no individuals are left while each group is subject to the Mendelian capacity constraints. Please refer to [37] for more details. Because the IMCS uses a greedy-based optimization model that has an combinatorial objective function, it is very likely that there exist alternate or multiple optimal solutions. In other words, there may be several different groups with the same number of individuals that can be assigned to the group. In order to obtain diverse solutions in the construction phase, a randomized weight perturbation scheme is introduced. The weight of individual i is defined by π_i and added to the objective function of the IMCS. The concept behind the randomized perturbation is motivated by the noise method proposed in [40]. Note that, without the loss of generality, one can say that the IMCS in [37] uses $\pi_i = 1,\ \forall i \in I$. In our case, the weight is perturbed by adding a noise with a uniform distribution $[1-\epsilon,\ 1+\epsilon]$, where ϵ is a small positive number. The perturbed IMCS (IMCSP) can then be formulated as follows. Define the following decision variables:

- $x_i \in \{0,\ 1\}$: indicates if individual i is assigned to be a member of the current sibling group;
- $y_k^l \in \{0,\ 1,\ 2\}$: indicates if any members in the current sibling group has distinct ($y_k^l = 1$) or homozygous ($y_k^l = 2$) allele(s) k at locus l;
- $v_{kk'}^l \in \{0\ 1\}$: indicates if allele k appears with allele k' in the current sibling group at locus l.

The optimization model of IMCSP is given by

$$\text{(IMCSP)} \quad \max \quad \sum_{i \in I} \pi_i x_i \tag{2.81}$$

$$\text{s.t.} \quad a_{ik}^l x_i \leq y_k^l \quad \forall\, i \in I,\ k \in K,\ l \in L \tag{2.82}$$

$$\sum_{k \in K} y_k^l \leq 4 \quad \forall\, l \in L \tag{2.83}$$

$$\sum_{i \in I} a_{ik}^l a_{ik'}^l x_i \leq M v_{kk'}^l \quad \forall\, k \in K,\ k' \in K \backslash k,\ l \in L \tag{2.84}$$

$$\sum_{k' \in K \backslash k} v_{kk'}^l \leq 2 \quad \forall\, k \in K,\ l \in L. \tag{2.85}$$

The objective in Equation (2.81) is to maximize the total weight of individuals selected to be in the sibling group. The constraint sets in Equations (2.82)-(2.83) are derived from the first rule of the Mendel's laws, which is to ensure that the sum of the total number of distinct alleles and the number of homozygous alleles is less than or equal to four. The constraint sets in Equations (2.84)-(2.85) are derived from the second rule of the Mendel's laws, which is to ensure that each and every allele does not appear with more than two other alleles, except itself, in each locus. The procedure of IMCSP approach is shown in Algorithm 2, which is to solve the IMCSP model iteratively.

Algorithm 2 IMCSP

```
Input: a set of individuals with genetic data
Output: a partition of sibling groups

procedure IMCSP(input)
    initialization: generate a perturbation randomly
    repeat
        solution ← solve IMCSP(input)
        solution ← Update(solution)                ▷ accumulate solution
        remove selected individuals from the input set
    until no individual is assigned
return output
end procedure
```

In addition to the randomized weight perturbation scheme, we introduce a cut constraint to explore and further diversify alternate optimal solutions of IMCSP. This

situation discussed in [37]. The cut constraint is defined by

$$\sum_{i \in \bar{I}} x_i \leq |\bar{I}| - 1, \tag{2.86}$$

where $\bar{I} \subset I$ contains only the individuals assigned in the current group. The implementation of this cut constraint is described as follows. We first solve the original IMCSP model, add the cut constraint to the IMCSP to remove the current optimal solution from the feasible space, and then resolve the IMCSP model with the cut constraint to obtain an alternate optimal solution. By using this cut constraint, we propose two variants other than the original IMCSP:

1. IMCSP_1: add the cut constraint to the original IMCSP in the first and second iterations;

2. IMCSP_2: add the cut constraint to the original IMCSP repeatedly in the first iteration.

Cluster Selection: Minimum Set Covering Problem

Cluster selection is the last step of the construction phase. The goal of cluster selection is to select the best subset of sibling groups from a pool of high-quality solution candidates generated by the iterative IMCSP. It can also be used to remove redundant or dominated groups from the solution pool. Cluster selection can thus be mathematically formulated as a SCP. Define a binary assignment matrix d_{ij}, which presents that individual $i \in I$ is assigned to sibling group $j \in S$, where S is a pool of all sibling group candidates. The SCP is given by $\min \sum_{j \in S} z_j$; s.t. $\sum_{i \in I} d_{ij} z_j \geq 1, \forall j \in S$. The objective of SCP is to find the minimum set of sibling groups. The constraint set ensures that each individual must be covered by at least one of sibling group candidates. Note that this SCP is relatively small, and it can be solved efficiently by any MIP solvers.

Enhancement Phase: Improving the solution quality

The goal in the enhancement phase is to improve the solution quality with respect to the pairwise similarity degree of individuals assigned to the same sibling groups by performing local search. Generally, a local search starts with an initial solution,

explores alternative solutions in the neighborhood, makes a move to a better solution, and terminates when no better solution is found. In our case, the initial solution is given as a set of sibling groups $j \in J$ selected in the construction phase. The associated feasible space is defined as all constructed sibling groups $j \in S$. The effectiveness of local search thus relies on its evaluation function, initial solution, neighborhood definition, and search strategies. The evaluation function, which we want to maximize, is herein defined by the pairwise similarity degree of individuals assigned to the same sibling groups, which is the second term in Equation (2.65),

$$\sum_{j \in J} \sum_{(i,\, i') \in E} (\sum_{l \in L} w_{ii'}^{l}) x_{ij} x_{ij}. \tag{2.87}$$

To improve the efficiency of search procedure, we employ a memory function, which is motivated by the tabu search [59, 61]. The memory function is used to collect the past movements, which are associated to solutions, and to guide the search path in an improving direction. In the search path, the most recently visited solution enters the memory, and the oldest one is removed from the memory. Each solution in the memory must be visited until it is removed from the list. This is mainly to prevent a local cyclic search where there are many similar solutions to explore. In addition, the memory length is one of keys to affect the search efficiency. Longer memory length may guide the search path in the wrong direction, while shorter memory length may not have any effect. However, there is not a standard setting for the memory length, which really depends on the problem complexity.

We herein propose a two-stage local search in cluster-based and individual-based neighborhoods. In the cluster-based search, a cluster switch is performed when a sibling group with a higher pairwise similarity is randomly selected from other solutions to replace a sibling group with a lower similarity in the current solution. To record the cluster movement, we define the memory structure as $(j_1, j_2, ..., j_n)$, where j is the label of sibling group visited and n is the memory length. Subsequently after the cluster-based search, local search in the individual-based neighborhood is performed. An individual shift is performed when an individual is randomly selected from one sibling group and

shifted to another sibling group, also selected randomly. Similar to the cluster-based search, the memory structure is defined as $([j_1, i_1], [j_2, i_2], ..., [j_m, i_m])$, where j and i are the labels of sibling group and individual visited, and m is the memory length. After some moves, the solution may no longer be feasible because the new individual added to the sibling group may violate the Mendelian capacity constraints. In such a case, this movement is forbidden and a new neighbor (solution) is reselected. Thus, it is necessary to check if the current movement is forbidden in every iteration. Note that, by definition of individual-based neighborhood, the feasible space is reduced from S to J and fixes on only sibling groups $j \in J$ determined from the first stage. The local search is performed iteratively. The stopping criteria are the maximum number of search iterations for both stages and the maximum number of no-improvement consecutive iterations. The local search terminates when whichever stopping criterion is reached first.

Final Cluster Selection

The final step of RGOA is to perform the final cluster selection to find the minimum set of sibling groups from a number of elite sets. This step is similar to the last step of the construction phase.

2.8.4 Computational Settings

In this study, all computational experiments were programmed in MATLAB, and all MIP models were solved using a callable GAMS library with CPLEX version 10.0 (default setting). All experiments were run on an Intel Xeon Quad Core 3.0GHz processor workstation with 8 GB RAM memory. Execution time reported in this section were obtained from the desktop's internal timing calculations, which include time used for preprocessing and postprocessing.

The parameter settings of algorithm implementation are as follows. Each test data instance was implemented in a 20-hour computing time limit. The maximum number of RGOA replications was set to $max\_r = 100$. The maximum number of construction iterations was set to $max\_t = 50$ in the construction phase, where three variants of IMCSP, IMCSP_1, and IMCSP_2 were applied. In the enhancement phase, the major

stopping criterion, the maximum number of search iterations, for two stages of local search were given by $50 \times |J|$ and $50 \times |I|$, respectively, and the auxiliary stopping criterion, the maximum number of no-improvement consecutive iterations, was set to 20, where $|J|$ is the cardinality of cluster set and $|I|$ is the cardinality of individual set.

2.8.5 Reconstruction Results of RGOA

As mentioned in the previous section, there are three variants of our approach in the reconstruction phase: IMCSP, IMCSP_1, and IMCSP_2, and there are two stages in the enhancement phase: *cluster-based* and *individual-based*. The average and standard deviation of the reconstruction accuracies of all three variants after each phase of the framework are reported in Table 2.8. It can be seen that there are not significant differences among the three variants. Overall the accuracies gradually increase from the construction phase to the enhancement phase with the exceptions of the salmon and shrimp data sets. However, for the ant data set, the local search achieved a 100% reconstruction accuracy.

Table 2.8: Reconstruction accuracies (%) in terms of *mean* ± *standard deviation* of the reconstruction results from different phases of RGOA tested on all data sets.

Species	Constructive strategy	Phase 1	Phase 2 *cluster-based*	Phase 2 *individual-based*
Salmon	IMCSP	98.29 ± 0	98.29 ± 0	98.29 ± 0
	IMCSP_1	98.01 ± 0	98.01 ± 0	98.29 ± 0
	IMCSP_2	98.29 ± 0	98.29 ± 0	98.29 ± 0
Shrimp	IMCSP	98.73 ± 2.54	98.73 ± 2.54	98.73 ± 2.54
	IMCSP_1	98.73 ± 2.54	98.73 ± 2.54	98.73 ± 2.54
	IMCSP_2	94.92 ± 0	94.92 ± 0	94.92 ± 0
Fly	IMCSP	52.82 ± 4.14	56.79 ± 4.78	59.59 ± 5.07
	IMCSP_1	54.74 ± 5.86	56.56 ± 4.83	58.02 ± 5.25
	IMCSP_2	53.16 ± 3.79	56.05 ± 3.90	58.36 ± 3.83
Ant	IMCSP	98.81 ± 0.94	99.60 ± 0.18	100 ± 0
	IMCSP_1	98.81 ± 0.56	99.47 ± 0	100 ± 0
	IMCSP_2	98.67 ± 0	99.47 ± 0	100 ± 0
Turtle	IMCSP	47.54 ± 1.87	48.57 ± 2.22	49.03 ± 2.05
	IMCSP_1	46.50 ± 2.26	48.00 ± 2.07	48.43 ± 2.11
	IMCSP_2	46.29 ± 6.47	46.29 ± 6.47	46.86 ± 5.66

Table 2.9 presents the best final results of reconstruction accuracies and the numbers of sibling groups from the last step of elite cluster selection. It is observed that the proposed RGOA achieved 100% reconstruction accuracy on the shrimp and ant data

sets. It is interesting to note that in other data sets that RGOA did not achieve 100% accuracy either there are missing allele information (fly and turtle) or violations in the Mendel's laws (salmon and turtle). For these reasons, RGOA did not provide accurate reconstruction results on those data sets. Nevertheless, even if the true optimal solutions were obtained, the reconstruction accuracies would be poor as well. The real reason is that the objective of our optimization framework and the Mendelian constraints assume that the data are not erroneous. In fact, most genetic data are erroneous. Thus a more robust optimization framework should be further investigated. From the table, it is also observed that IMCSP_1 and IMCSP_2 with the cut constraint are more time-consuming. From the last column in Table 2.9, for the same amount of time limit the numbers of replications of IMCSP_1 and IMCSP_2 are obviously smaller than IMCSP because each iteration of IMCSP takes much less time than that of IMCSP_1 and IMCSP_2. From our computational experience, we conclude that the IMCSP variant without the cut constraint should be used in order to save the computing time, yet maintain a good solution quality. On the other hand, the introduction of randomized perturbation can be helpful in terms of the diversification in the case where practitioners want to explore alternate solutions.

Table 2.9: Final results of the number of sibling groups, accuracy (%) and the number of replications. The computing time is limited within 20 hours (72,000 seconds). The perfect reconstruction are underlined.

			Final Results			
Species	Constructing strategy	Actual # of sibling groups	# of sibling groups	Accuracy (%)	# of replications	Time (sec.)
Salmon	IMSCP	7	7	98.29	6	> 72,000
	IMSCP_1	7	7	98.29	2	> 72,000
	IMSCP_2	7	7	98.29	1	> 72,000
Shrimp	IMSCP	13	<u>13</u>	<u>100.00</u>	4	> 72000
	IMSCP_1	13	13	94.92	4	> 72,000
	IMSCP_2	13	13	94.92	1	> 72,000
Fly	IMSCP	6	7	58.95	22	> 72,000
	IMSCP_1	6	7	65.79	22	> 72,000
	IMSCP_2	6	7	63.16	7	> 72,000
Ant	IMSCP	10	<u>10</u>	<u>100.00</u>	2	> 72,000
	IMSCP_1	10	<u>10</u>	<u>100.00</u>	2	> 72,000
	IMSCP_2	10	<u>10</u>	<u>100.00</u>	1	> 72,000
Turtle	IMSCP	26	18	56.57	10	> 72,000
	IMSCP_1	26	17	51.43	9	> 72,000
	IMSCP_2	26	18	42.86	2	> 72,000

2.8.6 Comparison with Other Existing Methods

To illustrate that our approach is among the best sibling reconstruction methods developed thus far, we compare the solution quality of RGOA and that of other state-of-the-art methods in the literature. The methods in the literature reported here include 2AOM, IMCS, A&F, B&M, KINGROUP, and COLONY. The IMCS approach solves a full optimization model 2AOM with 2-allele constraints to generate a partition of maximal sibling groups with 2-allele constraints while the statistical likelihood measure is not incorporated [37]. The A&F algorithm is based on a completely combinatorial approach to exhaustively enumerate all possible sibling groups satisfying the 2-allele constraints and obtain a maximal, not necessarily optimal, collection of sibling groups [13]. The B&M algorithm is based on a mixture of likelihood and combinatorial techniques used to construct a graph with individuals as the nodes and the edges weighted by the pairwise likelihood (relatedness) ratio. The algorithm identifies potential sibling groups by finding the connected components in the graph [24]. The KINGROUP algorithm is based on the likelihood estimates of partitions of individuals into sibling groups by comparing, for every individual, the likelihood of being part of any existing sibling group with the likelihood of starting its own group [86]. The COLONY approach uses the maximum likelihood method to assign sibship and parentage jointly [135].

Table 2.10: Comparison results in accuracy (%) with other state-of-the-art approaches on five different species. The best results are underlined.

Species	RGOA [a]	IMCS	2AOM	A&F	B&M	KG	COLONY
Salmon	98.29	98.29	94.02	–[b]	98.29	94.60	56.70
Shrimp	100.00	100.00	96.61	67.80	100.00	77.97	100.00
Fly	63.16	47.37	66.84	31.05	19.62	54.73	–[c]
Ant	100.00	93.10	–[d]	–[b]	97.61	97.10	100.00
Turtle	56.57	40.00	–[d]	–[b]	38.18	39.40	40.00

[a] We report the best accuracy among all experiments.
[b] A&F ran out of 4GB memory as it enumerates all possible sibling groups.
[c] There are no results available.
[d] No feasible solutions are obtained within 20 hours time limit.

Reconstruction accuracies of the above-mentioned reconstruction methods and RGOA on all biological data sets are shown in Table 2.10. Note that the best reconstruction results of RGOA among different parameter settings are reported. The most accurate

reconstruction results are underlined. In all cases, RGOA obtained the best reconstruction results and outperformed all other methods. It is worth noting that although the RGOA's construction phase is based on the IMCS approach, randomized perturbation and local search can greatly improve the reconstruction accuracies. Specifically for the fly and turtle data sets, in which there are a lot of missing values, RGOA was able to increase the accuracies by about 15%. Both B&M and KINGROUP appear to be inaccurate on the data sets with a lot of missing values. We were not able to obtain the reconstruction results from the A&F algorithm on the salmon, ant, and turtle data sets because it ran out of memory when enumerating all possible combinations.

In Figure 2.5, we show the reconstruction accuracies of RGOA with the constructing strategy IMCSP on two real data sets (ant and turtle) over the time shift, which are compared to 2AOM and IMCS. Accuracies of RGOA are averaged by the number of replications at the time of 4, 8, 12, and 16 hours, and accuracies at 20 hours are obtained by final cluster selection. RGOA approach can achieve as good as, even better than accuracies IMCS approach although it takes longer computing time to obtained solutions. Moreover, it guarantees to have more diverse solutions so that we obtain better reconstruction accuracies on these two data sets. On the other hand, compared to 2AOM, we can always obtain good feasible solutions in a relatively short time (< 20 hours) for large and complex data sets.

2.8.7 Performances on Simulated Data sets

To show the ability of the proposed RGOA approach for larger complex data sets, we apply a random population generator [37] to generate larger simulated data sets when the real data sets at hand are relatively small-size, even the largest available in the literature. Essentially, the mechanism of the random population generator is to first construct a group of parents with the full genetic information such that a single generation of true sibling groups is known a priori. The generation process is as follows with parameters required: M/F is the number of male/female adults, l is the number of sampled loci, a is the number alleles per locus, j is the number of juveniles in the

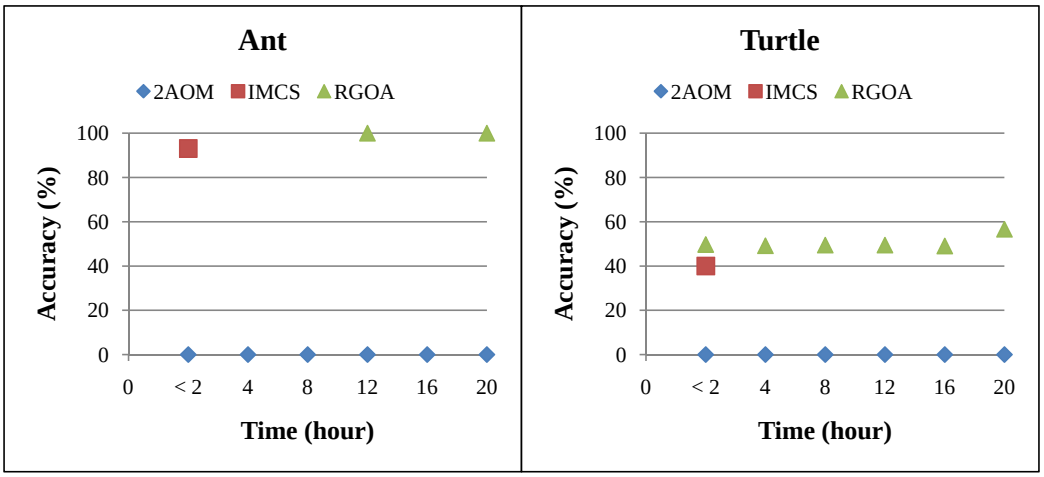

Figure 2.5: Averaged accuracies of RGOA on real data sets (ant and turtle) are obtained over time shift, compared to 2AOM and IMCS approaches in [37]. Accuracy = 0 represents that no feasible solution is available by 2AOM at the time. For IMCS, all solutions are obtained within two hours.

population per one adult female, and o is the number of maximum number of offsprings per parent couple.

Step 1. First, we generated the parent population of M males and F females with parents with l loci, each having a distinct alleles per locus.

Step 2. After the parents were generated, we created a population of their offsprings by randomly selecting j pairs of parents. A male and a female were chosen independently and uniformly at random from the parent population.

Step 3. For each of the chosen parent pairs, we generated a specified number of offsprings, o, each randomly receiving one allele from its mother and one from its father at each locus.

The parameter settings for larger simulated data sets are given: M and $F = 30$, $j = 10$, $o = 40$ and 50, $l = 2, 3$, and 4, and $a = 10$. Additional computational settings are considered as follows. As mentioned previously, we suggest to adopt the constructive strategy IMCSP in the construction phase to save the computing time. For diversification reason, we expect to have more replications of ROGA) within a fixed computing time by shortening the construction phase. We add a stopping criterion of the maximum number of no-improvement consecutive iterations based on the similarity score in the construction phase and slightly reduce the maximum number of construction

iterations to 20. Thus, the construction procedure terminates when whichever stopping criterion is reached first. The results are reported in Table 2.11, in turn, the number of sibling groups, accuracy (%) and the number of replications within 20 hours time limit, and compared to the known sibling relationships. We can still obtain good results in terms of the number of sibling groups and reconstruction accuracy. More accurate reconstruction is obtained when there are more genetic information (i.e., more loci). However, more loci make the problem more complex to solve, which can be seen that the number of replications of RGOA decreases with the complexity of problems because it is more time-consuming to solve for a single solution in the construction phase. Moreover, we compare the performance of the RGOA to IMCS and 2AOM approaches in [37]. The accuracies are reported in Table 2.12. With proposed randomized perturbation and local search, we obtain better reconstruction accuracies than IMCS. 2AOM can not be solved to obtain the solutions within 20 hours time limit. It is shown that our proposed approach is capable of solving larger complex problems effectively.

Table 2.11: Results of RGOA approach tested on larger simulated data sets. Final results are reported, in turn, the number of sibling groups, accuracy (%) and the number of replications within 20 hours (72,000 seconds) time limit, and compared to the known sibling relationships. The perfect reconstruction are underlined.

Simulated data set	Actual # of sibling groups	Final Results			
		# of sibling groups	Accuracy (%)	# of replications	Time (sec.)
Rand-j10-o40-l2-a10	10	10	91.00	24	> 72,000
Rand-j10-o50-l2-a10	10	10	91.60	13	> 72,000
Rand-j10-o40-l3-a10	10	<u>10</u>	<u>100.00</u>	7	> 72,000
Rand-j10-o50-l3-a10	10	10	99.80	7	> 72,000
Rand-j10-o40-l4-a10	10	<u>10</u>	<u>100.00</u>	3	> 72,000
Rand-j10-o50-l4-a10	10	<u>10</u>	<u>100.00</u>	3	> 72,000

Table 2.12: Accuracy results of RGOA approach compared to IMCS and 2AOM approaches [37] from the simulated data sets.

Simulated data set	RGOA	IMCS	2AOM[a]
Rand-j10-o40-l2-a10	91.00	89.00	-
Rand-j10-o50-l2-a10	91.60	79.40	-
Rand-j10-o40-l3-a10	100.00	98.25	-
Rand-j10-o50-l3-a10	99.80	96.80	-
Rand-j10-o40-l4-a10	100.00	99.25	-
Rand-j10-o50-l4-a10	100.00	100.00	-

[a] No feasible solution is obtained within 20 hours time limit.

2.9 Conclusion

In this chapter, we studied an important clustering problem, sibling reconstruction problem, in computational and population biology. The objective of the problem is to establish sibling relationships in a population with parental information. We first presented an optimization model 2AOM based on the 2-allele constraints derived from Mendel's laws and further extend the 2AOM to a complete optimization model integrated with the statistical likelihood of genetic data. The sibling reconstruction problem had shown to be a generalization of the well-known NP-hard set covering problem. We developed a heuristic approach IMCS to efficiently solve the 2AOM model based on a maximum covering approximation algorithm. Although the IMCS approach has been able to accurately reconstruct sibling groups, the solution (i.e., the number of sibling groups) is yet guaranteed to be optimal mathematically. A column generation approach was therefore proposed to obtain an exact solution (not a real objective for sibling reconstruction). Moreover, we modeled the problem as a capacitated clustering problem to be solved by a proposed randomized greedy optimization algorithm. From the comprehensive experiments for the real and simulated data sets, the computational results demonstrated the effectiveness and practicability of the proposed approaches, and better performance when compared to the existing approaches.

