Discovering Interacting Features for Prediction of Response submitted to Iowa State University

by

Maryam Nikouei Mehr

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Industrial and Manufacturing Systems Engineering, Minor: Statistics

Program of Study