Moreover, in practice, the full sibling reconstruction is limited to monogamous species. Various open questions in population biology that undergo the common challenges in accuracy and efficiency, such as half-sibling or high-level sibling relationship reconstructions, have been investigated. In [126], similar combinatorial optimization models and algorithms for half-sibling group reconstruction have been proposed.

Chapter 3

IMPROVED PATTERN GENERATION METHODS IN LAD USING MEDICAL DATA[1]

3.1 Introduction

Binary classification is one of nominal classification problems in data mining and typically deals with the determination of two-class data. The goal of classification is to classify the nature of new (testing) data in consistent with the hidden structural information that decision patterns or rules extract from the historical (training) data. For instance, in clinical medicine, a patient is diagnosed to have heart disease according to the health history from which the decision pattern is the fasting blood sugar being greater 120 mg/dl and the resting electrocardiographic result being abnormal; and otherwise. For such problem, supervised learning approaches are usually focused on discovering the decision patterns to precisely classify the new data.

While in the literature there have been several statistical and machine learning methods being proposed such as support vector machines (SVM), decision tree (*J48*), neural network (NN), and logistic regression (LR), Logical Analysis of Data (LAD) is a relatively new data mining framework based on combinatorics and optimization, and designed for data analysis with both binary input and output [68, 47]. The reason that the LAD appears to be more of a practical choice of classifiers is because the final LAD model can be easy to interpret by the end user. In contrast, the classification models of SVM or NN are commonly treated as a black box. In addition, the advantages of LAD also include exhaustively identifying the entire set of features less or more correlated

[1]The chapter is part of a submitted manuscript [43] in collaboration with Wanpracha Art Chaovalitwongse, Tibérius Oliveira Bonates, and Chungmok Lee.

to the outcome and the inter-effects among features, possibly providing the detailed explanation for the conclusion of LAD, and guiding to develop customized decision systems (e.g, a personalized treatment for breast cancer) [69].

The LAD framework consists of four main steps: data binarization, feature selection, pattern generation, and classification model construction [29]. Given a set of observed data of I objects in positive and negative classes represented by K features. As LAD is designed for binary data analysis, the first step is to employ data binarization, which converts each individual non-binary (e.g., nominal and numerical) feature into associated binary features. The binarization usually produces a much larger number of binary features than original features. Next the feature selection step is carried out to find a (minimum) support set of binary features that can best describe the characterization of original data. With the feature support set, positive or negative patterns are generated in conjunction with one or more binary features that can distinguish at least one object in one class from all objects in the other class. Finally, a classification model (called LAD model or classifier) is built by aggregating all patterns into a discriminant function that is used to classify new objects. The basic of the LAD framework will be described in more detail in Section 3.2.

In the literature, patterns in the LAD framework are shown to be the key building blocks [29, 71, 7, 10, 3, 8]. Patterns are combinatorially formed of one or more features to capture the indications of the nature of historical data and aggregated into a classifier to classify new data. In the past, most studies proposed enumeration-based approaches to generate all patterns to build a "good" classifier [29, 52, 7, 4]. Although enumerating all possible patterns can result in a good classifier, it is computationally expensive as the data size increases and some redundant patterns are governed by other dominant patterns to cover the target objects. On the other hand, some studies developed heuristic approaches to generate a limited set of "good" patterns by controlling parameters such as the degree (the number of features included), coverage (the number of observed data covered), and number of patterns [10, 26, 27, 121]. However, they are faced with the difficulty to improve accuracy due to limitations on the parameters of patterns. Therefore, generating good patterns with considering the tradeoff between

classification accuracy and computational efficiency appears to be very challenging.

Because pattern generation from any given (binary) features can be viewed as a combinatorial procedure, in this study, we propose a new combinatorial optimization approach to only solve for decisive and high-quality patterns without setting parameters in advance. Each pattern is generated based on only uncovered objects. Furthermore, we develop a new column generation framework to build an "optimal" LAD classifier to improve the classification accuracy and computational efficiency of LAD, where the proposed pattern generation approach is employed.

This chapter is organized as follows. In Section 3.2, the basic knowledge of LAD is reviewed. In Section 4.2.2, new mathematical optimization approaches for pattern generation are presented and then a new column generation framework is developed. In Section 3.4, the background of test data sets is described. In Section 4.2.3.3, the evaluation of LAD classification performance is introduced. In Section 3.6, the classification performance of the proposed approaches on widely used medical data sets from the UCI machine learning repository are demonstrated, compared with other existing pattern generation approaches in LAD and state-of-the-art classification methods. This chapter is concluded in Section 3.7.

3.2 Basics of Logical Analysis of Data

In this section, we review the basic implementation of LAD, which can be divided into four steps: data binarization, feature selection, pattern generation, and classification model construction. The LAD framework is displayed in Figure 3.1. For more detailed information, we refer the interested reader to the literature where the theory and implementation of LAD are exhaustively explained [68, 47, 29].

We first define the following notations that will be used thorough out the chapter. Given a set I of observed objects in both positive and negative classes, where $I = I^+ \cup I^-$ and $\emptyset = I^+ \cap I^-$. Each object is represented by a set of features, denoted by F. A set of binary features, denoted by K, is additionally defined as it is generated from the binarization of the original feature set F. Assume there are two set S^+ and S^- of

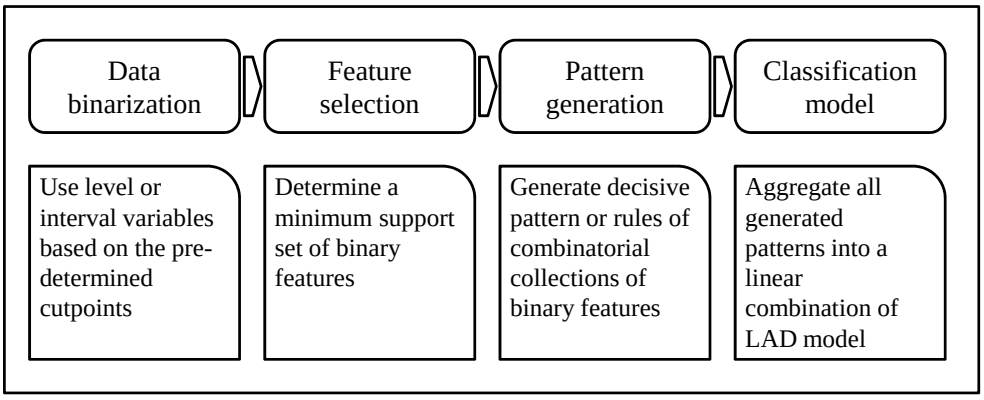

Figure 3.1: The LAD framework with four steps: data binarization, feature selection, pattern generation, and classification model.

positive and negative patterns, where $S^+ \cup S^+ = S$, generated on the basis of binary feature set K, are aggregated into a LAD classifier. Other variables will be subsequently defined later as needed.

3.2.1 Data Binarization

As LAD requires the binary input in data analysis, data binarization is needed as a preprocessing step for the data containing non-binary (e.g., nominal and numerical) features (or variables). The objective of data binarization is to design a binary mapping by generating a finite set of "cutpoints" to map the original features into a new set of binary features [28, 29]. A set of cutpoints $C^f = \{c_1^f, c_2^f, ..., c_{b_f}^f\}$ for feature f is determined according to the distribution of its values in relation to class information. First, the feature values along with class information are sorted in ascending order. Subsequently, searching from the smallest value, a cutpoint is determined by taking the average of the feature values of the two objects when the labels of their associated class change, and so on. Finally, a new set of binary features is mapped by using the set of cutpoints. There are usually two ways for a mapping of non-binary features using "level" and "interval" variables [29]. The level variable mapped from non-binary

variable x by a cutpoint c is defined as

$$L(x,\ c) = \begin{cases} 1, & \text{for } x \geq c\ ; \\ 0, & \text{for } x < c. \end{cases} \tag{3.1}$$

The interval variable mapped from non-binary variable x by a pair of cutpoints c_1 and c_2, where $c_1 < c_2$, is defined as

$$I(x,\ c_1,\ c_2) = \begin{cases} 1, & \text{for } c_1 \leq x < c_2\ ; \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

The determination of cutpoints largely depends on the user, so it also can be done in any other manner, such as a variant of interval variable considering an equal-sized interval based on the mean and variance of feature values [88].

After data binarization, as explained, the number of binary features is in turn increased to be much larger than the number of original features. To efficiently determine a small set of principal cutpoints that is critical to distinguishing the two classes of data, [28] not only provided a theoretical foundation of data binarization, but also developed an optimization approach. Later, [105] proposed an eliminative approach (IDEAL) to iteratively remove redundant cutpoints to achieve a minimal discriminant set. Other discretization approaches can be found in [97, 88].

3.2.2 Support Feature Selection

Redundant features to discriminate the same objects and irrelevant features unable to discriminate any objects may be produced after the data binarization. The selection of an irreducible set of features, called a "minimum support set", is carried out by eliminating such redundant or irrelevant features. The problem of finding the minimum support set can be formulated as a set covering problem in combinatorial optimization in Equations (3.3)-(3.5) [29]. First binary variables are defined: $y_k = 1$ indicates if feature k is selected, and $y_k = 0$ otherwise; $\varepsilon_{ij} = 1$ indicates if the pair of $i \in I^+$ and

$j \in I^-$ is not properly discriminated by new binarized features, and $\varepsilon_{ij} = 0$ otherwise.

$$\text{(FS-SCP)} \quad \min \quad \sum_{k \in K} \phi_k y_k + M \sum_{i \in I^+, j \in I^-} \varepsilon_{ij} \tag{3.3}$$

$$s.t. \quad \sum_{k \in K} c_{ij}^k y_k \geq d_{ij}(1 - \varepsilon_{ij}) \quad \forall\, i \in I^+,\ j \in I^- \tag{3.4}$$

$$y_k,\ \varepsilon_{ij} \in \{0, 1\},\ k \in K,\ i \in I^+, j \in I^-. \tag{3.5}$$

In FS-SCP, the coefficient ϕ_k is a weight controlling the importance of feature k. It is based on prior domain knowledge. However, when no domain knowledge is available, ϕ_k is usually set to 1. The binary indicator $c_{ij}^k \in \{0,\ 1\}$ defines whether or not the value of feature k is different for a pair of $i \in I^+$ and $j \in I^-$. The right-hand-side value $d_{ij} > 0$ is a minimum quantity to ensure that each pair of $i \in I^+$ and $j \in I^-$ is distinguished by at least d_{ij} features. In general cases, $d_{ij} = 1$ as default. For a case where a pair of $i \in I^+$ and $j \in I^-$ is not distinguishable on feature k, it is penalized with a cost $M = |K| + 1$ in the objective function in Equation (3.3).

Not that any feature selection methods can be employed in the LAD framework as long as the interpretation of selected features for the original data still keeps retained when new feature space is reduced.

3.2.3 Combinatorial Pattern Generation

Patterns in LAD are combinatorially formed by one or more binary features, which characterize common structural information that objects from the same class share. By definition, a pure positive (or negative) pattern consists of a set of conditions on the values of binary features. Such pattern must have has an empty intersection with the subset in the class and a nonempty intersection with any subset in the other class. Here we shall define parameters that are used to characterize patterns: "degree" is defined as the number of features used in a pattern, ranging from one up to the length of a minimum support set; "coverage" is defined as the number of objects covered in the class. Note that after the feature selection step, feature is referred to a binary feature of the minimum support set whenever mentioned.

Pattern generation is mostly carried out in an enumeration way. Several diverse types of patterns with different characteristics have been well studied in the literature. A prime pattern is a pattern that any one of features included cannot be removed; otherwise it is not considered a pattern [29]. A combinatorial top-down-bottom-up approach was proposed to enumerate all possible prime patterns that are small-sized (simplicity) and cover at least one objects (comprehensiveness). A spanned pattern is a pattern that is spanned if it does not include properly any other interval containing the same subset of objects. An incrementally polynomial time algorithm was proposed to enumerate all spanned patterns [7]. A maximum pattern is a pattern that covers as many objects in the class as possible and covers no objects in the other class [27]. The authors proposed several exact and heuristic approaches to generate maximum patterns. Meanwhile, a maximum box problem was studied for generating patterns directly from original data [52] and an accelerated algorithm was proposed for enumerating all possible patterns of limited degree [10]. Pareto-optimal patterns with respect to suitability criteria were analyzed [71]. More recently, a mixed-integer linear programming (MILP) based approach was proposed to iteratively generate a limited set of patterns by specifying the parameters [121].

3.2.4 Classification Model Construction

According to the definition of patterns, each pattern must cover at least one object, and each object is covered by at least one pattern. Ideally, a collection of such patterns is expected to be aggregated into a good classifier, which in turn interprets comprehensive information of the data. A LAD classification model therefore can be built as a linear combination of positive and negative patterns, given by

$$\Delta(x) = \sum_{s^+ \in S^+} \omega^+_{s^+} P_{s^+}(x) + \sum_{s^- \in S^-} \omega^-_{s^-} N_{s^-}(x), \tag{3.6}$$

where $\omega^+_{s^+} \geq 0$ and $\omega^-_{s^-} \leq 0$ are the weight coefficients for positive and negative patterns. Indicator $P_{s^+}(x) = 1$ if an object x is covered by positive pattern s^+, and $P_{s^+}(x) = 0$

otherwise. $N_{s^-}(x) = 1$ if an object x is covered by negative pattern s^-, and $N_{s^-}(x) = 0$ otherwise.

When classifying a new object x, if $\Delta(x) > 0$, x is classified as positive. On the other hand, if $\Delta(x) < 0$, x is classified as negative. If $\Delta(x) = 0$, new object x is left unclassified. Note that when prior information is not available, the equal weights, i.e., $\omega_s^+ = \frac{1}{|S^+|}$ and $\omega_s^- = \frac{-1}{|S^-|}$, are usually used for positive and negative patterns in the discriminant function in Equation (3.6).

3.2.5 Medical Applications

Since the introduction of LAD, it has been successfully applied to practical problems in biomedicine [94, 9, 6, 2, 118], finance [84, 70], and other disciplines. In clinical applications, LAD was applied to risk stratification of coronary artery disease according to health history, medication (beta blockers, verapamil, etc.), and specific measurements (resting abnormal ECG, resting heart rate, change in heart rate, etc.) [94, 9]. LAD was also applied to identifying high-risk patients who would benefit from aggressive therapy, and low-risk patients who needed to be treated with conservative care. Later, LAD was applied to the diagnosis of ovarian cancer from mass spectroscopy-generated proteomic data [6], the diagnosis of diffuse large B-cell lymphomas from gene expression data [2], and the prognosis of breast cancer from gene expression data [1]. More recently, [118] adapted the LAD technique to diagnose acute ischemic stroke. Other than these clinical applications, LAD was used for selecting features from a vast number of genomic and proteomic data based on different criteria [5], and for selecting short oligo probes in genotyping applications [83].

3.2.6 Illustrative Example

We illustrate the LAD procedure with an example of five data points with two numerical features in Figure 3.2. In binarization, we use the *level* and *interval* variables on feature f_1 and the *interval* variable on feature f_2 to discretize the original feature values of a data point into a new set of binarized features. The binarized features b_1, b_2, and b_3 are

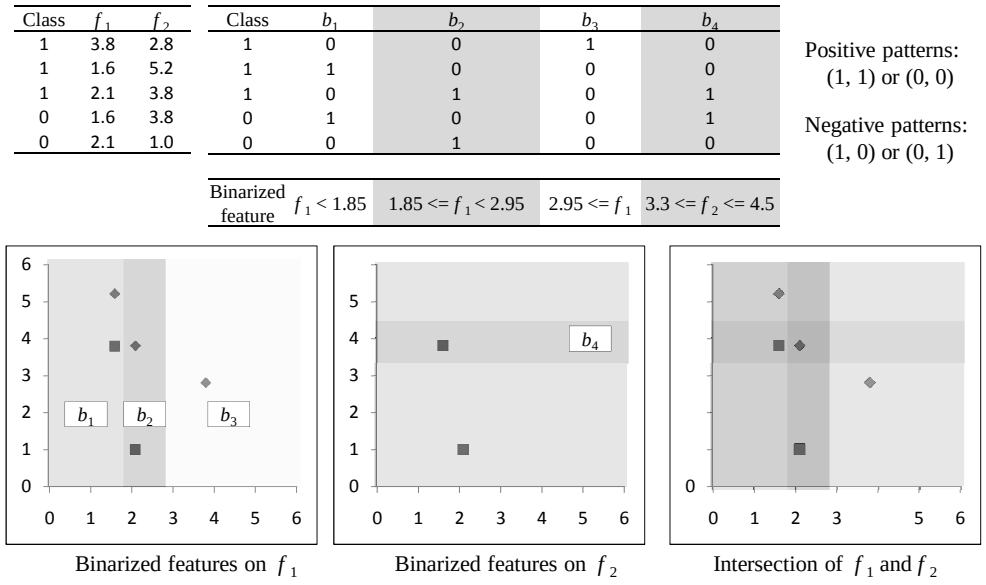

Figure 3.2: An illustration of the LAD procedure for a data set of two numerical features f_1 and f_2. Binarized features b_1, b_2, and b_3 are from f_1, and a binarized feature b_4 is from f_2. Two binarized features b_2 and b_4 are selected in feature selection. In pattern generation, two positive and two negative patterns are constructed and used in the LAD model.

associated to the feature f_1, and the binarized feature b_4 is associated to the feature f_2. Subsequently, feature selection enables one to obtain a minimum support set, $\{x_2, x_4\}$, where x_2 associates to $1.85 \leq f_1 < 2.95$ and x_4 associates to $3.3 \leq f_2 \leq 4.5$. Positive and negative patterns cane be generated based on these two binarized features. As a result, positive patterns $\{1, 1\}$ and $\{0, 0\}$ are generated to cover positive data points, while negative patterns $\{1, 0\}$ and $\{0, 1\}$ are generated to cover negative data points. We then construct a LAD model consisting of two positive and two negative patterns. A new data point with features $\{f_1, f_2\} = \{2.5, 1.0\}$ is classified negative using the discriminant function in Equation (3.6) (i.e., $\Delta = (0.5)\times(0)+(0.5)\times(0)+(-0.5)\times(1)+(-0.5)\times(0) = -0.5$).

3.2.7 Other Classification Methods

In the literature, classification techniques range from statistics, machine learning, data mining, to optimization. Here we briefly introduce the state-of-the-art methods proposed in recent years, which are also used for performance comparison in this chapter. There are support vector machines [125], decision tree (J48) [117], random forests [32], logistic regression [77], and multilayer perceptron [75]. They are available in Weka package [65, 140]. The support vector machines (SVM) was first introduced in [133, 125]. The SVM is an optimization based classification method, which constructs a linear classifier in a (possible) high-dimensional space (called hyperplanes) exploiting the *Kernel trick* to separate the two classes so that the margin between the support vectors is maximized. In general, *soft* margin is considered to allow some data points incorrectly classified, that is, to avoid overfitting (permits models to make errors). The Decision Tree (C4.5) method (J48) was first introduced in [117]. This induction method consists of a hierarchical partition of a given space into (nearly) homogenous spaces. In a decision tree, a rule on one or more features is involved at each node, and each branch of the node corresponds to one of possible outcomes of the decision rule. For example, $x_1 \geq 3$ is one outcome of the rule on the feature x_1, and $x_1 < 3$ is the other outcome of the rule. The random forests (RF) method proposed in [32] is a generalized model of the decision tree. The idea of RF is to build a collection of decision trees by bootstrapping from the data, which has been shown to be a more powerful classification method. The logistic regression (LR) is a regression technique especially tuned for binary classification problems. The maximum likelihood estimation is trained to build a logistic regression model that is fit in a linear function and able to predict a nonlinear function over input features. The multilayer perceptron is a neural network (NN) that comprises input, hidden, and output layers. The objective is to minimize the error between output of built network and actual weight value. The key settings involved are determining the number of layers and nodes (neurons), which vary depending on experienced experiments.

3.3 New Pattern Generation Methods in LAD

As pattern generation is a very hard combinatorial optimization problem, the main focus of this work is to propose new pattern generation methods to improve the accuracy and efficiency of the LAD framework. In the first part, we propose a new mathematical optimization approach using a mixed-integer programming (MIP) model for decisive pattern generation. In the second part, we develop a new column generation framework to construct an "optimal" LAD classification model so as to improve classification accuracy.

Without loss of generality, it is assumed that the data sets can be perfectly classified without noises (excluding missing data), and each object being classified belongs to one and only one class, The set K is redefined as a minimum support set of binary features since all features have been preprocessed by the steps of data binarization and feature selection. For the sake of brevity, all descriptions are given only for positive patterns, unless explicitly mentioned, because the symmetric definition of positive and negative patterns is obvious.

3.3.1 Mixed-Integer Programming Models for Pattern Generation

3.3.1.1 Maximum Coverage Patterns

From the definition of maximum patterns [27], a pattern is generated to cover as many objects in the class as possible and to cover no objects in the other class. Following this idea, we present a new approach to generate a maximum coverage pattern (MCP) such that the degree is minimal, the coverage is maximized and any opposite coverage is penalized. This approach is motivated by the soft margin concept in SVM. MCP can be formulated as a MIP problem. First we define binary variables used in the model. $x_i = 1$ indicates if object i is covered by the pattern, and $x_i = 0$ otherwise. $y_k = 1$ indicates if feature k is used in the pattern, and $y_k = 0$ otherwise. $\varepsilon_j = 1$ indicates if negative object j is misplaced due to the situation where certain positive and negative objects are identical (since it may be caused by mistakes in the sampling process), and

$\varepsilon_j = 0$ otherwise. The MIP formulation of MCP is given in Equations (3.7)-(3.10).

$$\begin{aligned}
(\mathsf{MCP}) \quad \max \quad & \sum_{i \in I^+} x_i - M \sum_{j \in I^-} \varepsilon_j && (3.7)\\
s.t. \quad & (1 - b_{ik}^+) y_k \leq 1 - x_i \quad \forall\, i \in I^+,\ k \in K && (3.8)\\
& \sum_{k \in K} (1 - b_{jk}^-) y_k \geq 1 - \varepsilon_j \quad \forall\, j \in I^- && (3.9)\\
& x_i,\ y_k,\ \varepsilon_j \in \{0, 1\}. && (3.10)
\end{aligned}$$

Binary indicator $b_{ik}^+ = 1$ indicates if the values of positive object i and the reference pattern are not distinguishable on feature k, and $b_{ik}^+ = 0$ otherwise. Binary indicator $b_{jk}^- = 1$ indicates if the values of negative object j and the reference pattern are not distinguishable on feature k, and $b_{jk}^- = 0$ otherwise. The objective in Equation (3.7) is to maximize the number of covered positive objects. The constraint set in Equation (3.8) ensures that positive objects are covered, whereas the constraint set in Equation (3.9) ensures that the pattern discourages negative objects covered. When a peculiar situation happens to a negative object, it is penalized with a cost $M = |I| + 1$ in the objective function.

In order to solve this MCP, a reference pattern is required a priori. How to choose a good starting reference pattern is, however, not intuitive. Here, we propose a heuristic approach to choose a reference pattern. An object r with the maximum dissimilarity to the opposite subset is chosen from the uncovered subset to be a reference pattern r. The dissimilarity is calculated by the sum of distances of an object $i \in I^+$ to all negative objects $j \in I^-$ over all features $k \in K$. The reference pattern is determined by

$$r = \arg\max_{i \in I^+} \{ \sum_{j \in I^-} \sum_{k \in K} h_{ij}^k : h_{ij}^k \in \{0, 1\},\ j \in I^-,\ k \in K \}, \tag{3.11}$$

where $h_{ij}^k = 1$ if the values of objects i and j are different on feature k, and $h_{ij}^k = 0$ otherwise. If two or more objects have the same degree of dissimilarity, then any one of them is chosen arbitrarily.

In most real-life problems, a single pattern cannot possibly cover all objects at a

time. Thus, we propose an iterative procedure to iteratively generate multiple patterns such that all objects are guaranteed to be covered. In every iteration, we determine a reference pattern from the uncovered subset, update b_{ik}^{+} and b_{jk}^{-}, and then solve MCP. After solving the MCP, we remove objects that have been covered. The procedure is performed until all objects are covered. In the end, patterns generated are very diverse, including "dominant patterns" and "odd patterns". The former is defined as a pattern that covers most objects in the class, while the latter is defined as a pattern that only covers a few objects hard to be covered by dominant patterns.

3.3.1.2 Weighted Maximum Coverage Patterns

While the MCP model emphasizes the pattern size of covered objected, it is also important to generate more patterns that are diverse. We We introduce the diversification to the MCP model by introducing a weighted coefficient $\frac{1}{\beta^{n_i}}$ to each object in the objective function, where β is a given adjustable variable ($\beta \geq 1$) and n_i is the number of times object i is covered by previously generated patterns. The modified MIP formulation of the weighted MCP (WMCP) is given by

$$\text{(WMCP)} \quad \max \quad \sum_{i \in I^+} \frac{1}{\beta^{n_i}} x_i - M \sum_{j \in I^-} \varepsilon_j \tag{3.12}$$

$$s.t. \quad (1 - b_{ik}^{+}) y_k \leq 1 - x_i \quad \forall\, i \in I^+,\ k \in K \tag{3.13}$$

$$\sum_{k \in K} (1 - b_{jk}^{-}) y_k \geq 1 - \varepsilon_j \quad \forall\, j \in I^- \tag{3.14}$$

$$x_i,\ y_k,\ \varepsilon_j \in \{0, 1\}. \tag{3.15}$$

The objective function in Equation (3.12) is to maximize the total weight sum of covered objects. The constraint sets in Equations (3.13)-(3.14) follow the same descriptions in MCP. A peculiar situation happening to a negative object is penalized with a cost $M = |I| + 1$ in the objective function. It is important to note that, in the iterative procedure, we do not remove the covered objects after solving the WMCP in each iteration. Thus each object can be covered by more than one pattern although we still determine a reference pattern from the uncovered subsets, and update b_{ik}^{+} and b_{jk}^{-}.

It can be observed that when β is very large, the weighted coefficient becomes rather small and the objects covered by previously generated patterns would not be covered again. This is conceptually equivalent to the MCP. On the other hand, when β is equal to one, no matter how many times objects are covered in previous pattern iterations, they can be covered by a new pattern as the covered object contributes to the objective function value as much as other uncovered objects. This situation is similar to the approach to exact maximum patterns (EMP) [27]. The difference is that WMCP only generates a limited set of decisive patterns while EMP generates a number of patterns bounded by the number of target objects. In short, this intermediate approach WMCP, between EMP and MCP, can generates flexible patterns by varying the value of parameter β.

3.3.1.3 Other Remarks

Here several advantageous properties of the patterns generated by MCP and WMCP are remarked. First, our approach can account for missing values that is very common in real-life data. In pattern generation, the missing value of an object is treated to be the opposite value. For example, a positive object with a missing value (1, 0, *) is replaced by (1, 0, 0) when compared to a positive pattern (1, 0, 1), and this object cannot be covered. On the contrary, a negative object with a missing value (1, 0, *), replaced by (1, 0, 1), is covered by the positive pattern (1, 0, 1). This property guarantees any improper objects not to be classified and is referred to the *robustness* mentioned in the literature [29, 27]. Secondly, a pattern generated can tolerate the peculiar situation to cover a few objects in the other class since they are identical to certain objects in the class. This property is referred to the *fuzziness* [27]. In addition, instead of enumerating all possible patterns, MCP and WMCP generate a relatively small set of decisive patterns, each using a minimum feature set (*simplicity*), that ensure every object to be covered (*comprehensiveness*).

3.3.2 Column Generation for Construction of LAD Classification Model

In the previous subsection, we propose a fast and effective pattern generation approach for a LAD classifier. However, the generated patterns may miss certain coverage in which some objects are hard to cover or already covered by the objects from other class. Such odd patterns are not desirable in a classification model. For these reasons, we realize that the construction of LAD classifier can be cast into a column generation framework. This study is then focused on how to construct the best possible LAD classification model by using the column generation framework. We propose two objective approaches to construct an optimal LAD classification model, which are (I) minimum positive and negative pattern sets and (II) a maximum separation margin between positive and negative subsets. For the sake of simplicity, we may use the terms of pattern and column interchangeably throughout this section.

In a column generation framework, the problem is decomposed into a master problem (MP) and a subproblem (SP). First, we solve a linear programming (LP) relaxation of restricted master problem (RMP) with a limited set of patterns. A set of (optimal) dual variables associated to the target objects is produced and passed to the SP as a guide for generating patterns. The purpose of SP is to price out improving (or beneficial) patterns with respect to the (optimal) dual variables. Subsequently, newly generated patterns are added into the RMP after checking the optimality of MP. The RMP is updated with new patterns and resolved. The procedure is iteratively performed until there are no patterns to improve the objective function value of MP, which implies that the current LP solution to the MP is optimal.

3.3.2.1 Master Problem

Objective I: Minimum Pattern sets

The first objective approach is to generate the minimum number of positive and negative pattern sets that cover all objects. The master problem can be modeled as a minimum set covering problem. Binary variable $z_{s^+} = 1$ indicates if positive pattern s^+ is selected to be a member of LAD classifier from the positive pattern subset S^+, and $z_{s^+} = 0$

otherwise. Indicator $P_{is^+} = 1$ indicates if positive object i is covered by positive pattern s^+, and $P_{is^+} = 0$ otherwise. The integer programming (IP) formulation of the problem is given in Equations (3.16)-(3.18).

$$\text{(Min-Pattern)} \quad \min \quad \sum_{s^+ \in \hat{S}^+} z_{s^+} \tag{3.16}$$

$$\text{s.t.} \quad \sum_{s^+ \in \hat{S}^+} P_{is^+} z_{s^+} \geq 1 \quad \forall\, i \in I^+ \tag{3.17}$$

$$z_{s^+} \in \{0, 1\}. \tag{3.18}$$

The objective in Equation (3.16) is to minimize the number of positive patterns used in the LAD classifier. The constraint set in Equation (3.17) ensures that every positive object i has to be covered by at least one positive pattern.

In column generation iterations, we can start Min-Pattern with a subset of feasible positive patterns $\hat{S}^+ \subset S^+$ as a RMP. We also need to relax binary variables $z_j \in \{0, 1\}$ to $z_j \in [0, 1]$ since Min-Pattern is an IP problem. Subsequently, the LP relaxation of RMP of Min-Pattern is solved to obtain a set of dual variables μ_i^+, which are associated to the constraint set in Equation (3.17). With these dual variables, it provides the information for generating patterns in SP. Finally, we solve the original IP model of Min-Pattern with all generated patterns to obtain the IP solution (i.e., the set of selected patterns) to Min-Pattern.

We note that the above Min-Pattern is only carried out to generate positive patterns. Because there is no inter-effect on both positive and negative pattern generations, in this objective approach, column generation procedures for both positive and negative patterns are performed independently.

Objective II: Maximum Separation Margin

The second objective approach directly associates the discriminant function in Equation (3.6) and model a LAD classification model as a maximization of the discriminant function. The objective function can be viewed as a hyperplane in the feature space with the margins of separation of two classes that are maximized. We first denote positive

and negative separation margins $r \geq 0$ and $t \leq 0$. Indicator $P_{is^+} = 1$ if positive object i is covered by positive pattern s^+ and $P_{is^+} = 0$ otherwise, and indicator $N_{js^-} = 1$ if negative object j is covered by negative pattern s^- and $N_{js^-} = 0$ otherwise. $\omega^+_{s^+}$ and $\omega^-_{s^-}$ are weight coefficients for positive and negative patterns, respectively. In addition, $\zeta_i \geq 0$ and $\zeta_j \geq 0$ are introduced to indicate if positive object i and negative object j is correctly classified, respectively. The Maximum Separation Margin formulation is given by

$$
\begin{aligned}
\text{(Max-Margin)} \quad \max \quad & r + t - M\Big(\sum_{i \in I^+} \zeta_i + \sum_{j \in I^-} \zeta_j\Big) && (3.19)\\
\text{s.t.} \quad & r - \sum_{s^+ \in \hat{S}^+} \omega^+_{s^+} P_{is^+} + \sum_{s^- \in S^-} \omega^-_{s^-} N_{is^-} - \zeta_i \leq 0 \quad \forall\, i \in I^+ && (3.20)\\
& t + \sum_{s^- \in \hat{S}^-} \omega^+_{s^+} P_{js^+} - \sum_{s^- \in S^-} \omega^-_{s^-} N_{js^-} - \zeta_j \geq 0 \quad \forall\, j \in I^- && (3.21)\\
& \sum_{s^+ \in \hat{S}^+} \omega^+_{s^+} = 1 && (3.22)\\
& \sum_{s^- \in \hat{S}^-} \omega^-_{s^-} = 1 && (3.23)\\
& r \geq 0,\ t \leq 0,\ \omega^+_{s^+} \geq 0,\ \omega^-_{s^-} \geq 0,\ \zeta_{s^+} \geq 0,\ \zeta_{s^-} \geq 0. && (3.24)
\end{aligned}
$$

The objective in Equation (3.19) is to maximize the sum of positive and negative separation margins. The constraint sets in Equations (3.20) and (3.21) determine the smallest positive and negative separation margins over all objects, respectively. If there is any object misclassified, it is penalized with a cost $M = |S^+| + |S^-| + 1$ in the objective function. The constraint sets in Equations (3.22) and (3.23) ensure that the margins for both positive and negative patterns do not increase to infinity by restricting the sums of positive and negative weights to 1.

In column generation iterations, similarly, we solve Max-Margin to obtain dual variables μ^+_i, μ^-_j, λ^+_i, λ^-_j, which are associated to Equations (3.20), (3.21), (3.22), and (3.23), respectively. With these dual variables, it provides the information for generating patterns in SP. Note that in Max-Margin, it has considered the effects of positive and negative patterns simultaneously on the construction on LAD classifier, so we do not

need to separately perform Max-Margin for positive and negative patterns. However, generating positive and negative patterns in SP are still performed independently in every iteration.

3.3.2.2 Pricing Subproblem: Apply MCP and WMCP

As mentioned previously, the purpose of SP is to price out beneficial patterns that can improve the objective function value of the MP. Here we directly employ the proposed pattern generation approaches MCP (and WMCP) while any exact and approximation approaches can be applied to solve the SP. Passed from the RMP, dual variables, e.g., μ_i^+ and μ_j^- in the Min-Pattern, provide the information about objects hardly being covered and are simply used as weight coefficients in the objective function of MCP in Equation (3.7). It is rewritten as $\max \sum_{i \in I^+} \mu_i^+ x_i - M \sum_{j \in I^-} \varepsilon_j$ for positive patterns and similarly for negative patterns. It is noted that when we solve the MCP to generate positive patterns, only the constraints in Equation (3.8) associated to the positive objects with non-zero dual variables need to be considered while all the constraints in Equation (3.9) associated to negative objects must be included. In such a way, we ensure that the identified pattern only covers the associated objects and avoid any objects in the other class. It can be done for Max-Margin in a similar way.

3.3.2.3 Calculating Reduced Costs

In every iteration, a new pattern generated in SP is associated with a reduced cost that is computed with dual variables and used to check if the new pattern is a candidate to be added in MP. Any pattern can be a candidate as long as its associated reduced cost is imposed to improve the objective function value of MP. When no improving patterns are generated, the optimal discriminant function is obtained for LAD classification model.

For Min-Pattern, we consider patterns with "negative" reduced costs to be candidates added in the RMP since it is a minimization problem. The reduced cost for positive

pattern is computed by

$$\gamma_{s^+} = 1 - \sum_{i \in I^+} \mu_i^+ P_{is^+} \quad \forall s^+ \in \hat{S}^+, \tag{3.25}$$

where μ_i^+ is the dual variable. The optimality condition is given by

$$\bar{z} := \min\{\gamma_{s^+} = 1 - \sum_{i \in I^+} \mu_i^+ P_{is^+} \mid s^+ \in \hat{S}^+\}. \tag{3.26}$$

If $\bar{z} \geq 0$, there are no improving patterns to be generated that is, optimality condition holds. Otherwise, patterns s^+ is a candidate. It can be done for negative patterns in a similar way.

Similarly, for Max-Margin, we obtain the dual variables associated to positive and negative object subsets. We only consider patterns with "positive" reduced costs to be candidates included in the RMP since it is a maximization problem. The reduced cost for a positive pattern is computed by

$$\gamma_{s^+}^+ = \lambda^+ + \sum_{i \in I^+} \mu_i^+ P_{is^+} - \sum_{j \in I^-} \mu_j^- P_{js^+} \quad \forall s^+ \in \hat{S}^+ \tag{3.27}$$

and the reduced cost for a negative pattern by computed by

$$\gamma_{s^-}^- = \lambda^- - \sum_{i \in I^+} \mu_i^+ N_{js^-} + \sum_{j \in I^-} \mu_j^- N_{js^-} \quad \forall s^- \in \hat{S}^-, \tag{3.28}$$

where μ_i^+, μ_j^-, λ_i^+, λ_j^- are duel variables. The optimality condition is similar to Equation (3.26) by replacing the reduced costs in Equations (3.27) and (3.28). The optimality condition holds if $\bar{z} \leq 0$; otherwise.

3.4 Description of Data Sets

To show the practicability of our proposed approaches in medical applications, we adopt 14 medical data sets available from the UCI machine learning repository [16]. These data sets include three sets of Wisconsin breast cancer (wbc, wdbc, and wpbc), five sets of heart disease diagnosis from different databases (hrt-c, hrt-h, hrt-s, hrt-lb, and hrt-stat), one set of hepatitis (hpts), one set of Bupa liver disorder (bld), one set of Pima Indians' diabetes(pid), two sets of Cardiac Single Proton Emission Computed Tomography (SPECTF and SPECT), and one set of Parkinson's disease (prks). The characteristics of the data sets are summarized in Table 3.1, including the numbers of observations, class representation, and numbers of features. For most data sets with non-binary features, the preprocessing steps of data binarization and feature selection are carried out in advance, as described in Section 3.2. The numbers of binary features and minimum support set are also reported in the last two columns in Table 3.1. The minimum support sets of binary features, instead of original support sets, are used in all experiments. The background of the data sets is given as follows:

Breast cancer: There are three data sets used in the applications of diagnosis and prognosis of breast cancer, which are obtained from William H. Wolberg at University of Wisconsin Hospitals, Madison. The objective of diagnosis is mainly to discriminate that the tumor is malignant or benign. For diagnosis, the data set (wbc) contains 699 observations in two classes (malignant and benign) with 9 sampled features, and the other data set (wdbc) contains 569 observations in two classes with 30 sampled features. There are 3% and 2% missing values, respectively. Another data set (wpbc) is used in prognosis to predict if breast cancer is likely to recur when a patient has the cancer excised. There are 198 observations in two classes (recurrence or non-recurrence) with 33 sampled features. These data sets were first analyzed in [103], and [102].

Heart disease: There are four data sets (hrt-c, hrt-h, hrt-s, and hrt-lb) obtained from Andras Janosi from Hungarian Institute of Cardiology in Budapest, William Steinbrunn from University Hospital in Switzerland, Matthias Pfisterer from University Hospital in Switzerland, and Robert Detrano from V.A. Medical Center, Long Beach and

Cleveland Clinic Foundation. They are used in the application of predicting the presence of heart disease. Numbers of observations are, respectively, 303, 294, 122, and 200 in two classes (sick and normal) with 13 sampled features. After eliminating features having a larger portion of missing values that primarily are unclassified, the numbers of used features are 13, 10, 10, and 8. Small portion of missing values still appear in the data sets. Another data set, StatLog (hrt-stat), is obtained from the Cleveland Clinic Foundation, courtesy of R. Detrano, which is in a slightly different format from above data sets. There are 270 observations in two classes with 13 sampled features and no missing values.

Hepatitis: This data set (hpts) was donated by Gail Gong and examined in the application of predicting whether or not the patient with hepatitis lives. It contains 142 observations in two classes (die and live) with 18 sampled features. Note that we use a subset of original data set by eliminating observations with a larger portion of missing values and age feature due to no significant discrimination. There are remaining 7% missing values.

BUPA liver disorders: This data set (bld) was donated by Richard S. Forsyth from BUPA Medical Research Ltd., which is used in the application of predicting whether or not the male patient has a liver disorder based on blood tests and alcohol consumption. There are 345 observations in two classes (positive and negative) with 6 sampled features. There are no missing values.

Pima Indians' diabetes: This data set (pid) was donated by Vincent Sigillito from National Institute of Diabetes and Digestive and Kidney Diseases, which is used for the diagnosis of whether or not the patient shows signs of diabetes. There are 768 observations in two classes (positive and negative) by 8 sampled features. All patients are at least 21-year-old females of Pima Indian heritage. There are no missing values.

Cardiac Single Proton Emission Computed Tomography: This data set (SPECTF) was originally donated by Lukasz A. Kurgan and Krzysztof J. Cios [91] from the University of Colorado at Denver, which is used for the diagnosis of cardiac Single Proton Emission Computed Tomography images . There are 267 observations in two classes (normal and abnormal) with 45 continuous-valued features. The logical

version of SPECTF data set (SPECT) is processed to obtain 22 binary features. There are no missing values.

Parkinsons disease: This data set (prks) was created by Max Little from the University of Oxford in collaboration with the National Centre for Voice and Speech, Denver, Colorado, which is used for the objective to discriminate healthy patients from ones with parkinsons disease by detecting dysphonia [96]. There are 195 observations with 22 sampled features. There are no missing values.

Table 3.1: Characteristics of real data sets.

	Observations			Class	Original	Binaried	Minimum
Dataset	Total	+	−	Representation (+, −)		Features	
wbc	699	458	241	(malignant, benign)	9	75	12
wpbc	198	47	151	(recurrent, nonrecurrent)	33	1361	15
wdbc	569	212	357	(malignant, benign)	30	3384	18
hrt-c	303	139	164	(sick, normal)	13	306	11
hrt-h[a]	294	106	188	(sick, normal)	10	239	23
hrt-s[a]	122	114	8	(sick, normal)	10	74	7
hrt-lb[a]	200	149	51	(sick, normal)	8	165	34
hrt-stat	270	120	150	(sick, normal)	13	296	16
hpts[a]	142	28	114	(die, live)	18	149	12
bld	325	200	125	(positive, negative)	6	269	17
pid	768	268	500	(positive, negative)	8	857	19
SPECTF	267	212	55	(abnormal, normal)	44	1031	18
SPECT[b]	267	212	55	(abnormal, normal)	22	22	16
prks	195	147	48	(parkisons, normal)	22	731	9

[a] Heart disease and hepatitis data sets are modified by removing some observations with relatively more missing values and not using the feature of age.
[b] This data is the logical version of SPECTF data set that contains non-binary values.

3.5 Performance Measurement

To evaluate the classification performance, we adopt the calculation by counting the number of correctly classified objects, which is widely used in the LAD studies [29, 69, 26, 27]. Let us denote sets of correctly classified positive and negative objects by I_c^+ and I_c^-, and sets of incorrectly classified positive and negative objects by I_u^+ and I_u^-. *Accuracy* is calculated as the average of sensitivity and specificity plus the average of unclassified objects (i.e., $\Delta = 0$) shown in Equation (3.29).

$$Accuracy = 0.5 \times (sensitivity \ + \ specificity \ + \ 0.5 \times (\frac{|I_u^+|}{|I^+|} \ + \ \frac{|I_u^-|}{|I^-|})), \qquad (3.29)$$

where *sensitivity* is the ratio of positive objects correctly classified to the entire positive data set, given by $\frac{|I_c^+|}{|I^+|}$, and *specificity* is the ratio of negative objects correctly classified to the entire negative data set, given by $\frac{|I_c^-|}{|I^-|}$.

Due to possible influences by the unbalanced or contaminated data, we repeat n times k-fold cross validation on randomly shuffled data sets in order to obtain unbiased outcomes. In cross validation, a target data set is equally divided into k subsets, in which one of subsets is used as a testing data set while the remaining $k-1$ subsets are used as a training data set. The classification accuracy is referred to the accuracy based on the testing data set validated for the model that the training data set is used to learn. The overall accuracy is reported by the average of $n \times k$ experiments, where $n = 10$ and $k = 5$ are set in our experiments

3.6 Experimental Results

3.6.1 Results of Analyzing Patterns by MCP and WMCP

In this subsection, we present the analyses of patterns generated by the proposed approaches MCP and WMCP. We first recall that WMCP is an intermediate approach that can generate more flexible patterns by varying controlling parameter β. For the purpose of comparison, we choose two relatively extreme cases, WMCP-2 and WMCP-M, with $\beta = 2$ and $\beta = M$, where M is a very large number. Figure 3.3 illustrates the statistics of patterns generated among MCP, WMCP-2, and WMCP-M. The subset of Cleveland heart disease (hrt-c) data set is used for demonstration. The degree of positive (on the left) and negative (on the right) patterns are shown on the top, and the coverage of positive (on the left) and negative (on the right) patterns are shown on the bottom. Compared to WMCP, MCP uses more diverse patterns of large ranging degrees on the target data set. For the coverage, WMCP generates more dominant patterns while MCP generates more odd patterns. Besides, it is hard to see significant differences between WMCP-2 and WMCP-M.

We also show the performance of these three approaches on 14 data sets in Table 3.2. The numbers of generated patterns, accuracies, and computational times are reported

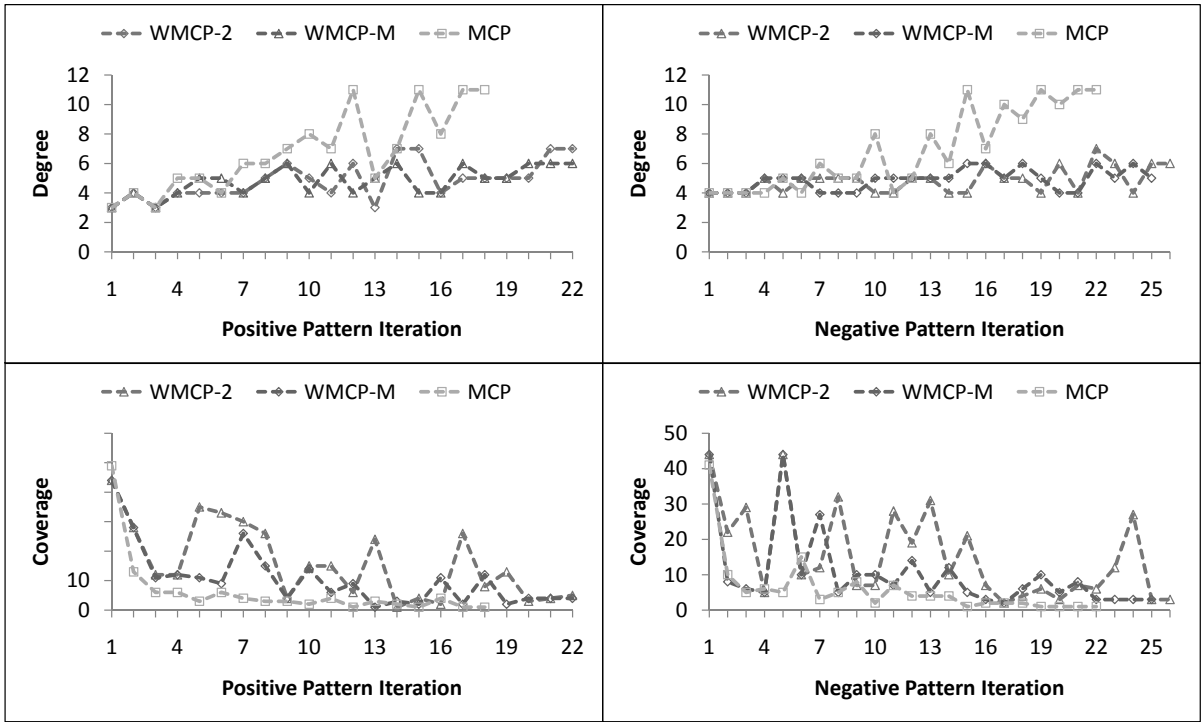

Figure 3.3: Illustration of the numbers of used features (i.e., degree) and intra-class coverage varying in positive (right-hand side) and negative (left-hand side) pattern iterations. Cleveland heart disease data set (hrt-c) is used for demonstration.

for each approach. All experiments were performed within about 30 minutes. From the results, we do not see any significant differences among all in pattern size and accuracy. Note that WMCP-M is reduced to MCP when $\beta = M$. We can see that the quantities of patterns generated by both WMCP-M and MCP are very close.

In addition, as mentioned in Section 3.3.1, WMCP-1 with setting $\beta = 1$ is somehow equivalent to the approach EMP to generate a number of maximum patterns [27]. Table 3.3 present the performance of both approaches. In can be seen that there is no significant difference in accuracy. However, compared to WMCP-1, EMP needs to generate relatively more patterns and takes much more computational time on most instances.

3.6.2 Results of Column Generation for LAD Classification Model

Before we present the results of the proposed column generation framework with the two objective approaches, there are several implementation settings needed to address ahead. In Min-Pattern, the objective function in Equation (3.16) is to find a linear combination of patterns and there may be very similar or duplicate combinations occurring as the quantity of candidate patterns generated gets larger. As studied in the literature [19, 138], in column generation, it is hard to obtain the real optimal solution due to degeneracy and tail-off effects in the large-scale IP problems. For this reason, we introduce a perturbation to the objective function of Min-Pattern in order to attempt to find different possible combinations (diversification). We can rewrite Equation (3.16) as $\sum_{s^+ \in \hat{S}^+} \sigma_{s^+} z_{s^+}$, where σ_{s^+} is a random coefficient uniformly ranging between $[1-\epsilon,\ 1+\epsilon]$ and ϵ is a small positive number. The perturbation is executed only when there is no improvement on the objective function value over several iterations. Note that, however, we do not need to apply the perturbation to Max-Margin because the objective function considers the separation margin resulting from the generated patterns instead of the combination of them. For the termination criterion of the procedure, we terminates it when whichever the following stopping criteria is reached first within a computational time limit of 10 hours. First, the optimality is reached. We also consider the degeneracy, i.e., $\bar{z}_j = 0$, so different patterns may be priced out after a certain number of iterations. Secondly, there is no improvement of the objective function value of RMP after a maximum number of iterations.

Table 3.4 presents the comparison of the performance of Min-Pattern and Max-Margin, compared to MCP, on 14 data sets. The numbers of patterns, accuracies, and computational times are reported. In the column generation of Min-Pattern and Max-Margin, We adopt the initial (feasible) patterns generated by MCP to start with. These patterns are expected to provide a lower bound in terms of the accuracy and the quantity of patterns for Min-Pattern and Max-Margin. Compared to MCP, Min-Pattern yields competitive accuracies with lower quantity of patterns used in the LAD classification model in most instances. Whereas, Max-Margin achieve higher accuracies

by 10-20% with a lot more of patterns in 9 out of 14 instances. As for the computation time, the procedure are finished with the time limit of 10 hours in most stances. We note that for Min-Pattern, it requires double computational time because both positive and negative pattern generations are carried out separately. Besides, we want to point out the efficiency of our approach to generate only decisive patterns. We take an the most complex example of the Pima Indians diabetes data set (pid). Applying Max-Margin results in very good accuracy by using 902 positive and 79 negative patterns approximately for the LAD classification model, whereas a brute-force enumeration needs $\sum_{k=1}^{19} 2^k \begin{pmatrix} 19 \\ k \end{pmatrix}$ in total.

Table 3.5 presents the statistics of the degree of patterns of Min-Pattern, Max-Margin, and MCP. Compared to the minimum support set, it is clear that lower degrees are required to form patterns in most instances. It is worth mentioning that we observe that the LAD classification model with positive patterns of degree 1 yields to 100% accuracies in the instances of the wpbc, wdbc, and prks data sets. It can be interpreted that every measured feature is an independently key feature in the diagnosis and prognosis of these diseases. On the other hand, in the LAD framework, it reflects that the steps of data binarization and feature selection are successfully to extract very critical features.

In addition, here is an important point when implementing column generation for IP problems. Because Min-Pattern is an IP problem, it needs to be relaxed to a LP formulation in the column generation algorithm. Although the final LP solution to Min-Pattern is obtained to be optimal, it is not a real optimal solution and we need to solve the original problem to obtain the final IP solution in the end of procedure. To make sure if the final solution is optimal, we further employ a branch-and-price approach to obtain an exact IP optimal solution, where the column generation algorithm is carried out in a branch and bound framework [19]. In our experiments, the results showed that the final solution of implementing the column generation algorithm is good enough for the LAD classification model although it is still not guaranteed to be optimal. On the other hand, however, there is not a matter in implementing Max-Margin because it is a LP formulation and the final LP solution can be obtained to be optimal.

3.6.3 Comparisons with Existing Approaches

We have presented the effective results of our proposed approaches. To show the classification accuracy, we compare our approaches to a number of related LAD methods proposed in the literature and several state-of-the-art classification algoirhms in freely used package Weka [140, 65].

Table 3.8 presents the accuracies of our pattern generation approaches MCP and WMCP-1, compared to the other similar methods EMP [27], CAP-LAD [27], and MILP [121]. These approaches were carried out using different cross validation and tested on four medical data sets. These approaches somehow have a similar idea of their mathematical optimization models. EMP is an approach to generate an exact number $|I|$ of maximum patterns, each generated based on an object. CAP-LAD is a heuristic approach to improve the computational efficiency. MILP is a MILP-based approach to generate a number of patterns with setting the parameter of degree in advance, each generated by minimizing the number of objects that cannot be covered. All patterns are used to construct a LAD classification model. CAP-LAD and MILP yield among all higher accuracies on average.

Table 3.6 presents the accuracies of our column generation algorithms Max-Margin-MCP and Max-Margin-WMCP-M compared to an existing approach LM-LAD proposed by [26]. Similarly, LM-LAD proposed an objective approach to construct a LAD classification model based on a column generation framework, where the difference is that a branch and bound approach is proposed to generate patterns in subproblem. The results show a significant performance in all instances. Further, compare all results in Table 3.8 and 3.6, our column generation algorithms Max-Margin-MCP and Max-Margin-WMCP-M yield among all higher accuracies on average.

Among statistical and machine learning algorithms, we choose five widely used algorithms that are usually used to compare, including support vector machines (SVM) [125], decision tree (*J48*) [117], random forests (RF) [32], logistic regression (LR) [77], and multilayer perceptron (NN) [75]. They are all available in the software package Weka [140]. To perform a fair comparison, we also run 10 times 5-fold cross validation

and choose the best results among all calibrated parameters for each algorithm. In SVM, *radial basis function* is chosen as the kernel function and complexity parameters are 1, 5, and 50. In *J48*, we use the settings of reduced error pruning and binary splits on nominal features, and consider the minimum number of instances per leaf in the tree by taking values 1 and 2. In RF, only a few features (e.g., 2 or $\log_2(n)$, where n is the number of features) are selected in a single decision tree. We use the numbers of features to be used in random selection by taking values 2 and 10, and trees to be generated by taking values 10, 100, and 1000. In LR, we consider the setting of stoping fitting of logistic models if no new error minimum has been reached in the last iteration (e.g., 50 and 100) and use an error on the probabilities as measure when determining the best number of the LogitBoost iterations. Also, the maximum numbers of LogitBoost iterations are 100 and 500. In NN, we consider the learn rates of 0.3 and 0.5, momentums of 0.2 and 0.4, numbers of hidden layers by taking number of features, and sum of numbers of features and classes. Table 3.8 reports the best accuracies among our proposed approaches, as well as the above-mentioned classification algorithms. It can be seen that our column gneration algorithm Max-Margin-MCP yields the best performance in 9 of 14 instances and the competitive results in the remaining instances. It is worth noting that Max-Margin achieves exactly 100% accuracies in the instances of the wpbc, wdbc, and prks data sets, while the other algorithms have lower accuracies by 10-20%. Max-Margin-MCP achieves good accuracies 10% higher than the other algorithms in the instances of larger and more complex bld and pid data sets.

3.7 Conclusion

As LAD has been shown to be an effective data mining technique for binary classification in many disciplines, there are still underlying issues encountered in classification accuracy and computational efficiency. In this study, we present a new pattern generation approach using a mathematical optimization technique in the LAD framework and develop a new column generation algorithm for the construction of LAD classification model. Tested on a number of widely used medical data sets, the results evidently show

the effectiveness of the proposed approaches for LAD classification. Our approaches are delivered with the following points: (1) to achieve higher accuracy by using less critical patterns in the classification model compared to other enumeration-based LAD methods, (2) to generate diverse patterns including dominant and odd patterns without additional limitations on patterns (parameter-free), and (3) to construct among the most accurate classification model compared to other algorithms such as support vector machines, decision tree, random forests. neural network, and logical regression.

From the perspective of computational implementation, the proposed column generation framework can be generalized. One can propose different objective approaches to meet particular purposes and employ any exact or approximation approaches for pattern generation. To develop more effective and efficient LAD framework, there are many studies to undertake, such as proposing novel data binarization and feature selection methods for logical data, developing new integrated LAD algorithms. and so on. Also, it can be extended to multi-classification problems.

For practicality in the application to medical diagnosis and prognosis, one of the most important advantages of LAD technique is to provide clear and interpretable solutions that are beneficial of decision making in treatment plans. This study indeed offers a new useful tool to overcome the difficulties in efficiency and effectiveness in practical.

Table 3.2: Results of MCP, WMCP-M, andWMCP-2 on 14 data sets. The numbers of patterns, accuracies, and computational times are reported by running 10 times 5-fold cross validation. Cleveland heart disease data set (hrt-c) is used for demonstration.

	MCP					WMCP-99					WMCP-2				
	# of patterns		Accuracy (%)		Time	# of patterns		Accuracy (%)		Time	# of patterns		Accuracy (%)		Time
Data	+	−	Training	Testing	(sec.)	+	−	Training	Testing	(sec.)	+	−	Training	Testing	(sec.)
wbc	17.6	11.7	1.00 ± 0.00	0.95 ± 0.02	270	17.8	11.6	1.00 ± 0.00	0.95 ± 0.01	245	20.0	14.4	1.00 ± 0.00	0.95 ± 0.01	316
wpbc	14.7	1.0	1.00 ± 0.00	0.97 ± 0.05	75	14.7	1.0	1.00 ± 0.00	0.98 ± 0.02	75	14.8	1.0	1.00 ± 0.00	0.98 ± 0.02	77
wdbc	15.6	1.0	1.00 ± 0.00	0.99 ± 0.01	225	15.8	1.0	1.00 ± 0.00	0.99 ± 0.00	247	6.1	1.0	1.00 ± 0.00	0.99 ± 0.00	224
hrt-c	21.4	22.2	0.99 ± 0.00	0.78 ± 0.05	181	21.0	22.5	0.99 ± 0.00	0.77 ± 0.02	176	24.1	25.9	0.99 ± 0.00	0.79 ± 0.02	214
hrt-h	22.2	19.8	0.97 ± 0.01	0.76 ± 0.06	369	22.1	19.3	0.97 ± 0.00	0.75 ± 0.01	352	23.0	20.4	0.97 ± 0.00	0.74 ± 0.03	368
hrt-s	5.7	3.4	0.98 ± 0.01	0.69 ± 0.20	19	5.9	3.4	0.98 ± 0.00	0.74 ± 0.06	20	5.8	3.4	0.98 ± 0.00	0.74 ± 0.03	19
hrt-lb	29.7	21.5	0.85 ± 0.01	0.66 ± 0.07	460	29.9	21.4	0.85 ± 0.00	0.66 ± 0.03	454	31.0	21.4	0.85 ± 0.00	0.65 ± 0.03	455
hrt-stat	24.4	21.9	1.00 ± 0.00	0.73 ± 0.04	302	24.5	21.8	1.00 ± 0.00	0.72 ± 0.02	287	26.5	24.8	1.00 ± 0.00	0.72 ± 0.03	314
hpts	9.2	13.2	0.82 ± 0.05	0.65 ± 0.10	79	12.9	16.8	0.80 ± 0.02	0.65 ± 0.02	111	12.9	18.9	0.81 ± 0.02	0.64 ± 0.03	116
bld	38.8	38.5	1.00 ± 0.00	0.61 ± 0.05	502	38.7	38.3	1.00 ± 0.00	0.61 ± 0.02	493	40.4	39.4	1.00 ± 0.00	0.61 ± 0.01	514
pid	58.7	61.9	1.00 ± 0.00	0.67 ± 0.04	1785	58.4	61.7	1.00 ± 0.00	0.67 ± 0.01	1768	63.5	68.4	1.00 ± 0.00	0.68 ± 0.02	2078
SPECTF	17.8	16.9	1.00 ± 0.00	0.75 ± 0.08	233	17.6	16.8	1.00 ± 0.00	0.75 ± 0.06	215	18.0	17.2	1.00 ± 0.00	0.75 ± 0.06	221
SPECT	24.1	17.1	0.96 ± 0.01	0.68 ± 0.07	236	24.4	17.0	0.96 ± 0.00	0.70 ± 0.06	222	26.9	17.2	0.96 ± 0.00	0.68 ± 0.05	249
prks	8.7	1.0	1.00 ± 0.00	1.00 ± 0.00	33	8.7	1.0	1.00 ± 0.00	0.98 ± 0.06	31	8.7	1.0	1.00 ± 0.00	0.98 ± 0.06	31

Table 3.3: Results of WMCP-1 compared to EMP. The numbers of patterns, accuracies, and computational time are reported by running 10 times 5-fold cross validation.

	WMCP-1					EMP				
	# of patterns		Accuracy (%)		Time	# of patterns		Accuracy (%)		Time
Data	+	−	Training	Testing	(sec.)	+	−	Training	Testing	(sec.)
wbc	22.9	14.9	1.00 ± 0.00	0.95 ± 0.01	355	31.9	21.5	1.00 ± 0.00	0.96 ± 0.01	3492
wpbc	14.7	1.0	1.00 ± 0.00	0.97 ± 0.02	75	17.0	1.0	1.00 ± 0.00	0.99 ± 0.01	253
wdbc	16.7	1.0	1.00 ± 0.00	0.99 ± 0.00	235	19.2	1.0	1.00 ± 0.00	0.99 ± 0.01	1326
hrt-c	27.6	32.5	0.99 ± 0.00	0.80 ± 0.01	241	33.2	45.0	0.99 ± 0.00	0.81 ± 0.01	820
hrt-h	24.3	22.2	0.97 ± 0.00	0.76 ± 0.02	387	42.4	38.9	0.97 ± 0.00	0.76 ± 0.01	739
hrt-s	5.8	3.4	0.98 ± 0.00	0.71 ± 0.06	17	6.3	3.4	0.98 ± 0.00	0.75 ± 0.06	182
hrt-lb	32.4	21.8	0.85 ± 0.00	0.66 ± 0.02	458	48.8	28.4	0.85 ± 0.00	0.64 ± 0.02	535
hrt-stat	28.9	27.7	1.00 ± 0.00	0.73 ± 0.01	330	47.6	44.2	1.00 ± 0.00	0.75 ± 0.02	691
hpts	13.6	21.1	0.80 ± 0.02	0.65 ± 0.03	146	21.8	23.5	0.77 ± 0.01	0.63 ± 0.02	333
bld	45.9	43.4	1.00 ± 0.00	0.61 ± 0.02	612	102.9	83.5	1.00 ± 0.00	0.61 ± 0.01	1253
pid	78.3	91.5	1.00 ± 0.00	0.69 ± 0.01	2685	162.6	235.5	1.00 ± 0.00	0.70 ± 0.01	6694
SPECTF	18.1	17.6	1.00 ± 0.00	0.75 ± 0.05	225	29.5	29.2	1.00 ± 0.00	0.77 ± 0.02	580
SPECT^d	34.3	17.4	0.96 ± 0.00	0.69 ± 0.06	300	33.8	18.6	0.50 ± 0.00	0.50 ± 0.00	1155
prks	8.8	1.0	1.00 ± 0.00	0.98 ± 0.06	32	8.8	1.0	1.00 ± 0.00	1.00 ± 0.00	306

Table 3.4: Results of column generation algorithms Min-Pattern and Max-Margin, compared to the MCP. The numbers of patterns, accuracies, and computational time are reported by running 10 times 5-fold cross validation.

	MCP					Min-Pattern-MCP					Max-Margin-MCP				
	# of patterns		Accuracy (%)		Time	# of patterns		Accuracy (%)		Time	# of patterns		Accuracy (%)		Time
Data	+	−	Training	Testing	(sec.)	+	−	Training	Testing	(sec.)	+	−	Training	Testing	(sec.)
wbc	17.6	11.7	1.00 ± 0.00	0.95 ± 0.02	270	14.5	10.8	0.99 ± 0.00	0.95 ± 0.02	5540	83.9	25.1	0.99 ± 0.00	0.97 ± 0.02	9598
wpbc	14.7	1.0	1.00 ± 0.00	0.97 ± 0.05	75	14.8	1.0	1.00 ± 0.00	1.00 ± 0.00	4075	15.0	1.0	1.00 ± 0.00	1.00 ± 0.00	19789
wdbc	15.6	1.0	1.00 ± 0.00	0.99 ± 0.01	225	15.3	1.0	1.00 ± 0.00	0.99 ± 0.01	35554	16.9	1.0	1.00 ± 0.00	1.00 ± 0.00	> 36000
hrt-c	21.4	22.2	0.99 ± 0.00	0.78 ± 0.05	181	17.3	17.0	0.95 ± 0.01	0.79 ± 0.04	4698	42.0	20.2	0.90 ± 0.02	0.86 ± 0.04	9816
hrt-h	22.2	19.8	0.97 ± 0.01	0.76 ± 0.06	369	19.7	15.8	0.93 ± 0.02	0.77 ± 0.05	61882	47.6	17.3	0.87 ± 0.01	0.85 ± 0.05	32410
hrt-s	5.7	3.4	0.98 ± 0.01	0.69 ± 0.20	19	5.3	3.4	0.97 ± 0.01	0.77 ± 0.20	1987	8.9	3.2	0.93 ± 0.04	0.86 ± 0.18	1486
hrt-lb	29.7	21.5	0.85 ± 0.01	0.66 ± 0.07	460	25.5	20.9	0.84 ± 0.01	0.67 ± 0.07	81107	33.1	11.0	0.71 ± 0.01	0.71 ± 0.06	6267
hrt-stat	24.4	21.9	1.00 ± 0.00	0.73 ± 0.04	302	19.8	15.1	0.93 ± 0.01	0.77 ± 0.06	59496	115.0	20.7	0.90 ± 0.01	0.87 ± 0.04	> 36000
hpts	9.2	13.2	0.82 ± 0.05	0.65 ± 0.10	79	7.7	9.9	0.83 ± 0.07	0.71 ± 0.08	5264	11.1	11.7	0.83 ± 0.05	0.81 ± 0.09	2826
bld	38.8	38.5	1.00 ± 0.00	0.61 ± 0.05	502	31.8	30.4	0.96 ± 0.01	0.64 ± 0.06	23193	302.4	37.4	0.90 ± 0.01	0.84 ± 0.04	10123
pid	58.7	61.9	1.00 ± 0.00	0.67 ± 0.04	1785	45.9	45.1	0.96 ± 0.01	0.68 ± 0.04	>72000	902.6	79.3	0.90 ± 0.01	0.83 ± 0.03	> 36000
SPECTF	17.8	16.9	1.00 ± 0.00	0.75 ± 0.08	233	15.5	14.5	0.97 ± 0.02	0.79 ± 0.08	59407	104.1	17.2	0.94 ± 0.01	0.87 ± 0.06	22819
SPECT	24.1	17.1	0.96 ± 0.01	0.68 ± 0.07	236	18.3	13.9	0.95 ± 0.02	0.73 ± 0.08	53356	28.8	13.5	0.86 ± 0.02	0.83 ± 0.06	3502
prks	8.7	1.0	1.00 ± 0.00	1.00 ± 0.00	33	8.6	1.0	1.00 ± 0.00	1.00 ± 0.00	3766	9.0	1.0	1.00 ± 0.00	1.00 ± 0.00	4926

Table 3.5: Statistics of the degree of patterns generated by MCP, Min-Pattern, and Max-Margin.

Data	Minimum features	MCP +	MCP -	Min-Pattern-MCP +	Min-Pattern-MCP -	Max-Margin-MCP +	Max-Margin-MCP -
wbc	12	4.82 ± 2.78	7.57 ± 1.95	4.40 ± 2.32	7.64 ± 2.10	5.72 ± 2.64	9.20 ± 2.33
wpbc	15	1.00 ± 0.00	14.74 ± 0.00	1.00 ± 0.00	15.00 ± 0.00	1.00 ± 0.00	15.00 ± 0.00
wdbc	18	1.00 ± 0.00	17.88 ± 0.00	1.00 ± 0.00	18.00 ± 0.00	1.00 ± 0.00	18.00 ± 0.00
hrt-c	11	5.93 ± 2.14	5.52 ± 1.66	5.69 ± 1.85	4.93 ± 1.08	6.62 ± 2.36	5.97 ± 2.26
hrt-h	23	11.12 ± 5.41	10.19 ± 5.59	10.48 ± 5.21	10.55 ± 5.22	12.35 ± 5.64	13.52 ± 5.21
hrt-s	7	2.67 ± 1.20	5.65 ± 0.46	2.84 ± 1.29	5.41 ± 0.58	3.02 ± 1.08	5.37 ± 0.58
hrt-lb	34	10.55 ± 4.49	17.23 ± 4.53	10.10 ± 4.65	16.98 ± 4.22	10.58 ± 4.97	18.00 ± 3.77
hrt-stat	16	9.46 ± 4.26	9.35 ± 4.41	9.40 ± 3.89	8.10 ± 3.56	10.10 ± 3.74	11.88 ± 5.31
hpts	12	6.39 ± 2.45	4.31 ± 1.83	6.40 ± 2.60	4.02 ± 1.64	7.29 ± 2.76	4.58 ± 1.83
bld	17	7.58 ± 3.38	7.81 ± 3.30	7.04 ± 2.56	7.63 ± 2.75	8.46 ± 3.41	11.03 ± 4.61
pid	19	9.01 ± 3.35	8.09 ± 2.95	8.74 ± 2.80	7.60 ± 2.22	9.27 ± 3.17	13.51 ± 5.39
SPECTF	18	7.73 ± 4.40	9.97 ± 4.26	6.82 ± 3.65	9.05 ± 3.50	9.04 ± 4.01	13.33 ± 4.95
SPECT	16	7.26 ± 4.79	9.34 ± 3.77	6.44 ± 5.04	8.46 ± 3.58	8.99 ± 5.17	11.72 ± 4.38
prks	9	1.00 ± 0.00	9.00 ± 0.00	1.00 ± 0.00	9.00 ± 0.00	1.00 ± 0.00	9.00 ± 0.00

Table 3.6: Comparison of the results of pattern generation approaches MCP and WMCP-1, and the other approaches EMP, CAP-LAD, and MILP on four data sets.

Data	MCP	WMCP-1[a]	EMP[b]	CAP-LAD[b]	MILP[c]
wbc	0.95 ± 0.02	0.95 ± 0.01	0.95 ± 0.01	0.97 ± 0.01	0.97 ± 0.01
hrt-c	0.78 ± 0.05	0.80 ± 0.01	0.81 ± 0.01	0.83 ± 0.03	0.82 ± 0.06
bld	0.61 ± 0.05	0.61 ± 0.02	0.64 ± 0.04	0.73 ± 0.05	0.70 ± 0.04
pid	0.67 ± 0.04	0.69 ± 0.01	0.58 ± 0.03	0.75 ± 0.02	0.76 ± 0.03

[a] WMCP-1 with $\beta = 1$ is conceptually equilalent to EMP and CAP-LAD.
[b] The results are reported by 10-fold cross validation [27].
[c] The results are reported by 5-fold cross validation [121].

Table 3.7: Comparison of the results of our column generation algorithms Max-Margin-MCP, Max-Margin-WMCP-M, and the other approach LM-LAD by [26] on four data sets.

Data	Max-Margin-MCP	Max-Margin-WMCP-M	LM-LAD
wbc	0.97 ± 0.02	0.97 ± 0.01	0.94 ± 0.02
hrt-c	0.86 ± 0.04	0.85 ± 0.05	0.81 ± 0.03
bld	0.84 ± 0.04	0.84 ± 0.06	0.68 ± 0.02
pid	0.83 ± 0.03	0.83 ± 0.03	0.68 ± 0.03

Table 3.8: Comparison of the results of our best approaches and five state-of-the-art algorithms on 14 data sets.

Data	Our[a]	SVM	*J48*	RF	NN	LR
wbc	0.97 ± 0.02	0.97 ± 0.02	0.94 ± 0.02	0.97 ± 0.01	0.96 ± 0.02	0.96 ± 0.02
wpbc	1.00 ± 0.00	0.77 ± 0.02	0.75 ± 0.05	0.80 ± 0.04	0.77 ± 0.05	0.80 ± 0.05
wdbc	1.00 ± 0.00	0.97 ± 0.01	0.93 ± 0.02	0.96 ± 0.02	0.97 ± 0.01	0.97 ± 0.02
hrt-c	0.86 ± 0.04	0.84 ± 0.05	0.78 ± 0.05	0.83 ± 0.05	0.79 ± 0.05	0.83 ± 0.05
hrt-h	0.85 ± 0.05	0.81 ± 0.04	0.79 ± 0.04	0.80 ± 0.04	0.78 ± 0.05	0.83 ± 0.05
hrt-s	0.86 ± 0.18	0.94 ± 0.02	0.93 ± 0.02	0.93 ± 0.03	0.89 ± 0.06	0.92 ± 0.04
hrt-lb	0.71 ± 0.06	0.75 ± 0.01	0.72 ± 0.05	0.75 ± 0.04	0.69 ± 0.07	0.74 ± 0.04
hrt-stat	0.87 ± 0.04	0.84 ± 0.05	0.78 ± 0.05	0.83 ± 0.04	0.80 ± 0.05	0.83 ± 0.05
hpts	0.81 ± 0.09	0.87 ± 0.05	0.82 ± 0.06	0.87 ± 0.05	0.81 ± 0.06	0.85 ± 0.06
bld	0.84 ± 0.04	0.58 ± 0.00	0.62 ± 0.05	0.73 ± 0.06	0.68 ± 0.06	0.69 ± 0.05
pid	0.83 ± 0.03	0.77 ± 0.03	0.74 ± 0.03	0.76 ± 0.03	0.75 ± 0.03	0.77 ± 0.03
SPECTF	0.87 ± 0.06	0.79 ± 0.00	0.78 ± 0.05	0.81 ± 0.03	0.77 ± 0.05	0.79 ± 0.04
SPECT	0.83 ± 0.06	0.83 ± 0.04	0.80 ± 0.03	0.82 ± 0.04	0.80 ± 0.04	0.82 ± 0.05
prks	1.00 ± 0.00	0.87 ± 0.04	0.83 ± 0.07	0.91 ± 0.05	0.92 ± 0.05	0.85 ± 0.06

[a] Max-Margin-MCP yields among the best results in our experiments.

Chapter 4

OPTIMIZATION-BASED FEATURE SELECTION FOR CLASSIFICATION[1]

4.1 Introduction

While there are more and more available data (e.g., the genome data containing hundreds or thousands features [62, 14]), more sufficient information can be provided to solve classification and clustering problems. However, along the line, it may encounter a difficulty of computational complexity increased by data size and noise (including missing and erroneous values). To handle this, feature selection is a process to find a subset of "good" features from the original feature set [25, 64], which has been shown to be beneficial for the classification performance of learning models/algorithms in some applications [23, 39, 79, 55, 115]. In general, there are several advantages of feature selection in classification: data reduction, noise reduction, and interpretability. By reducing the feature space, a smaller feature subset obtained is more desirable to reduce the computational complexity and cost of learning a classification model/algorithm, and noise could be removed to improve the classification accuracy. Moreover, a small subset of the good features out of a huge quantity of features is favorable for interpretably identify the detailed characteristics or functions behind the problem.

To solve the feature selection problem, three types of approaches are usually suggested: filter, wrapper, and hybrid approach of filters and wrappers [85, 49, 122]. The filter approach can be viewed as a pre-processing step that selects a feature subset based

[1]The chapter is part of two working papers in collaboration with Wanpracha Art Chaovalitwongse, Chungmok Lee, Myong-K. Jeong, and Shouyi Wang

on the inherent characteristics of training data and does not convolve with any classification models. The wrapper approach searches for the candidate features according to the performance of a pre-determined classification model learnt by which features are selected. Furthermore, a hybrid approach combined filter and wrapper approaches is to use a goodness measure (e.g., classification accuracy or number of selected features) as an objective function in a classification framework to find a best combination of features [137, 39, 55].

Among statistic and machine learning methods, mathematical optimization provides a perspective in feature selection to improve the classification performance [31]. The objectives usually includes separation margin maximization [26, 43], classification accuracy maximization [39, 55], and others [60, 101]. In the chapter, we attempt to develop new optimization-based approaches to solve feature selection problem in classification. In the first part, we propose an optimization model, integrated with statistical information from features as inputs, solved by a hybrid heuristic algorithm. The optimal characterization of selected features relies on the separability with respect to the target class (maximum relevancy) and the correlation with other selected features (minimum redundancy). In the second part, we propose a pattern-based classification method, called decomposed feature support machine. The idea, modified from the feature support machine (FSM) [39], is to maximize the classification accuracy using a decomposed k-nearest neighbor rule (DKNN). The preliminary results for the test data sets are presented.

4.2 Separation-Correlation Feature Selection Using Statistical Information

Recently, mutual information (MI), first discussed in statistics and information theory in 1990s [89, 131], has been widely used as a criterion in feature selection to evaluate how good selected features are. Relevancy and redundancy are common criteria [143, 114, 54, 119]. A feature is expected to be individually selected with high relevancy with respect to the target class. Among all selected features, it is more likely

to have redundant features leading to the identical classification. Thus, a redundancy criterion is used to find a feature less correlated with the other selected features. Consequently, a combined criterion of maximum-relevancy and minimum-redundancy has been shown a more significant advance. Furthermore, many hybrid heuristic approaches were developed to search for top ranked features based on different mutual information criteria with pre-determined parameters such as number of selected features and selection threshold in the literature [20, 92, 129, 114, 54]. Along the research direction, there are some other studies that adopt the similar statistical concepts of correlation [66, 143] and divergence [90, 128] used for the feature selection and classification. Recently, a quadratic programming approach was proposed, integrated with similarity (correlation) measure [119]. The interested reader is referred to the literature for detailed reviews [98, 134].

In feature selection, however, it is shown that "good" individual features with higher information are not always a good combination yielding good classification performance. In fact, the feature selection can be viewed as a combinatorial optimization problem. In this study, we propose a new concept of combinational optimization using the mutual information to find a best and compact subsets of features that maximize the separability of individual selected features with respect to the target class and minimize the correlation with other selected features. On the other hand, it is computational expensive to solve such combinational optimization problem as the size and dimensionality of data increase drastically. We therefore develop an incremental heuristic search algorithm to solve it with selection criteria such as maximum-relevancy, minimum-redundancy, and maximum-relevancy-minimum-redundancy.

The remainder of this section is structured as follows. In Section 4.2.1, the background of statistical information in feature selection is reviewed. In Section 4.2.2, a new mathematical optimization model for feature selection is presented based on the mutual information and then an efficient heuristic algorithm is developed. In Section 4.2.3, the effectiveness of the proposed approach is tested for widely used data sets from the UCI machine learning repository and in the literature using several classification techniques, along with exhaustive computational experiments. This chapter is concluded in Section

4.4.

4.2.1 Statistical Information in Feature Selection

In this section, we review the background of mutual information that has been increasingly used for feature selection in data mining. Also, we briefly introduce the similar concepts of correlation and divergency.

We first define the following notations that will be used thorough out the paper. In this study, we mainly focus on the two-class data. Given a set I of samples in both positive and negative classes, where $I = I^+ \cup I^-$ and $\emptyset = I^+ \cap I^-$. Each sample is represented by a set of features, denoted by F. The target class of samples is denoted by C. A feature subset $S \subseteq F$ is defined a set selected from the whole feature set. Other variables will be defined later as needed. Moveover, we note that the terms "feature" and "variable" are used interchangeably sometimes.

4.2.1.1 Mutual Information

MI (also called cross entropy) is shown to be favorable in feature selection to measure the relationship (e.g., relevancy) of any two target random features. Consider two continuous random features X and Y, the MI between them is defined based on the probability distribution as follows.

$$I(X;Y) = \int_y \int_x p(x;y) \log \frac{p(x;y)}{p(x)p(y)} dxdy, \tag{4.1}$$

where $p(x)$ and $p(y)$ are the marginal probability density functions for X and Y, respectively, and $p(x,y)$ is the joint probability density function. It is defined for discrete features in a similar way as follows.

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x;y) \log \frac{p(x;y)}{p(x)p(y)} dxdy. \tag{4.2}$$

The MI has the following favorable properties due to using probability density function [89, 54]. The MI is capable of measuring any kind of features because it is not based on statistics of any order. The characterization and order of original variables are retained because the calculated value does not rely on the transformation in the variable space. In calculation, the value of MI is not upper bounded by 1 and dependent on intrinsic characteristic of variables. The two features are independent when $I(X;Y) = 0$ and get more relevant as $I(X;Y)$ increases. In addition, The pairwise MI is symmetric so that $I(X;Y) = I(Y;X)$.

Because the value of MI highly depends on features themselves, the measure should be revised in order to evaluate features at the same standard. [54] therefore proposed a concept of normalized mutual information (NMI) and the NMI is defined between feautres X and Y as the entropy of MI normalized by the minimum entropy of both features, given by

$$NI(X;Y) = \frac{I(X;Y)}{\min\{H(X),\ H(Y)\}}, \tag{4.3}$$

where $H(\cdot)$ is the entropy.

In feature selection, there are several useful criteria derived on the basis of mutual information as the goal of feature selection is to find the most correlated features with the target class. $I(f_j;C)$ is defined to quantify the relationship of an individual feature f_j with respect to the target class C, and $I(f_j;f_k)$ is defined to quantify the relationship between features f_j and f_k. A feature is selected among all considered features such that it has the maximum relevancy with the target class. The "Max-relevancy" (MR) selection criterion is defined as

$$\max_{f_j \in F} I(f_j;C). \tag{4.4}$$

To extend a single feature to any feature subset S, it can consider the average Max-relevancy for a feature subset given by

$$\max_{S \subseteq F} \frac{1}{|S|} \sum_{j \in S} I(f_j;C). \tag{4.5}$$

However, when selecting such relevant features, some of them may be correlated to each other and redundant to govern the discrimination of the target class. To avoid this redundancy, it needs to find out redundant features from the relevant features with the target class. A feature subset S with the least with the least correlation with other features is selected by "Min-redundancy" (mR) selection criterion, which is defined by

$$\min_{S \in F} \frac{1}{|S|^2} \sum_{f_j} \sum_{f_k \in S} I(f_j; f_k). \tag{4.6}$$

Consequently, a combined selection criterion, "Min-redundancy and Max-relevancy" (mRMR), was proposed to select a feature subset; each feature is highly dependent on the target class and very less correlated with the other features [51, 114]. The mRMR is defined as

$$\max_{S \subseteq F} \frac{1}{|S|} \sum_{j \in S} I(f_j; C) - \beta \frac{1}{|S|^2} \sum_{f_j} \sum_{f_k \in S} I(f_j; f_k), \tag{4.7}$$

where the parameter β controls the tradeoff between the two terms mR and MR.

With the selection criteria based on mutual information, many feature selection methods have been proposed over the past years such as MIFS [20], MIFS-U [92], AMIFS [129], mRMR [114], NMIFS [54]. The MIFS is the first MI-based approach to select a feature subset $S \subseteq F$ that maximizes the MI $I(S, C)$ using the MR criterion in Equation 4.5. The authors proposed a heuristic to iteratively select informative features based on the criterion given by

$$\max_{f_j \in F} I(f_j; C) - \beta \sum_{f_j \in F} \sum_{f_k \in S} I(f_j; f_k), \tag{4.8}$$

where $\beta = \frac{1}{|S|}$, until the pre-determined size of feature subset is met. Note that the above criterion is a reduction of Equation (4.7) with $f_1 \in F$ and $|F| = 1$. MIFS-U is an improved version of MIFS by changing the selection criterion. Later, an enhancement AMIFS over MIFS and MIFS-U was proposed, which considers an adaptive parameter β for the tradeoff of relevancy and redundancy. More recently, NMIFS is a comprehensive approach that takes into account all limitations appearing in the previous MI-based

approaches. The authors proposed an average normalized MI so that the parameter β is no longer considered a matter in the selection criterion and developed a new genetic algorithm to solve it. In addition, the mRMR is an approach that exactly uses the selection criterion in Equation (4.7). the authors have proven that the first-order selection is equivalent to "Maximum dependency", which considers the joint mutual information of features with respect to the target class, i.e., max $I(\{f_j,\ j = 1, ..., |S|\}; C)$. Then they proposed the first-order incremental search by using $\beta = 1$ to iteratively select one informative feature at a time. Moreover, they proposed a two-stage algorithm: in the first stage, a feature subset with pre-determined size is selected using the first-order incremental search and in the second stage, they consider both forward and backward wrapper scheme to finally reach a compact feature subset. Meanwhile, another approach using conditional mutual information was proposed for binary feature selection [57].

4.2.1.2 Correlation

Correlation is a classical measurement in statistics to evaluate the relevancy of two random variables. There are two correlation-based approaches widely used. The first approach is simply based on the classical linear correlation coefficient r. The r takes the values between -1 and 1. If $r < 0$, two features are negatively correlated while if $r > 0$, two features are positively correlated. If $r = 0$, two features are completely independent. In feature selection, it does not a matter if two features are positively or negatively correlated, so we take the absolute value of r and the pairwise correlation is defined $|r| \in [0,\ 1]$. However, the linearity property may not suitable to most real-life features because their correlation are not linear in nature. The second approach based on the information-theoretical concept of entropy is considered to overcome the above limitation. While features with highly relevancy with respect to the target class are easily identified, some studies recently proposed approaches to identify the features with significant redundancy using a correlation-based measure. The interested reader is referred to the rich literature [66, 67, 142, 143, 141, 48].

4.2.1.3 Divergence

Kullback-Leibler (KL) divergence (also called relative-entropy) is well-known as a measure to quantify the dissimilarity between two features [90, 89]. The definition of KL divergence is related to the MI. The entropy of KL is also calculated based on the probability distributions $p(x)$ and $p(y)$ of both discrete features given by

$$KL(X;Y) = \sum_{x \in X} p(x) \log \frac{p(x)}{p(y)}. \tag{4.9}$$

Since the KL divergence is not symmetric, the entropy from $p(x)$ to $p(y)$ is not equal to the entropy from $p(y)$ to $p(x)$. To avoid the bias of measurement, the Jensen-Shannon (JS) divergence [81] is generally considered by taking the average entropy given by

$$JS(X;Y) = 0.5(\sum_{x \in X} p(x) \log \frac{p(x)}{p(y)} + \sum_{y \in Y} p(y) \log \frac{p(y)}{p(x)}). \tag{4.10}$$

For use of divergence in feature selection, we propose a simple way for relevancy calculation. It can be used for the relevancy of a feature with respect to the target class. We separate the samples on the feature into two groups based on the class information. Because both sizes of the original features may not be equal, feature values of the samples in both groups are discretized on the same domain with a preset interval. In such way, we obtain two discrete probability distributions employed in Equation (4.10) to calculate the relevancy. It can be similarly used for the relevancy between two features.

4.2.2 The Proposed Optimization-based Approach

This section is divided into two parts. In the first part, we propose a new optimization model to find a compact subset of informative features based on statistical information as input (the mutual information is the main focus). In the second part, we propose an efficient algorithm to solve the proposed optimization problem as the computational complexity of the model is increased by the feature dimensionality.

4.2.2.1 Optimization Model

With the mutual information, the goal of the feature selection here is to select a subset $S \subseteq F$ of features; each individual feature has high relevancy on the target class C and has less relevancy with the other selected features $j \in S$. Ultimately, this feature subset is expected to yield a better performance in classification. How to find such a feature subset can be formulated as a combinatorial optimization problem with mutual information as an input. We define $y_j \in \{0, 1\}$ as a binary variable indicating if feature j is selected. Inputs of mutual inforamtion include: P is a $|J| \times 1$ vector, where p_j is a value of relevancy of feature $j \in J$ with respect to the target class, i.e., $p_j = I(f_j; C)$, and $Q = (q_{jk})$ is a $|J| \times |J|$ symmetric matrix, where q_{jk} is a value of pairwise relevancy between feature j and k, i.e., $q_{jk} = I(f_j; f_k)$. The quadratic programming formulation of SCOM-Q is given by

$$
\begin{aligned}
(\mathsf{SCOM\text{-}Q}) \quad \max \quad & \sum_{j \in J} p_j y_j && (4.11)\\
\text{s.t.} \quad & \sum_{j,\, k \in J} q_{jk} y_j y_k \leq \theta, && (4.12)\\
& \sum_{j \in J} y_j \geq \alpha, && (4.13)\\
& y_j \in \{0, 1\}, && (4.14)
\end{aligned}
$$

where $\theta \in [0, |J|^2]$ is a threshold to control the relevancy of selected features and α is a positive number to control the number of selected features. The objective in Equation (4.11) is to maximize the overall relevancy of selected features with respect to the target class. The constraint in Equation (4.12) ensures that the overall relevancy among selected features has to be less than a pre-determined threshold θ. The constraint in Equation (4.13) ensures that at least α features have to be selected. For simplicity, α is usually set to 1. Consequently, we employ a linearization technique by [38] to linearize the quadratic constraint in Equation (4.12). We define $u_j \geq 0$ as the total pairwise relevancy for feature j and $v_j \geq 0$ as a surplus variable. The linearized model

of SCOM-Q is reformulated by

$$\text{(SCOM-MI)} \quad \max \quad \sum_{j\in J} p_j y_j - \beta \sum_{j\in J} u_j \tag{4.15}$$

$$\text{s.t.} \quad \sum_{k\in J\setminus j} q_{jk} y_k = u_j + v_j \quad \forall j \in J, \tag{4.16}$$

$$v_j \leq M(1-y_j) \quad \forall j \in J, \tag{4.17}$$

$$\sum_{j\in J} y_j \geq \alpha, \tag{4.18}$$

$$y_j \in \{0,1\}, \tag{4.19}$$

where we delete the parameter θ and add a new parameter β as a tradeoff between two terms in the objective function. The objective in Equation (4.15) is to maximize the overall relevancy of selected features with respect to the target class and additionally to minimize the sum of the average relevancy of selected features. The constraints in Equation (4.16) is to calculate the total pairwise relevancy between feature j and the other selected features. The constraints in Equation (4.17) ensure that feature j is activated to select. The constraint in Equation (4.18) ensures that at least α features have to be selected.

It is worth noting that, the linearized model SCOM-L is more interpretable that each feature is selected with high relevancy with respect to the target class and less relevancy to all other selected features. It is obviously seen that the objective function in Equation (4.15) of SCOM-L is conceptually equivalent to the mRMR criterion in Equation (4.7). On the other hand, when solving the SCOM-L, it is difficult to calibrate the value of the parameter β at a precise level so that the two terms in the objective function are comparable. To tackle this issue, we employ the result of normalized mutual information proposed in [54] by replacing $I(f_j; f_k)$ with $NI(f_j; f_k) = \frac{I(f_j;f_k)}{\min\{H(f_j),\, H(f_k)\}}$. Thus, the parameter can be set to be a constant $\beta = \frac{1}{|J|}$, where $|J|$ is the cardinality of feature set J, without the calibration of β.

For generality, the redundancy and relevancy of selected features can be evaluated by the correlation and divergence as mentioned in Section 4.2.1. In SCOM-MI, the inputs can be simply replaced with the divergence $p_j = JS(f_j^+, f_j^-)$ for relevancy and

the correlation coefficient $q_{jk} = |r|$ for redundancy. The SCOM-MI is carried out to obtain a feature subset based on both statistical information.

To conclude, the proposed approach SCOM-MI has several advantages compared to the existing MI-based feature selection methods. Firstly, the SCOM-MI does not need to pre-determine the number of selected features, and always find a very compact selected feature subset. Whereas, other MI-based methods find a feature subset with a fixed size. This may lead to an additional step to remove irrelevant or redundant features. Secondly, the SCOM-MI also does not need to consider any threshold parameters to filter out features due to imposing the relief by normalized mutual information on the objective function. Finally, the resultant feature subset is considered an "optimal" combination of high informative features, not the first $|S|$ "good" features based on the mutual information. Most importantly, the model indeed considers the inter-effects among all candidate features. For example, given a subset of ordered features f_1, f_2, and f_3 based on the mRMR criterion. The best solution may be a combination of f_1 and f_2 substituted for f_1 although f_1 alone gives the highest mutual information.

4.2.2.2 Incremental Optimization Search Algorithm

It is challenging to obtain an optimal solution among $\sum_{k=1}^{|J|} \binom{|J|}{k}$ combinatorial solutions in a reasonable computational time as the data (feature) size increases drastically. For this reason, we propose an incremental optimization search algorithm (IOSA), integrated with the SCOM-MI to find the best combination of selected features. The idea of IOSA is to iteratively solve the SCOM-MI by adding one feature at a time; the feature to be added to the candidate feature set is selected based on the different criteria of mutual information, such as MR and mRMR. Note that here we redefine the candidate feature set S and the unselected feature set F. The entire iterative procedure is described in Algorithm 3. The classification accuracy is calculated by the percentage of samples correctly classified by a sophisticated classifier such as SVM, LDA, or KNN.

To determine a feature to be added to the candidate feature set, we propose three selection criteria as follows.

Algorithm 3 Incremental Optimization Search Algorithm

Input: a training subset with the whole feature set F.
Output: a best combination of selected features B with the associated classification accuracy.

procedure INCREMENTAL_OPTIMIZATION_SEARCH_ALGORITHM(input)
    Initialization: feature set $F = \{F\}$, candidate set $S = \emptyset$
    Choose the best feature $f_c^1 \in F$ in terms of the MR criterion.
    Update: $S \leftarrow S = \{f_c^1\}$ and $F \leftarrow F \backslash \{f_c^1\}$.
    repeat
        Determine a candidate feature f_c^t according to the given selection criterion.
        Update: $S \leftarrow S = \{f_c^t\}$ and $F \leftarrow F \backslash \{f_c^t\}$.
        Solve SCOM-MI with undated P and Q as inputs from S.
        Calculate classification accuracy on the training subset.
        Update: selected features B.
        Check stopping criterion: no improvement on either classification accuracy (as high as possible) or the number of selected features (as less as possible). If it is met, stop search.
    until $t > T$
return output
end procedure

IOSA-1 Consider a feature f_c^t having maximum relevancy among all unselected features with respect to the target class C. The criterion is given by

$$MI_1 : \ t = \arg\max_{f_j \in F} \{I(f_i; C) | f_j \in F\}. \tag{4.20}$$

IOSA-2 Consider a feature f_c^t having maximum relevancy among all unselected features with respect to the target class C and minimum total pairwise relevancy (that is, minimum redundancy) with other unselected features. The criterion is given by

$$MI_2 : \ t = \arg\max_{f_j \in F} \{I(f_j; C) - \frac{1}{|F| - 1} \sum_{f_k \in F \backslash f_j} NI(f_j; f_k) | f_j, \ f_k \in F\}. \tag{4.21}$$

IOSA-3 Consider a feature f_c^t having maximum relevancy among all unselected features with respect to the target class C and minimum total pairwise relevancy (that is, minimum redundancy) with all candidate features. The criterion is given

by

$$MI_3: \ t = \arg\max_{f_j \in F} \{ I(f_j; C) - \frac{1}{|S|} \sum_{f_k \in S} NI(f_j; f_k) | f_j \in F, \ f_k \in S \}. \qquad (4.22)$$

The proposed algorithm IOSA can be viewed as a hybrid forward selection scheme, in which a filter approach (i.e., SCOM-MI) to select a good combination of features is convolved in a wrapper approach to evaluate the classification performance. Compared to the existing MI-based feature selection algorithms such as MIFS, NMIFS, mRMR, etc., the solving process is relatively fast although the cost of computational complexity does not change very much. One reason is that in the IOSA framework, the SCOM-MI only targets at the candidate feature set starting from 2 features and the size is much smaller.

4.2.3 Preliminary Experiments

To test the proposed feature selection approaches, we use several state-of-the-art classification techniques for a number of data sets. All experiments were implemented on Intel Xeon Quad Core 3.0GHz processor workstation with 8 GB RAM and were coded in MATLAB with synchronization of CPLEX version 10.0 in GAMS. For classification techniques, we directly employ the developed MATLAB toolboxes. Computational times reported were obtained from the desktop's internal timing calculations, which include the time used for preprocessing and postprocessing.

4.2.3.1 Data sets

we use 6 two-class data sets from the UCI repository [16]: Cleveland and Statlog heart disease, Wisconsin breast cancer, bupa liver disorders, Pima Indians diabetes, and Parkinson's disease. For the purpose of noise reduction, the data is discretized into a binary format by using a threshold for each feature. The threshold c_j for feature j is determined by the average of the means μ_j^+ and μ_j^- of the subsets in two classes, i.e., $c_j = \frac{\mu_j^+ + \mu_j^-}{2}$. For example, a binarized feature value $x = 1$ if $x \geq c_j$, and $x = 0$ otherwise.

Table 4.1: Characteristics of data sets.

Data	Samples			Class	Features
	Total	+	−	(+, −)	
Breast cancer-Wisconsin	699	458	241	(malignant, benign)	9
Heart disease-Cleveland	303	139	164	(sick, normal)	13
Heart disease-Statlog	270	120	150	(sick, normal)	13
Bupa liver disorders	325	200	125	(positive, negative)	6
Pima Indians Diabetes	768	268	500	(positive, negative)	8
Parkinson's disease	195	147	48	(parkinsons, normal)	22
Leukemia	72	47	25	(ALL, AML)	7070
Colon Cancer	62	40	22	(tumor, normal)	2000

Note that the feature space still remains the same. We also use two biomedical data sets for comparison: leukemia [62] and colon cancer [14], that have been tested in [114]. For the reason to reduce noise, the obtained data sets have been preprocessed by discretizing all feature values into 3-state values with respect to the mean μ and standard deviation σ. For example, a feature value $x = 2$ if $x > \mu + \sigma/2$, $x = 0$ if $\mu - \sigma/2 \leq x \leq \mu + \sigma/2$, and $x = -2$ if $x < \mu - \sigma/2$. The characteristics of all data sets are summarized in Table 4.1.

In addition, missing values usually appear in real-life data due to sampling errors. According to the literature [73], the feature selection and the construction of classification model are susceptible to missing values in the data analysis. To remove this effect, we consider missing values in one class to be simply replaced by the means of the feature values in the opposite class.

Synthetic Data sets

To show the effect of increase in data size on the solving process, we create new data sets of 100, 500, and 1000 features by adding new artificial features to the original Wisconsin breast cancer and Parkinson's disease data sets. The sample sizes of both new data sets remain unchanged. New feature values are randomly generated from the uniform distribution over the interval $[ub, lb]$, where ub is a random integral number between 1 and 50 and lb is a random integral number between 51 and 100. A set of feature values is generated with the same seed.

4.2.3.2 Classifiers

The proposed feature selection approaches in this paper do not convolve with any specific classification models or techniques. To test the effectiveness of selected features, we therefore choose three widely used classification techniques such as linear discriminate analysis (LDA) [108], k nearest neighbor rule (KNN) [46], and support vector machine (SVM) [125]. We directly employ the MATLAB toolboxes with specific parameter settings. LDA is to find a combination (or classification boundary) of features that can distinguish one class data from the other class data. We choose a linear type of discriminant function. KNN is amongst the simplest machine learning algorithm. The idea is that a sample is classified based on the closest known (training) k samples, where k is the number of samples used for comparison. Here we propose to use $k = \min\{|I^+|, |I^-|\}$. SVM is a mathematical programming technique for classification, which constructs a boundary in highly dimensional feature space with a kernel function based on all known samples. We use the LIBSVM package [35] to implement experiments using a linear type of kernel function.

4.2.3.3 Performance Accuracy

To evaluate the classification performance, we adopt the accuracy that is defined by the percentage of correctly classified samples to the target sample set. Let us denote sets of correctly classified positive and negative objects by I_c^+ and I_c^-. The accuracy is calculated by $accur = \frac{|I_c^+|+|I_c^-|}{|I|}$. Due to possible influences by the unbalanced or contaminated data, we repeat n times k-fold cross validation on randomly shuffled data sets in order to obtain unbiased outcomes. For cross validation, a target data set is equally divided into k subsets, in which one of subsets is used as a testing data set while the remaining $k-1$ subsets are used as a training data set. The classification accuracy is referred to the accuracy in terms of the testing data set to validate the effectiveness of the selected features from which the training data set is used to learn. The overall accuracy is reported by the average of $n \times k$ experiments, where $n = 10$ and $k = 5$ are set in our experiments. Note that we use the leave-one-out cross validation in order for

comparison with the mRMR approach in [51] on the leukemia and colon cancer data sets. One sample is left out as a testing sample while the remaining samples in the data set is used as a training subset at a time. The entire cross validation process is bounded by the number of samples.

4.2.3.4 Computational Results

Tables 4.2 and 4.3 display the results of classification performance of the proposed approaches obtained by 10 times 5-fold cross validation using classifiers LDA, KNN, and SVM for both original and binarized UCI data sets. The first column presents the baseline results by using all features in classification for comparison purpose. The second column presents the results of solving SCOM-MI directly. The remaining columns present the results of solving SCOM-MI by the heuristic searches with MI_1, MI_2, and MI_3 selection criteria. The accuracy is calculated based on the testing data subsets. It is clearly seen that compared to the baseline, there are relatively less features selected by the proposed approaches for classifiers to yield very competitive classification performance. Among all the proposed approaches, the heuristic search with MI_3 selection criterion gives better performance in terms of both the number of features and classification accuracy. There is no significant difference among all applied classifiers. Because the data may be contaminated by the noises (sampled errors), the significant classification improvement is obtained for the binarized data sets of the Cleveland and statlog heart disease compared to the original data sets.

Figure 4.1 illustrates the behaviors of accuracy and number of selected features over the feature iteration by the heuristic search. It shows how the search terminates when there is no improvement on either the classification accuracy or the number of selected features. We observed that not all features in the candidate subset are selected in the optimal combination in each iteration. It reflects the fact that more or better selected features turns out to be a better performance.

To show the capability of the proposed heuristic search to improve computational efficiency, we use larger synthetic data sets of Wisconsin breast cancer and Parkinson's

Table 4.2: Comparison results of three heuristic searches, IOSA-1-SCOM-MI, IOSA-2-SCOM-MI and ISOA-3-SCOM-MI for the original data sets using classifiers LDA (top), KNN (middle), and SVM (bottom).

	Baseline		SCOM-MI		IOSA-1-SCOM-MI		IOSA-2-SCOM-MI		IOSA-3-SCOM-MI	
Data	features	accur	features	accur	features	accur	features	accur	features	accur
Breast cancer-Wisconsin	9	0.96 ± 0.02	7.6	0.96 ± 0.02	4.2	0.95 ± 0.02	2.96	0.95 ± 0.02	4.5	0.95 ± 0.02
Heart disease-Cleveland	13	0.83 ± 0.04	4.4	0.79 ± 0.05	2.1	0.72 ± 0.08	3.3	0.79 ± 0.07	4.4	0.78 ± 0.07
Heart disease-statlog	13	0.84 ± 0.04	3.8	0.78 ± 0.06	1.9	0.71 ± 0.08	3.1	0.79 ± 0.07	3.8	0.76 ± 0.06
Bupa liver disorders	6	0.62 ± 0.06	1.9	0.58 ± 0.04	1.3	0.54 ± 0.06	1.24	0.53 ± 0.06	1.6	0.56 ± 0.06
Pima Indians diabetes	8	0.76 ± 0.03	2.0	0.73 ± 0.03	1.6	0.64 ± 0.04	1.78	0.72 ± 0.05	1.8	0.73 ± 0.03
Parkinsons	22	0.83 ± 0.04	7.9	0.75 ± 0.06	1.7	0.73 ± 0.07	1.74	0.74 ± 0.07	1.6	0.76 ± 0.07

	Baseline		SCOM-MI		IOA-1-SCOM-MI		IOA-2-SCOM-MI		IOA-3-SCOM-MI	
Data	features	accur	features	accur	features	accur	features	accur	features	accur
Breast cancer-Wisconsin	9	0.94 ± 0.02	7.5	0.94 ± 0.02	3.3	0.92 ± 0.02	3.0	0.92 ± 0.02	2.9	0.93 ± 0.02
Heart disease-Cleveland	13	0.67 ± 0.06	4.3	0.64 ± 0.06	1.7	0.68 ± 0.08	3.0	0.77 ± 0.05	4.3	0.71 ± 0.11
Heart disease-statlog	13	0.68 ± 0.06	3.7	0.66 ± 0.06	1.7	0.69 ± 0.09	2.6	0.76 ± 0.06	3.4	0.66 ± 0.09
Bupa liver disorders	6	0.64 ± 0.07	1.9	0.61 ± 0.05	1.6	0.58 ± 0.07	1.6	0.59 ± 0.07	1.6	0.60 ± 0.06
Pima Indians diabetes	8	0.65 ± 0.04	2.0	0.65 ± 0.04	1.3	0.65 ± 0.03	1.1	0.74 ± 0.04	1.0	0.74 ± 0.03
Parkinsons	22	0.83 ± 0.06	7.8	0.85 ± 0.06	1.7	0.81 ± 0.07	1.6	0.80 ± 0.07	1.5	0.85 ± 0.07

	Baseline		SCOM-MI		IOA-1-SCOM-MI		IOA-2-SCOM-MI		IOA-3-SCOM-MI	
Data	features	accur	features	accur	features	accur	features	accur	features	accur
Breast cancer-Wisconsin	9	0.96 ± 0.01	7.6	0.97 ± 0.01	5.2	0.96 ± 0.01	4.5	0.96 ± 0.01	5.7	0.96 ± 0.02
Heart disease-Cleveland	13	0.83 ± 0.05	4.3	0.76 ± 0.06	2.0	0.73 ± 0.07	3.4	0.79 ± 0.06	4.3	0.78 ± 0.07
Heart disease-statlog	13	0.83 ± 0.05	3.7	0.76 ± 0.06	2.1	0.72 ± 0.07	2.8	0.79 ± 0.06	4.0	0.77 ± 0.05
Bupa liver disorders	6	0.65 ± 0.05	1.9	0.58 ± 0.05	1.1	0.58 ± 0.05	1.2	0.57 ± 0.05	1.1	0.58 ± 0.04
Pima Indians diabetes	8	0.76 ± 0.03	2.0	0.74 ± 0.03	1.4	0.66 ± 0.03	1.4	0.74 ± 0.03	1.3	0.74 ± 0.03
Parkinsons	22	0.83 ± 0.06	7.9	0.84 ± 0.06	1.6	0.82 ± 0.06	1.5	0.81± 0.07	1.6	0.84 ± 0.06

Table 4.3: Comparison results of three heuristic approaches, IOA-1-SCOM-MI, IOSA-2-SCOM-MI and IOSA-3-SCOM-MI for the binarized (2-state) data sets using classifiers LDA (top), KNN (middle), and SVM (bottom).

	Baseline		SCOM-MI		IOSA-1-SCOM-MI		IOSA-2-SCOM-MI		IOSA-3-SCOM-MI	
Data	features	accur	features	accur	features	accur	features	accur	features	accur
Breast cancer-Wisconsin	9	0.96 ± 0.02	5.3	0.95 ± 0.02	2.7	0.93 ± 0.03	2.3	0.93 ± 0.03	1.6	0.92 ± 0.03
Heart disease-Cleveland	13	0.83 ± 0.04	9.4	0.83 ± 0.04	2.1	0.72 ± 0.06	4.1	0.79 ± 0.07	3.1	0.84 ± 0.04
Heart disease-statlog	13	0.84 ± 0.04	9.5	0.83 ± 0.05	3.3	0.78 ± 0.05	3.4	0.79 ± 0.05	3.2	0.85 ± 0.05
Bupa liver disorders	6	0.62 ± 0.06	3.0	0.55 ± 0.06	1.7	0.53 ± 0.05	1.6	0.54 ± 0.05	2.1	0.56 ± 0.05
Pima Indians diabetes	8	0.76 ± 0.03	4.6	0.73 ± 0.03	1.1	0.64 ± 0.04	2.0	0.70 ± 0.04	2.1	0.72 ± 0.03
Parkinsons	22	0.83 ± 0.04	8.7	0.78 ± 0.06	2.6	0.74 ± 0.08	1.9	0.70 ± 0.08	2.8	0.77 ± 0.06

	Baseline		SCOM-MI		IOA-1-SCOM-MI		IOA-2-SCOM-MI		IOA-3-SCOM-MI	
Data	features	accur	features	accur	features	accur	features	accur	features	accur
Breast cancer-Wisconsin	9	0.94 ± 0.02	5.2	0.93 ± 0.03	1.3	0.91 ± 0.03	1.4	0.90 ± 0.04	1.1	0.92 ± 0.03
Heart disease-Cleveland	13	0.67 ± 0.06	9.3	0.82 ± 0.05	3.3	0.78 ± 0.06	3.7	0.79 ± 0.06	3.0	0.84 ± 0.04
Heart disease-statlog	13	0.68 ± 0.06	9.6	0.82 ± 0.04	3.4	0.78 ± 0.08	3.4	0.80 ± 0.07	3.0	0.85 ± 0.04
Bupa liver disorders	6	0.64 ± 0.07	3.1	0.57 ± 0.04	1.8	0.57 ± 0.05	1.7	0.58 ± 0.05	2.2	0.58 ± 0.06
Pima Indians diabetes	8	0.65 ± 0.04	4.6	0.71 ± 0.04	1.1	0.65 ± 0.03	2.0	0.70 ± 0.04	2.2	0.72 ± 0.04
Parkinsons	22	0.83 ± 0.06	8.7	0.84 ± 0.05	2.5	0.84 ± 0.06	2.2	0.82 ± 0.09	2.8	0.87 ± 0.05

	Baseline		SCOM-MI		IOA-1-SCOM-MI		IOA-2-SCOM-MI		IOA-3-SCOM-MI	
Data	features	accur	features	accur	features	accur	features	accur	features	accur
Breast cancer-Wisconsin	9	0.96 ± 0.02	5.4	0.95 ± 0.02	3.2	0.93 ± 0.02	2.8	0.93 ± 0.02	1.9	0.93 ± 0.03
Heart disease-Cleveland	13	0.83 ± 0.04	9.5	0.82 ± 0.04	2.6	0.77 ± 0.05	2.6	0.77 ± 0.06	3.1	0.84 ± 0.05
Heart disease-statlog	13	0.83 ± 0.05	9.3	0.83 ± 0.05	2.4	0.77 ± 0.05	2.4	0.78 ± 0.06	3.2	0.85 ± 0.04
Bupa liver disorders	6	0.63 ± 0.07	3.0	0.56 ± 0.06	1.4	0.56 ± 0.06	1.7	0.55 ± 0.05	1.6	0.57 ± 0.06
Pima Indians diabetes	8	0.72 ± 0.03	4.6	0.72 ± 0.04	1.1	0.65 ± 0.04	1.8	0.71 ± 0.04	1.9	0.72 ± 0.03
Parkinsons	22	0.81 ± 0.05	8.6	0.86 ± 0.05	2.5	0.84 ± 0.06	2.3	0.84 ± 0.06	2.5	0.86 ± 0.07

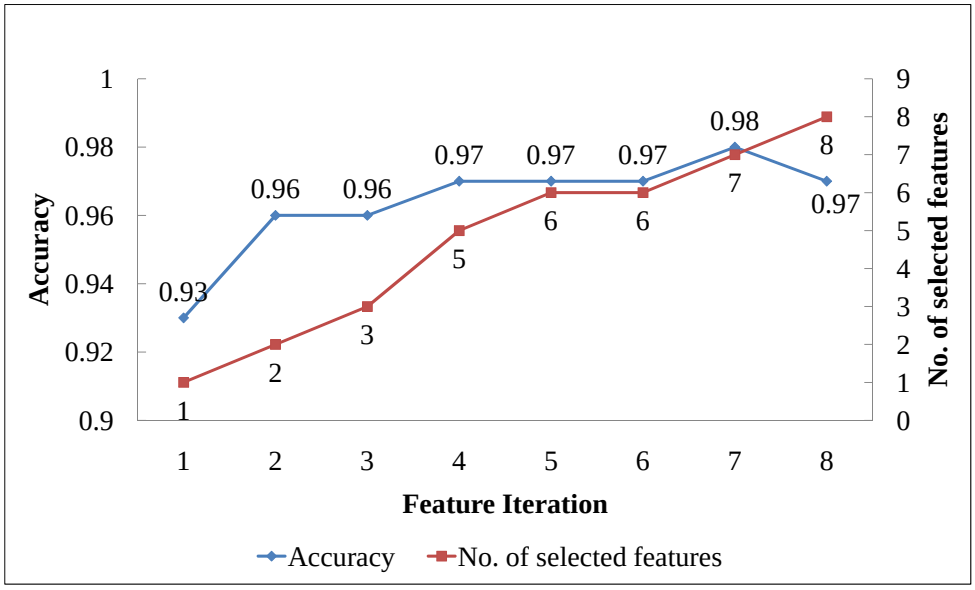

Figure 4.1: Behaviors of accuracy and number of selected features over the feature iteration. The iteration is terminated based on the stopping criterion that there is no improvement on classification accuracy when adding more features. The data used for illustration is a training subset of the breast cancer data set.

disease. Tables 4.4 and 4.5 shows the performance of the SCOM-MI and the heuristic search with MI_3 selection criterion in terms of the number of selected features, accuracy, and computational time. The reason that we only apply the heuristic approach with MI_3 selection criterion is because it overall outperforms the other two selection criteria. Note that the blank part means not results obtained because it ran out of computational time (20 hours for each cross validation). Obviously, the computational time increases with the number of features. Applying the heuristic approach to the SCOM-MI indeed saves a lot more computational time than solving the SCOM-MI directly. For the heuristic approach, we obtained the consistent results no matter if the data size increases; it always selects the most reliable (original) features as the artificial features are treated as contaminated features. Whereas, directly solving the SCOM-MI is not tractable for larger data sets.

Table 4.4: Comparison results of the SCOM-MI and IOSA-3-SCOM-MI for larger synthetic data set of Wisconsin breast cancer using classification techniques LDA (top), KNN (middle), and SVM (bottom).

	SCOM-MI			IOSA-3-SCOM-MI		
Data size	features	accur	Time [a]	features	accur	Time [a]
9	7.6	0.96 ± 0.02	1	4.54	0.95 ± 0.02	6
100	13.14	0.96 ± 0.02	2604	5	0.95 ± 0.02	16
500	[b]			4.76	0.95 ± 0.02	53
1000				5.42	0.95 ± 0.02	129

Data size	features	accur	Time [a]	features	accur	Time [a]
9	7.5	0.94 ± 0.02	2	2.9	0.93 ± 0.02	5
100	13.08	0.74 ± 0.05	2917	2.68	0.93 ± 0.02	9
500				3.04	0.93 ± 0.02	17
1000				2.62	0.93 ± 0.02	23

Data size	features	accur	Time [a]	features	accur	Time [a]
9	7.56	0.97 ± 0.01	9	5.74	0.96 ± 0.02	33
100	13.08	0.96 ± 0.02	3430	5.32	0.95 ± 0.02	57
500				5.82	0.96 ± 0.02	124
1000				6.2	0.96 ± 0.02	221

[a] The computational time (in second) is reported by the average time of 10 repetitions of cross validation.
[b] The blank means no results obtained because the experiments ran out of time limitation (20 hours for each cross validation).

Table 4.5: Comparison results of the SCOM-MI and IOSA-3-SCOM-MI for larger synthetic data set of Parkinson's disease using classification techniques LDA (top), KNN (middle), and SVM (bottom).

	SCOM-MI			IOSA-3-SCOM-MI		
Data size	features	accur	Time [a]	features	accur	Time [a]
22	7.9	0.75 ± 0.06	96	1.62	0.76 ± 0.07	275
100	[b]			1.48	0.76 ± 0.08	314
500				1.68	0.76 ± 0.06	602
1000				1.66	0.76 ± 0.06	843

Data size	features	accur	Time [a]	features	accur	Time [a]
22	7.82	0.85 ± 0.06	92	1.5	0.85 ± 0.07	256
100	18.94	0.83 ± 0.05	23720	1.46	0.84 ± 0.06	326
500				1.34	0.83 ± 0.07	470
1000				1.38	0.84 ± 0.06	599

Data size	features	accur	Time [a]	features	accur	Time [a]
22	7.88	0.84 ± 0.06	303	1.58	0.84 ± 0.05	529
100				1.58	0.85 ± 0.05	631
500				1.72	0.85 ± 0.06	920
1000				1.58	0.84 ± 0.06	1098

[a] The computational time (in second) is reported by the average time of 10 repetitions of cross validation.
[b] The blank means no results obtained because the experiments ran out of time limitation (20 hours for each cross validation).

Table 4.6: Comparison results of the proposed approaches SCOM-MI, IOSA-1-SCOM-MI, and IOSA-3-SCOM-MI with the mRMR method [114] for the discretized (3-state) data sets using classification techniques LDA (top), KNN (middle), and SVM (bottom).

	SCOM-MI		IOSA-1-SCOM-MI		IOSA-3-SCOM-MI		mRMR	
Data	features	accur[a]	features	accur	features	accur	features	accur
Leukemia	x[b]	x	2.4	42/72	4.68	67/72	5.0	71/72
Colon cancer	x	x	1.3	30/62	2.08	50/62	2.0	54/62

	SCOM-MI		IOA-1-SCOM-MI		IOA-3-SCOM-MI		mRMR	
Data	features	accur	features	accur	features	accur	features	accur
Leukemia	x	x	1.2	38/72	3.03	70/72	–[c]	–
Colon cancer	x	x	1.8	31/62	2.74	51/62	–	–

	SCOM-MI		IOA-1-SCOM-MI		IOA-3-SCOM-MI		mRMR	
Data	features	accur	features	accur	features	accur	features	accur
Leukemia	x	x	1.0	47/72	2.96	68/72	3.0	68/72
Colon cancer	x	x	1.2	38/62	2.18	51/62	2.0	54/62

[a] The accuracy is the number of correctly classified samples by leave-one-out cross validation.
[b] The program ran out of memory.
[c] The result is not available in the original paper.

4.2.3.5 Comparison with mRMR Feature Selection Method

We compare the performance of the proposed approaches (besides IOSA-2-SCOM-MI) with the mRMR method in [114] for the same large gene expression data sets in Table 4.6. The accuracy is reported as the number of correctly classified samples using leave-one-out cross validation. Due to high dimensionality of feature space, it ran out of memory as solving the SCOM-MI directly for all instances. The MR and mR selection criteria used in the heuristic search are not effective to achieve the same performance as the mRMR selection criterion. Compared to the mRMR approach, our heuristic search with the MI_3 selection criterion obtains competitive performance when both use the similar selection criterion. Note that their approach needs to pre-determine the number of features to be selected (and our approach do not), so the reported results is extracted from the implementation with the approximate number of selected features as what we obtained.

4.2.3.6 Generalization of the Proposed Framework

In Tables 4.8 and 4.8, we also present the comparison results by the SCOM using different statistical information such as the correlation coefficient and the divergence. The first column presents the result by the SCOM-MI (using mutual information)

Table 4.7: Comparison results of SCOM-MI and SCOM-Cor-Div for the original data sets using classifiers LDA (top), KNN (middle), and SVM (bottom).

	SCOM-MI		SCOM-Cor-Div	
Data	features	accur	features	accur
Breast cancer-Wisconsin	5.3	0.95 ± 0.02	4.1	0.95 ± 0.02
Heart disease-Cleveland	9.4	0.83 ± 0.04	3.6	0.63 ± 0.06
Heart disease-statlog	9.5	0.83 ± 0.05	5.4	0.73 ± 0.07
Bupa liver disorders	3.0	0.55 ± 0.06	3.6	0.63 ± 0.06
Pima Indians diabetes	4.6	0.73 ± 0.03	5.4	0.75 ± 0.04
Parkinsons	8.7	0.78 ± 0.06	12.0	0.77 ± 0.05

	SCOM-MI		SCOM-Cor-Div	
Data	features	accur	features	accur
Breast cancer-Wisconsin	5.2	0.93 ± 0.03	3.9	0.92 ± 0.02
Heart disease-Cleveland	9.3	0.82 ± 0.05	5.4	0.63 ± 0.06
Heart disease-statlog	9.6	0.82 ± 0.04	5.3	0.64 ± 0.04
Bupa liver disorders	3.1	0.57 ± 0.04	3.5	0.63 ± 0.06
Pima Indians diabetes	4.6	0.71 ± 0.04	5.4	0.65 ± 0.04
Parkinsons	8.7	0.84 ± 0.05	12.0	0.85 ± 0.05

	SCOM-MI		SCOM-Cor-Div	
Data	features	accur	features	accur
Breast cancer-Wisconsin	5.4	0.95 ± 0.02	4.1	0.95 ± 0.02
Heart disease-Cleveland	9.5	0.82 ± 0.04	5.4	0.73 ± 0.06
Heart disease-statlog	9.3	0.83 ± 0.05	5.2	0.72 ± 0.06
Bupa liver disorders	3.0	0.56 ± 0.06	3.5	0.60 ± 0.07
Pima Indians diabetes	4.6	0.72 ± 0.04	5.4	0.76 ± 0.04
Parkinsons	8.6	0.86 ± 0.05	12.0	0.82 ± 0.05

and the second column presents the results by the SCOM-Cor-Div (using correlation coefficient and divergence). We observed that the SCOM-Cor-Div gives worse accuracy ($\tilde{2}$0% lower) for the original data sets of Wisconsin and statlog heart disease while better accuracy for the original data sets of Bupa liver disorders. For the binarized data sets, there is no significant difference for all instances. It is noted that the SCOM-MI used relatively less features than the SCOM-Cor-Div to achieve the same classification performance.

4.3 Decomposed Feature Support Machine

Feature support machine (FSM), motivated by the support vector machines (SVM) in [31], is a pattern-based classification approach based on a nearest neighbor rule and was first applied to abnormal brain activity classification problem [39]. The goal of the FSM is to maximize the classification accuracy (or minimize the classification error) by selecting "good" features that have strong separability with respect to the target class. A nearest neighbor rule is used to identify (vote) the class of samples according

Table 4.8: Comparison results of SCOM-MI and SCOM-Cor-Div for the binarized data sets using classifiers LDA (top), KNN (middle), and SVM (bottom).

	SCOM-MI		SCOM-Cor-Div	
Data	features	accur	features	accur
Breast cancer-Wisconsin	5.3	0.95 ± 0.02	4.8	0.94 ± 0.02
Heart disease-Cleveland	9.4	0.83 ± 0.04	8.1	0.84 ± 0.04
Heart disease-statlog	9.5	0.83 ± 0.05	8.2	0.84 ± 0.04
Bupa liver disorders	3.0	0.55 ± 0.06	3.6	0.56 ± 0.06
Pima Indians diabetes	4.6	0.73 ± 0.03	4.2	0.72 ± 0.03
Parkinsons	8.7	0.78 ± 0.06	x	x

	SCOM-MI		SCOM-Cor-Div	
Data	features	accur	features	accur
Breast cancer-Wisconsin	5.2	0.93 ± 0.03	4.9	0.93 ± 0.03
Heart disease-Cleveland	9.3	0.82 ± 0.05	8.1	0.81 ± 0.04
Heart disease-statlog	9.6	0.82 ± 0.04	8.4	0.82 ± 0.05
Bupa liver disorders	3.1	0.57 ± 0.04	3.6	0.58 ± 0.06
Pima Indians diabetes	4.6	0.71 ± 0.04	4.2	0.72 ± 0.04
Parkinsons	8.7	0.84 ± 0.05	9.2	0.81 ± 0.05

	SCOM-MI		SCOM-Cor-Div	
Data	features	accur	features	accur
Breast cancer-Wisconsin	5.4	0.95 ± 0.02	4.9	0.94 ± 0.02
Heart disease-Cleveland	9.5	0.82 ± 0.04	8.0	0.83 ± 0.05
Heart disease-statlog	9.3	0.83 ± 0.05	8.3	0.84 ± 0.04
Bupa liver disorders	3.0	0.56 ± 0.06	3.6	0.57 ± 0.06
Pima Indians diabetes	4.6	0.72 ± 0.04	4.2	0.72 ± 0.03
Parkinsons	8.6	0.86 ± 0.05	9.2	0.84 ± 0.06

to the nature of the closest baseline samples. The FSM is modeled as follows. We define the following sets and decision variables. I is a set of target samples and J is a set of features. Binary variable $y_j = 1$ indicates if feature j is selected by FSM, and 0 otherwise. Binary variable $x_i = 1$ indicates if sample i is correctly classified. Given a data set, an input parameter is defined: accuracy matrix A, where $a_{ij} = 1$ indicates if the nearest neighbor rule correctly classifies sample i on feature j, and 0 otherwise. The mathematical program for the FSM is given by

$$\text{(V-FSM)} \quad \max \quad \sum_{i \in I} x_i \tag{4.23}$$

$$\text{s.t} \quad \sum_{j \in J} (a_{ij} - \frac{1}{2}) y_j \leq M x_i \quad \forall i \in I \tag{4.24}$$

$$\sum_{j \in J} (\frac{1}{2} - a_{ij}) y_j + \epsilon \leq M(1 - x_i) \quad \forall i \in I \tag{4.25}$$

$$\sum_{j \in J} y_j \geq 1 \tag{4.26}$$

$$y_j,\ x_i \in \{0, 1\}, \tag{4.27}$$

where $M = |J|/2$ and $0 < \epsilon < 1/2$ is used to break a tie during the voting. The objective in Equation (4.23) is to maximize the total number of correctly classified samples. There are two constraint sets in Equations (4.24) and (4.25) used to ensure that the samples are classified based on the nearest neighbor rule. A logical constraint in Equation (4.26) ensures that at least one feature is used in the nearest neighbor rule for classification. Later, a relaxation of FSM (rFSM) was proposed [55], which relaxes the binary decision variable y_j to $y_j \in \mathbf{R}$ and applied to medical prognosis and diagnosis.

Here we propose a new FSM, called decomposed feature support machine (dFFM). Specifically, a decomposed k-nearest neighbor (DKNN) rule is proposed and applied to identify (vote) the class of samples so as to construct an accuracy matrix.

4.3.1 Accuracy Matrix by Decomposed k-Nearest Neighbor

A k-nearest neighbor (KNN) rule is an intuitive and effective technique, which has long been associated to classification. Traditionally, the KNN rule identifies the class of a sample through the majority voting based on the closest k samples in the baseline data set. A DKNN rule is proposed in a similar way. The idea is that a baseline data set is divided into two subsets (positive and negative) and the class of a sample is identified through the majority voting based on distances to the closest k samples from both positive and negative data subsets. The distance is measured between the sample and the average of the closest k samples. The sample is assigned (voted) to closer subset.

In terms of computational complexity, a KDNN rule runs faster than a KNN rule because $O(|N| \cdot |D| + |N|^2) \leq O(|N| \cdot |D| + |N^+|^2 + |N^-|^2)$, where N is the baseline data set, D is the data dimension (feature space), and $N = N^+ \cup N^-$, where N^+ is positive baseline data subset and N^- is negative baseline data subset. Moreover, it is easy to prove that both KDNN and KNN rules equivalently yields identical classification.

An accuracy matrix A is constructed based on the baseline data set in the training stage. For clarity, the baseline data set here refers to a subset of the training data set. The nature of the training samples is known. The classification of every feature of every

training samples can be directly identified by applying the proposed KDNN rule with a pre-determined parameter k. In the voting scheme, $a_{ij} = 1$ indicates if the KDNN correctly classifies sample i on feature j, and 0 otherwise. The correct classification is based on average intra-class distance d_{ij} and average inter-class distance $\bar{d}_{ij}$ of positive and negative baseline data subsets. That is, if $d_{ij} < \bar{d}_{ij}$, then $a_{ij} = 1$, and 0 otherwise.

4.3.2 Optimization Models

Given an accuracy matrix A. We denote I^+ is a positive data subset and I^- is a negative data subset, where $I^+ \cap I^- = I$ and $I^+ \cap I^- = \emptyset$. The mathematical program for the accuracy maximization problem is given by

$$\begin{aligned}
(\mathsf{VAMM}) \quad \max \quad & \frac{\alpha}{|I^+|} \sum_{i \in I^+} x_i + \frac{1-\alpha}{|I^-|} \sum_{i \in I^-} x_i && (4.28)\\
\text{s.t} \quad & \sum_{j \in J} (a_{ij} - \frac{1}{2}) y_j \leq M x_i \quad \forall i \in I && (4.29)\\
& \sum_{j \in J} (\frac{1}{2} - a_{ij}) y_j + \epsilon \leq M(1 - x_i) \quad \forall i \in I && (4.30)\\
& \sum_{j \in J} y_j \geq 1 && (4.31)\\
& x_i,\ y_j \in \{0, 1\}, && (4.32)
\end{aligned}$$

where $0 < \alpha < 1$ is an important weight of true positive ratio, $M = |J|/2$, and $0 < \epsilon < 1/2$ is used to break a tie during the voting. The objective in Equation (4.28) is to maximize the sum of weighted classification accuracies of both positive and negative classes. The constraint sets in Equations (4.29) and (4.30) ensure that the training samples are classified based on the k-nearest neighbor rule. The logical constraint in Equation (4.31) ensures that at least one feature is used in the k-nearest neighbor rule.

On the other hand, with the accuracy matrix A, we also can consider classification error minimization. Binary variable $z_i = 1$ indicates if sample i is not correctly classified by a KDNN rule, and 0 otherwise. The mathematical program for the error

minimization problem is given by

$$\text{(VEMM)} \quad \min \quad \frac{\beta}{|I^+|}\sum_{i\in I^+} z_i + \frac{1-\beta}{|I^-|}\sum_{i\in I^-} z_i \tag{4.33}$$

$$\text{s.t} \quad \sum_{j\in J}(a_{ij} - \frac{1}{2})y_j + Mz_i \geq 0 \quad \forall i \in I \tag{4.34}$$

$$\sum_{j\in J} y_j \geq 1 \tag{4.35}$$

$$z_i,\ y_j \in \{0,1\}, \tag{4.36}$$

where $0 < \beta < 1$ is an important weight of false positive ratio and $M = |J|/2$. The objective in Equation (4.33) is to maximize the sum of weighted classification errors of both false positive and false negative. The constraint set in Equation (4.34) ensures that each training samples receives enough votes with a penalty cost M, where M is a large positive number. The logical constraint in Equation (4.35) ensures that at least one feature is used in the k-nearest neighbor rule.

4.3.3 Experimental Results

The classification performance of the proposed approach is tested for four medical data sets (Wisconsin breast cancer, heart disease-Cleveland, Bupa liver disorder, and Pima Indian's diabetes). Table 4.9 summarizes the characteristics of the data sets when applying a KDNN rule. In the experiment, the number of baseline data set needs to be pre-determined. We define k as the percentage of the baseline data set to the whole data set. We perform 10 times 5-fold cross validation for each instance. Tables 4.10-4.13 present the computational results by varying the value of k for each data set. The accuracy is calculated by the average of *sensitivity* and *specificity* and leaving aside unclassified samples. We observed that the classification accuracy does not vary with the parameter k. The VAMM outperforms the EAMM. In Table 4.14, we compare the performance of the VAMM and the V-FSM. It is seen that we obtain a little higher classification accuracy by using relatively small number of features on all data sets.

Table 4.9: Characteristics of data sets.

Data	Samples Total	+	−	Class (+, −)	Features
Breast cancer-Wisconsin	699	458	241	(malignant, benign)	9
Heart disease-Cleveland	303	139	164	(sick, normal)	13
Bupa liver disorders	325	200	125	(positive, negative)	6
Pima Indians diabetes	768	268	500	(positive, negative)	8

Table 4.10: Results of the VAMM and VEMM for the Wisconsin breast cancer data set. Accuracies of training and testing data sets, numbers of selected features, and computational times (in second) are reported for each approach.

	VAMM				VEMM			
	Accuracy		# of		Accuracy		# of	
k	Training	Testing	features	Time	Training	Testing	features	Time
0.1	0.93	0.92	3.6	120.29	0.96	0.86	4.92	101.00
0.2	0.95	0.94	4.52	123.96	0.97	0.93	5.6	106.93
0.3	0.96	0.95	4.52	124.58	0.97	0.93	5.8	107.11
0.4	0.95	0.94	4.4	125.37	0.97	0.93	5.64	108.13
0.5	0.94	0.93	4.16	127.45	0.97	0.92	6.6	113.05
0.6	0.94	0.94	4.68	132.42	0.96	0.91	7.68	114.76
0.7	0.94	0.94	4.88	128.65	0.96	0.91	7.68	113.86
0.8	0.93	0.93	4.56	129.36	0.96	0.89	6.28	111.86
0.9	0.92	0.92	5.08	128.52	0.96	0.88	6.04	109.86
1.0	0.92	0.91	4.12	131.03	0.94	0.86	5.24	109.65

Table 4.11: Results of the VAMM and VEMM for the Cleveland heart disease data set. Accuracies of training and testing data sets, numbers of selected features, and computational times (in second) are reported for each approach.

	VAMM				VEMM			
	Accuracy		# of		Accuracy		# of	
k	Training	Testing	features	Time	Training	Testing	features	Time
0.1	0.69	0.69	1.00	74.86	0.86	0.37	2.00	45.24
0.2	0.72	0.70	2.28	135.83	0.87	0.41	2.00	56.41
0.3	0.80	0.76	2.80	87.30	0.90	0.55	2.76	56.82
0.4	0.85	0.83	3.36	67.11	0.92	0.64	3.96	56.89
0.5	0.85	0.83	3.88	74.99	0.91	0.64	4.56	61.01
0.6	0.85	0.83	3.84	71.94	0.92	0.63	4.32	63.36
0.7	0.85	0.82	4.96	82.83	0.91	0.56	3.68	68.26
0.8	0.85	0.82	4.96	82.67	0.91	0.52	3.52	63.25
0.9	0.85	0.82	4.40	81.07	0.91	0.52	3.80	64.73
1.0	0.82	0.78	5.56	99.20	0.92	0.50	3.12	62.67

Table 4.12: Results of the VAMM and VEMM for the Pima Indians diabetes data set. Accuracies of training and testing data sets, numbers of selected features, and computational times (in seconds) are reported for each approach.

	VAMM				VEMM			
	Accuracy		# of		Accuracy		# of	
k	Training	Testing	features	Time	Training	Testing	features	Time
0.1	0.73	0.72	1.00	118.29	0.86	0.51	2.00	97.87
0.2	0.74	0.73	1.00	123.91	0.86	0.54	2.00	102.02
0.3	0.74	0.74	1.00	121.01	0.86	0.54	2.00	101.67
0.4	0.74	0.74	1.00	122.45	0.86	0.55	2.00	101.86
0.5	0.74	0.74	1.00	123.61	0.85	0.55	2.00	100.76
0.6	0.74	0.74	1.04	120.29	0.85	0.55	2.00	106.31
0.7	0.74	0.74	1.00	119.76	0.85	0.55	2.00	102.92
0.8	0.74	0.74	1.00	119.76	0.85	0.56	2.00	105.88
0.9	0.74	0.74	1.00	120.21	0.83	0.57	2.00	104.59
1.0	0.74	0.74	1.00	118.50	0.81	0.59	2.00	103.63

Table 4.13: Results of the VAMM and VEMM for the bupa liver disorders data set. Accuracies of training and testing data sets, the numbers of selected features, and computational times (in second) are reported for each approach.

	VAMM				VEMM			
	Accuracy		# of		Accuracy		# of	
k	Training	Testing	features	Time	Training	Testing	features	Time
0.1	0.61	0.59	1.44	45.72	0.80	0.35	2.00	36.49
0.2	0.62	0.59	1.44	50.37	0.79	0.37	2.00	38.20
0.3	0.62	0.58	1.40	43.41	0.78	0.35	2.00	39.92
0.4	0.60	0.56	1.52	46.33	0.79	0.33	2.00	36.93
0.5	0.60	0.55	1.76	44.23	0.79	0.33	2.00	35.70
0.6	0.60	0.55	2.08	44.13	0.78	0.36	1.98	36.65
0.7	0.60	0.56	2.04	46.29	0.74	0.39	1.98	36.15
0.8	0.59	0.56	2.04	43.14	0.72	0.41	1.84	38.65
0.9	0.58	0.57	1.52	40.80	0.62	0.51	1.48	38.81
1.0	0.58	0.58	1.20	40.88	0.58	0.57	1.10	38.69

Table 4.14: Comparison results of VAMM and V-FSM from [55]. The accuracy is reported on the testing data sets.

	VAMM[a]			V-FSM	
Data set	Accuracy	K	No. of features	Accuracy	No. of features
Breast cancer-Wisconsin	0.95	0.3	4.5	0.94	11.6
Heart disease-Cleveland	0.83	0.4	0.59	0.82	7.4
Pima liver disorders	0.74	0.1	1.0	0.72	4.3
Bupa Indian's diabetes	0.59	0.1	1.4	0.58	3.3

[a] The best results is reported among all instances for each data set.

4.4 Conclusion

In feature selection, selecting critical and informative features is very important for classification problems with massive data sets in practice. In the first part of this study, we proposed a new concept of combinatorial optimization, SCOM-MI, to find a best combination of features based on the statistical information. To reduce computational complexity, we proposed an incremental optimization search algorithm to solve the SCOM-MI using different proposed selection criteria. We demonstrated the classification performance using support vector machine, k nearest neighbor rule, and linear discriminant analysis for various data sets. The exhaustive experimental results showed that the heuristic search with the selection criterion of "minimum-redundancy and maximum-relevancy" gives amongst the best performance and the discretization of the original data sets yields better performance for the data sets of Cleveland and statlog heart disease. In addition, we obtained the competitive results for very large data sets of gene expressions when compared to the existing mRMR method. We finally showed that the proposed framework can be generalized by employing any different

useful information as inputs in the SCOM-MI model. In the second part of this study, we have proposed a new patten-based optimization approach (dFSM) for direct classification by selected features based on the decomposed k-nearest neighbor rule. We presented better performance compared to the original FSM.

According to the preliminary results by testing a number of real data sets, we saw the potential of the proposed feature selection approaches in classification. In future research, there are studies in several directions to be continued. (1) Comprehensive experiments with comparisons with other existing approached needs to be further implemented. (2) Since the statistical information is shown to be effective to select good features, it may be combined in the dFSM to directly solve the classification problem. (3) The results show that it is faster to solve the problem with binary input features than numerical input features. Thus, the SCOM-MI can be employed for the feature selection in the LAD framework. In addition, the feature selection is not limited to classification problems while clustering problems are commonly faced with the similar challenge by massive data.

Chapter 5

CONCLUSION

In this dissertation, we presented new combinatorial optimization modeling and computational algorithms for large-scale clustering and classification problems with highly computational complexity increased by the size of dimensionality of massive and complex data.

In the first part (clustering), we have studied a very important problem in computational and population biology, a sibling reconstruction problem. It can be mathematically formulated as a spacial case of capacitated clustering problem. We proposed mathematical optimization models based on the concepts of combinatorics and similarity likelihood. We proposed exact and heuristic solution approaches that were able to solve the problems comparably and significantly outperform other existing approaches on the same real biological data sets. In the second part (classification), we focused on the development of effective approaches for improving the classification performance of LAD method. We proposed a new mathematical optimization model for generating decisive patterns. Moreover, we proposed a column generation framework, where the proposed pattern generation approaches, to improve the classification accuracy and computation efficiency. We demonstrated the effectiveness and practicability in medical applications and better performance compared to other existing approaches. We showed that the column generation technique in optimization is favorable for above two types of problems in data mining with huge and complex data.

In the third part (feature selection), we proposed new optimization-based feature selection methods. The first approach is an optimization model incorporated with statistical information to select a compact subsets of informative features that can be used for any classifiers. The second approach is a pattern-based optimization approach

using a decomposed nearest neighbor rule, which can be directly used as a classifier. Along the line, there are many potential studies to be continued for the future research.

References

[1] Gabriela Alexe, Sorin Alexe, David E. Axelrod, Tibérius O. Bonates, Irina Lozina, Michael Reiss, and Peter L. Hammer. Breast cancer prognosis by combinatorial analysis of gene expression data. *Breast Cancer Research*, 8(41), 2006.

[2] Gabriela Alexe, Sorin Alexe, David E. Axelrod, Peter L. Hammer, and D. Weissmann. Logical analysis of diffuse large b-cell lymphomas. *Artificial Intelligence in Medicine*, 34(3):235–267, 2005.

[3] Gabriela Alexe, Sorin Alexe, and Peter L. Hammer. Pattern-based clustering and attribute analysis. *Soft Computing*, 10(5):442–452, 2006.

[4] Gabriela Alexe, Sorin Alexe, Peter L. Hammer, and Alexander Kogen. Comprehensive vs. comprehensible classifiers in logical analysis of data. *Discrete Applied Mathematics*, 156(6):870–882, 2008.

[5] Gabriela Alexe, Sorin Alexe, Peter L. Hammer, and Bela Vizvari. Pattern-based feature selection in genomics and proteomics. *Annals of Operations Research*, 148(1):189–201, 2006.

[6] Gabriela Alexe, Sorin Alexe, Lance A. Liotta, Emanuel Petricoin, Michael Reiss, and Peter L. Hammer. Ovarian cancer detection by logical analysis of proteomic data. *Proteomics*, 4(3):766–783, 2004.

[7] Gabriela Alexe and Peter L. Hammer. Spanned patterns for the logical analysis of data. *Discrete Applied Mathematics*, 154(7):203–225, 2006.

[8] Gabriela Alexe and Peter L. Hammer. Pattern-based discriminants in the logical analysis of data. *Data Mining in Biomedicine*, 7:3–23, 2007.

[9] Sorin Alexe, Eugene Blackstone, Peter L. Hammer, Hemant Ishwaran, Michael S. Lauer, and Claire E. Pothier Snader. Coronary risk prediction by logical analysis of data. *Annals of Operations Research*, 19(1-4):15–42, 2003.

[10] Sorin Alexe and Peter L. Hammer. Accelerated algorithm for pattern detection in logical analysis of data. *Discrete Applied Mathematics*, 154(7):1050–1063, 2006.

[11] Anthony Almudevar. A simulated annealing algorithm for maximum likelihood pedigree reconstruction. *Theoretical Population Biology*, 63:63–75, 2003.

[12] Anthony Almudevar. A graphical approach to relatedness inference. *Theoretical Population Biology*, 71(2):213–229, 2007.

[13] Anthony Almudevar and C. Field. Estimation of single generation sibling relationships based on DNA markers. *Journal of Agricultural, Biological, and Environmental Statistics*, 4:136–165, 1999.

[14] U. Alon, Naama Barkai, Daniel A. Notterman, Kenneth W. Gish, S. Ybarra, Daniel Mack, and Arthur J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, 96:6745–6750, 1999.

[15] Mary V. Ashley, Tanya Y. Berger-Wolf, Piotr Berman, Wanpracha Chaovalitwongse, Bhaskar DasGupta, and Ming-Yang Kao. On approximating four covering and packing problems. *Journal of Computer and System Sciences*, 75:287–302, 2009.

[16] Arthur Asuncion and David J. Newman. UC irvine machine learning repository, 2007.

[17] Pasquale Avella, Maurizio Boccia, Antonio Sforza, and Igor Vasil'ev. An effective heuristic for large-scale capacitated facility location problems. *Journal of Heuristics*, 15:597–615, 2009.

[18] Roberto Baldacci, Eleni Hadjiconstantinou, Vittorio Maniezzo, and Aristide Mingozzi. A new method for solving capacitated location problems based on a set partitioning approach. *Computers & Operations Research*, 29:365–386, 2002.

[19] Cynthia Barnhart, Ellis L. Johnson, and George L. Nemhauser. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[20] Poberto Battiti. Using mutual information for selecting features in supervised neural net learning. *Neural Networks, IEEE Transactions on*, 5(4):537 –550, jul 1994.

[21] Tanya Y. Berger-Wolf, Bhaskar DasGupta, Wanpracha Chaovalitwongse, and Mary V. Ashley. Combinatorial reconstruction of sibling relationships. In *Proceedings of the 6th International Symposium on Computational Biology and Genome Informatics (CBGI 05)*, pages 1252–1255, 2005.

[22] Tanya Y. Berger-Wolf, Sadd Sheikh, Bhaskar DasGupta, Mary V. Ashley, Isabel C. Caballero, Wanpracha Chaovalitwongse, and S.L. Putrevu. Reconstructing sibling relationships in wild populations. *Bioinformatics*, 23:49–56, 2007.

[23] Paola Bertolazzi, Giovanni Felici, Paola Festa, and Giuseppe Lancia. Logic classification and feature selection for biomedical data. *Computers and Mathematics with Applications*, 55(5):889–899, 2008.

[24] Jennifer Beyer and Bernie May. A graph-theoretic approach to the partition of individuals into full-sib families. *Molecular Ecology*, 12:2243–2250, 2003.

[25] Avrim L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[26] Tibérius O. Bonates. *Optimization in Logical Analysis of Data*. PhD dissertation, Rutgers University, RUTCOR, 2007.

[27] Tibérius O. Bonates, Peter L. Hammer, and Alexander Kogen. Maximum patterns in datasets. *Discrete Applied Mathematics*, 156(6):846–861, 2008.

[28] Endre Boros, Peter L. Hammer, Toshihide Ibaraki, and Alexander Kogan. Logical analysis of numerical data. *Mathematical Programming*, 79(1-3):163–190, 1997.

[29] Endre Boros, Peter L. Hammer, Toshihide Ibaraki, Alexander Kogan, Eddy Mayoraz, and Ilya Muchnik. An implementation of logical analysis of data. *IEEE transactions on knowledge and data engineering*, 12(2):292–306, 2000.

[30] Peter J. Bowler. *The Mendelian Revolution: The Emergence of Hereditarian Concepts in Modern Science and Society*. The Johns Hopkins University Press, 1989.

[31] Paul S. Bradley, Usama M. Fayyad, and Olvi L. Mangasarian. Mathematical programming for data mining: Formulations and challenges. *INFORMS Journal on Computing*, 11:217–238, 1999.

[32] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[33] K. Butler, C. Field, C.M. Herbinger, and B.R. Smith. Accuracy, efficiency and robustness of four algorithms allowing full sibship reconstruction from dna marker data. *Molecular Ecology*, 13:1589–1600, 2004.

[34] Alberto Ceselli, Federico Liberatore, and Giovanni Righini. A computational evaluation of a general branch-and-price framework for capacitated network location problems. *Annals of Operations Research*, 167:209–251, 2009.

[35] 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.

[36] Wanpracha Chaovalitwongse, Tanya Y. Berger-Wolf, Bhaskar DasGupta, and Mary V. Ashley. A robust combinatorial approach for sibling relationships reconstruction. *Optimization Methods and Software*, 22(1):11–24, 2007.

[37] Wanpracha Chaovalitwongse, Chun-An Chou, Tanya Y. Berger-Wolf, Bhaskar DasGupta, Saad Sheikh, S. Lahari Putrevu, Mary V. Ashley, and Isabel C. Caballero. New optimization model and algorithm for sibling reconstruction from genetic markers. *INFORMS Journal on Computing*, 22(2):180–194, 2010.

[38] Wanpracha Chaovalitwongse, Panos M. Pardalos, and Oleg A. Prokopyev. A new linearization technique for multi-quadratic 0-1 programming problems. *Operations Research Letters*, 32:517–522, 2004.

[39] Wanpracha Art Chaovalitwongse, Ya-Ju Fan, and Rajesh C. Sachdeo. Novel optimization models for abnormal brain activity classification. *Operations Research*, 56(6):1450–1460, 2008.

[40] Irène Charon and Olivier Hudry. The noise method: a new method for combinatorial optimization. *Operations Research Letters*, 14:133–137, 1993.

[41] Antonio Augusto Chaves and Luiz A.N. Lorena. Clustering search algorithm for the capacitated centred clustering problem. *Computers & Operations Research*, 37:552–558, 2010.

[42] Chun-An Chou, Wanpracha Art Chaovalitwongse, Tanya Y. Berger-Wolf, Bhaskar DasGupta, and Mary V. Ashley. Capacitated clustering problem in computational biology: Combinatorial and statistical approach for sibling reconstruction. *Computer & Operations Research*, 39:609–619, 2012.

[43] Chun-An Chou, Wanpracha Art Chaovalitwongse, Tibérius Oliveira Bonates, and Chungmok Lee. Improved pattern generation approaches in logical analysis of medical data. *Submiited to INFORMS Journal of Computing*, 2011.

[44] Chun-An Chou, Zhe Liang, Wanpracha Art Chaovalitwongse, Tanya Y. Berger-Wolf, Bhaskar DasGupta, Saad Sheikh, S. Lahari Putrevu, Mary V. Ashley, and Isabel C. Caballero. Column generation framework of nonlinear similarity model for reconcstuction sibling relationships. *Submiited to INFORMS Journal of Computing*, 2011.

[45] Jeffrey K. Conner. Personal communication, 2006.

[46] Thomas Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

[47] Yves Crama, Peter L. Hammer, and Toshihide Ibaraki. Cause-effect relationships and partially defined boolean functions. *Annals of Operations Research*, 16(1):299–325, 1988.

[48] Yue Cui, Jesse Jin, Shiliang Zhang, Suhuai Luo, and Qi Tian. Correlation-based feature selection and regression. In Guoping Qiu, Kin Lam, Hitoshi Kiya, Xiang-Yang Xue, C.-C. Kuo, and Michael Lew, editors, *Advances in Multimedia Information Processing - PCM 2010*, volume 6297 of *Lecture Notes in Computer Science*, pages 25–35. Springer Berlin / Heidelberg, 2010.

[49] Sanmay Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *International Conference on Machine Learning*, pages 74–81, 2001.

[50] Yumin Deng and Jonathan Bard. A reactive grasp with path relinking for capacitated clustering. *Journal of Heuristics*, 17(2):119–152, 2011.

[51] Chris Ding and Hanchuan Peng. Mimimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(2):185–205, 2005.

[52] Jonathan Eckstein, Peter L. Hammer, Ying Liu, Mikhail Nediak, and Bruno Simeone. The maximum box problem and its application to data analysis. *Computational Optimization and Applications*, 23(3):285–298, 2002.

[53] Eleazar Eskin, Eran Halperin, and Richard M. Karp. Efficient reconstruction of haplotype structure via perfect phylogeny. *Journal of Bioinformatics and Computational Biology*, 1(1):1–20, 2003.

[54] Pablo A Estévez, Michel Tesmer, Claudio A Perez, and Jacek M Zurada. Normalized mutual information feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):189–201, 2009.

[55] Ya-Ju Fan and Wanpracha Art Chaovalitwongse. Optimizing feature selection to improve medical diagnosis. *Annals of Operations Research*, 174(1):169–183, 2010.

[56] Thomas A. Feo and Mauricio G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[57] François Fleuret and Isabelle Guyon. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[58] Paulo M. Franca, Nelida M. Sosa, and Vitoria Pureza. Adaptive tabu search approach for solving the capacitated clustering problem. *International Transactions of Operations Research*, 6:665–678, 1999.

[59] Fred W. Glover. Tabu search - part I. *ORSA, Journal on Computing*, 1:190–206, 1989.

[60] Fred W. Glover. Improved linear programming models for discriminant analysis. *Decision Sciences*, 21(4):771–785, 1990.

[61] Fred W. Glover. Tabu search - part II. *ORSA, Journal on Computing*, 2:4–32, 1990.

[62] Todd R. Golub, Donna K. Slonim, Pablo Tamayo, C. Huard, M. Gaasenbeek, Jill P. Mesirov, H. Coller, Mignon L. Loh, James R. Downing, M. A. Caligiuri, C. D. Bloomfield, and Eric S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[63] Dan Gusfield. Partition-distance: A problem and class of perfect graphs arising in clustering. *Information Processing Letters*, 82(3):159–164, 2002.

[64] Lsahelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of MAchine Learning Research*, 3:1157–1182, 2003.

[65] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update; sigkdd explorations. *SIGKDD Explorations*, 11(1):11–18, 2009.

[66] Mark A. Hall. *Correlation-based feature selection for machine learning*. PhD dissertation, University of Waikato, Department of Computer Science, 1999.

[67] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 359–366, 2000.

[68] Peter L. Hammer. *The logic of Cause-effect relationships*. Lecture at the International Conference on Multi-Attribute Decision Making via Operations Research-based Expert systems, Passau, Germany, 1986.

[69] Peter L. Hammer and Tibérius O. Bonates. Logical analysis of data - an overview: From combinatorial optimization to medical applications. *Annals of Operations Research*, 148(1):1039–1049, 2006.

[70] Peter L. Hammer, Alexander Kogan, and Miguel A. Lejeune. Reverse-engineering country risk ratings: a combinatorial non-recursive model. *Annals of Operations Research*, 188(1):185–213, 2010.

[71] Peter L. Hammer, Alexander Kogan, Bruno Simeone, and Sándor Szedmák. Pareto-optimal patterns in logical analysis of data. *Discrete Applied Mathematics*, 144(1):79–102, 2004.

[72] Rob L. Hammond, Andrew F. G. Bourke, and M. W. Broford. Mating frequency and mating system of the polygynous ant, *Leptothorax acervorum*. *Molecular Ecology*, 10:2719–2728, 2001.

[73] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems, 2006.

[74] Pierre Hansen and Brigitte Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79:191–215, 1997.

[75] Simon Haykin. *Applied logistic regression*. Prentice Hall, 1998.

[76] Christophe M. Herbingera, Patrick T. O'Reilly, Roger W. Doylea, Jonathan M. Wrighta, and Fiona O'Flynn. Early growth performance of atlantic salmon full-sib families reared in single family tanks or in mixed family tanks. *Aquaculture*, 173:105–116, 1999.

[77] David W. Hosmer and Stanley Lemeshow. *Applied logistic regression*. John Wiley & Sons, 1989.

[78] Saad Hseikh, Wanpracha Chaovalitwongse, Tanya Y. Berger-Wolf, Mary V. Ashley, Isabel C. Caballero, Wanpracha Chaovalitwongse, and Bhaskar DasGupta. Error-tolerant sibship reconstruction in wild populations. In *Proceedings of 7th Annual International Conference on Computational Systems Bioinformatics*, 2008.

[79] Jianping Huaa, Waibhav D. Tembeb, and Edward R. Dougherty. Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recognition*, 42(3):409–424, 2009.

[80] Dean R. Jerrya, Brad S. Evansa, Matt Kenwayb, and Kate Wilson. Development of a microsatellite DNA parentage marker suite for black tiger shrimp penaeus monodon. *Aquaculture*, 255:542–547, 2006.

[81] Don H. Johnson and Sinan Sinanovic. Symmetrizing the kullback-leibler distance. Technical report, IEEE Transactions on Information Theory, 2000.

[82] Kristina L. Kickler, Mark T. Holder, Scott K. Davis, Rene Márquez-M, and David W. Owens. Detection of multiple paternity in the kemp's ridley sea turtle with limited sampling. *Molecular Ecology*, 8(5):819–830, 1999.

[83] Kwangsoo Kim and Hong Seo Ryoo. A lad-based method for selecting short oligo probes for genotyping applications. *OR Spectrum*, 30:249–268, 2008.

[84] Alexander Kogan and Miguel A. Lejeune. Combinatorial methods for constructing credit risk ratings. *Handbook of Quantitative Finance and Risk Management*, pages 639–664, 2010.

[85] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[86] Dmitry A. Konovalov, Clint Manning, and Michael .T. Henshaw. KINGROUP: A program for pedigree relationship reconstruction and kin group assignments using genetic markers. *Molecular Ecology Notes*, 4(4):779–782, 2004.

[87] Yiannis A. Koskosidis and Warren B. Powell. Clustering algorithms for consolidation of customer orders into vehicle shipments. *Transportation Research Part B*, 26:365–379, 1992.

[88] Sotiris Kotsiantis and Dimitris Kanellopoulos. Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32:47–58, 2006.

[89] Solomon Kullback. *Information Theory and Statistics*. Courier Dover, 1997.

[90] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[91] Lukasz Kurgan, Krzysztof Cios, Ryszard Tadeusiewicz, Marek Ogiela, and Lucy Goodenday. Knowledge discovery approach to automated cardiac spect diagnosis. *Artificial Intelligence in Medicine*, 23(2):149–69, 2001.

[92] Nojun Kwak and Chong-Ho Choi. Improved mutual information feature selector for neural networks in supervised learning. In *Neural Networks, 1999. IJCNN '99. International Joint Conference on*, volume 2, pages 1313–1318 vol.2, jul 1999.

[93] Giuseppe Lancia and Paolo Serafini. A set-covering approach with column generation for parsimony haplotyping. *INFORMS Journal on Computing*, 21(1):151–166, 2009.

[94] Michael S. Lauer, Sorin Alexe, Claire E. Pothier Snader, Eugene H. Blackstone, Hemant Ishwaran, and Peter L. Hammer. Use of the logical analysis of data method for assessing long-term mortality risk after exercise electrocardiography. *Circulation*, 106:685–590, 2002.

[95] Jing Li and Tao Jiang. Efficient inference of haplotypes from genotype on a pedigree. *Journal of Bioinformatics and Computational Biology*, 1(1):41–69, 2003.

[96] Max A. Little, Patrick E. McSharry, Eric J. Hunter, Jennifer Spielman, and Lorraine O. Ramig. Suitability of dysphonia measurements for telemonitoring of parkinsons disease. *IEEE Transactions on Biomedical Engineering*, 56(4):1015–1022, 2009.

[97] Huan Liu, Farhad Hussain, Chew Lim Tan, and Manoranjan Dash. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, pages 393–423, 2004.

[98] Huawen Liu, Lei Liu, and Huijie Zhang. Feature Selection Using Mutual Information: An Experimental Study. In Tu-Bao Ho and Zhi-Hua Zhou, editors, *PRICAI 2008: Trends in Artificial Intelligence*, volume 5351 of *Lecture Notes in Computer Science*, chapter 24, pages 235–246. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[99] Luiz Lorena Lorena, Edson L. F. Senne, and Estadual Paulista. A column generation approach to capacitated p-median problems. *Computers & Operations Research*, 31:863–876, 2004.

[100] Macro E. Lubbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[101] Olvi L. Mangasarian. Mathematical programming in data mining. *Data Min. Knowl. Discov.*, 1:183–201, January 1997.

[102] Olvi L. Mangasarian, W. Nick Street, and William H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, 1995.

[103] Olvi L. Mangasarian and William H. Wolberg. Cancer diagnosis via linear programming. *SIAM News*, 23(5):1–18, 1990.

[104] Vittorio Maniezzo, Aristide Mingozzi, and Roberto Baldacci. A bionomic approach to the capacitated p-median problem. *Journal of Heuristics*, 4:263–280, 1998.

[105] Eddy Mayoraz and Miguel Moreira. Combinatorial approach for data binarization. In *Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '99, pages 442–447, London, UK, 1999. Springer-Verlag.

[106] Anuj Mehrotra and Michael A. Trick. Cliques and clustering: A combinatorial approach. *Operations Research Letters*, 22:1–12, 1998.

[107] Gregor Mendel. Experiments on plant hybridization (versuche ber pflanzenhybriden). *Journal of the Royal Horticultural Society*, 26:1–32, 1901.

[108] Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[109] John M. Mulvey and M.P. Beck. Solving capacitated clustering problems. *European Journal of Operations Research*, 18:339–348, 1984.

[110] Marcos Negreiros and Augusto Palhano. The capacitated centred clustering problem. *Computers & Operations Research*, 33:1639–1663, 2006.

[111] Ibrahim H. Osman and N. Creistofides. Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions of Operations Research*, 1:317–336, 2002.

[112] Ibrahim H. Osman and Ahmadi Samad. Guided construction search for the capacitated p-median problem. *Working Paper, School of Business, American Univeristy of Beirut, Lebanon*, 2002.

[113] Ian Painter. Sibship reconstruction without parental information. *Journal of Agricultural, Biological, and Environmental Statistics*, 2:212–229, 1997.

[114] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[115] Yonghong Peng, Zhiqing Wua, and Jianmin Jianga. A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics*, 43(1):15–23, 2010.

[116] David C. Queller, Joan E. Strassmann, and Colin R. Hughes. Microsatellites and kinship. *Trends in Ecology and Evolution*, 8:285–288, 1993.

[117] John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[118] Anupama Reddy, Honghui Wang, Hua Yu, Tiberius O Bonates, Vimla Gulabani1, Joseph Azok, Gerard Hoehn, Peter L Hammer1, Alison E Baird, and King C Li. Logical analysis of data (lad) model for the early diagnosis of acute ischemic stroke. *BMC Medical Informatics and Decision Making*, 8(30), 2008.

[119] Irene Rodriguez-Lujan, Ramon Huerta, Charles Elkan, and Carlos Santa Cruz. Quadratic programming feature selection. *Journal of Machine Learning Research*, 11:1491–1516, 2010.

[120] David M. Ryan and B.A. Foster. An integer programming approach to scheduling. In A. Wren, editor, *Computer Schedule of Public Transport Urban Passenger Vehicle and Crew Scheduling*, pages 269–280. 1981.

[121] Hong Seo Ryoo and In-Yong Jang. Milp approach to pattern generation in logical analysis of data. *Discrete Applied Mathematics*, 157(4):749–761, 2009.

[122] Yvan Saeys, Iñki Inza, and Pedro Larra naga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[123] Ahmadi Samad and Ibrahim H. Osman. Density based problem space search for the capacitated clustering problems. *Annals of Operations Research*, 131:21–43, 2002.

[124] Ahmadi Samad and Ibrahim H. Osman. Greedy random adaptive memory programming search for the capacitated clustering problems. *European Journal of Operations Research*, 162:30–44, 2005.

[125] Bernhard Schökopf and Alexander J. Smola. *Learning with Kernels*. MIT Press Cambridge, Massachusetts, USA, 2002.

[126] Saad Sheikh, Tanya Y. Berger-Wolf, Ashfaq A. Khokar, Chun-An Chou, Wanpracha Chaovalitwongse, Mary V. Ashley, Isabel C. Caballero, and Bhaskar DasGupta. Combinatorial reconstruction of half-sibling groups: Models and algorithms. *Journal of Bioinformatics and Computational Biology*, 8(2):1–20, 2010.

[127] Bruce R. Smith, Christophe M. Herbinger, and Heather R. Merry. Accurate partition of individuals into full-sib families from genetic data without parental information. *Genetics*, 158:1329–1338, 2001.

[128] Dacheng Tao, Xuelong Li, Xindong Wu, and S.J. Maybank. General averaged divergence analysis. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 302 –311, oct. 2007.

[129] Michel Tesmer and Pablo A. Estévez. Amifs: adaptive feature selection by using mutual information. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 1, pages 1415–1420 vol.4, july 2004.

[130] Stuart C. Thomas and William G. Hill. Sibship reconstruction in hierarchical population structures using markov chain Monte Carlo techniques. *Genetic Research*, 79:227–234, 2002.

[131] Joy A. Thomas Thomas M. Cover. *Elements of Information Theory*. New York: Wiley, 2006.

[132] François Vanderbeck. *Decomposition and Column Generation for Integer Programs*. Ph.D Thesis, Universite Catholique de Louvain, Belgium, 1994.

[133] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[134] Michel Verleysen, Fabrice Rossi, and Damien François. Advances in feature selection with mutual information. In Michael Biehl, Barbara Hammer, Michel Verleysen, and Thomas Villmann, editors, *Similarity-Based Clustering*, volume 5400 of *Lecture Notes in Computer Science*, pages 52–69. Springer Berlin / Heidelberg, 2009.

[135] Jinliang Wang. Sibship reconstruction from genetic data with typing errors. *Genetics*, 166(4):1968–1979, 2004.

[136] Jinliang Wang and Anna W. Santure. Parentage and sibship inference from multilocus genotype data under polygamy. *Genetics*, 181(4):1579–1594, 2009.

[137] Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, and Vladimir N. Vapnik. Feature selection for svms. In *Advances in Neural Information Processing Systems 13*, pages 668–674. MIT Press, 2001.

[138] Welbert E. Wilhelm. A technical review of column generation in integer programming. *Optimization and Engineering*, 2:159–200, 2002.

[139] Alex Wilson, Paul Sunnucks, and J. Barker. Isolation and characterization of 20 polymorphic microsatellite loci for scaptodrosophila hibisci. *Molecular Ecology Notes*, 2:242–244, 2002.

[140] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, second ed.* Morgan Kaufmann, San Francisco, 2005.

[141] Lei Yu. Redundancy based feature selection for microarray data. In *Proceedings of SIGKDD*, pages 737–742. ACM Press, 2004.

[142] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 856–863, 2003.

[143] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, 2004.

Vita

Chun-An (Joe) Chou

EDUCATION

09/2007 - 10/2011 **Ph.D.** in Industrial and Systems Engineering
Rutgers University, New Jersey, USA

09/2005 - 02/2007 **M.S.** in Operations Research
Columbia University, New York, USA

09/2000 - 05/2002 **M.S.** in Bioenvironmental Systems Engineering
National Taiwan University, Taipei, Taiwan

09/1996 - 06/2000 **B.S.** in Forestry and Resource Conservation
National Taiwan University, Taipei, Taiwan

OCCUPATION

09/2010 - 05/2011 **Instructor and Teaching Assistant**
Department of Industrial and Systems Engineering
Rutgers University, New Jersey, USA

02/2007 - 05/2010 **Research Assistant**
Department of Industrial and Systems Engineering
Rutgers University, New Jersey, USA

07/2004 - 07/2005 **Research Fellow**
Department of Bioenvironmental Systems Engineering
National Taiwan University, Taipei, Taiwan

09/2001 - 06/2002 **Research Assistant**
Department of Bioenvironmental Systems Engineering
National Taiwan University, Taipei, Taiwan

09/2000 - 06/2001 **Teaching Assistant**
Department of Bioenvironmental Systems Engineering
National Taiwan University, Taipei, Taiwan

PUBLICATIONS

Chun-An Chou, W. Art Chaovalitwongse Tanya Y. Berger-Wolf, Bhaskar DasGupta, and Mary V. Ashley. Column Generation Framework of Nonlinear Similarity Model for

Reconstructing Sibling Groups, revision submitted to INFORMS *Journal on Computing*, 2011

Chun-An Chou, W. Art Chaovalitwongse, Tibérius O. Bonates, and Chungmok Lee. Improved Pattern Generation Approach in Logical Analysis of Medical Data, submitted to INFORMS *Journal on Computing*, 2011

Chun-An Chou, W. Art Chaovalitwongse, Tanya Y. Berger-Wolf, Bhaskar DasGupta, and Mary V. Ashley. Capacitated Clustering Problem in Computational Biology: Combinatorial and Statistical Approach for Sibling Reconstruction, *Computer & Operations Research*, Vol. 39, 609-619, 2012

W. Art Chaovalitwongse, **Chun-An Chou**, Tanya Y. Berger-Wolf, Bhaskar DasGupta, Saad Sheikh, Mary V. Ashley, and Isabel C. Caballero. New Optimization Model and Algorithm for Sibling Reconstruction from Genetic Markers. INFORMS *Journal on Computing*, Vol. 22(2), 188-194, 2010

Saad I. Sheikh, Tanya Y. Berger-Wolf, Ashfaq Khokar, Isabel C. Caballero, Mary V. Ashley, W. Art Chaovalitwongse, **Chun-An Chou**, Bhaskar DasGupta. Combinatorial Reconstruction of Half-Sibling Groups: Models and Algorithms. *Journal of Bioinformatics and Computational Biology*, Vol. 8(1), 1-20, 2010

Ching-Pin Tung, **Chun-An Chou**. Pattern Classification Using Tabu Search to Identify the Spatial Distribution of Groundwater Pumping. *Hydrogeology Journal*, Vol. 12(5), 488-496, 2004

Ching-Pin Tung, **Chun-An Chou**. Application of Tabu Search to Groundwater Parameter Zonation. *Journal of the American Water Resources Association*, Vol. 38(4), 1115-1126, 2002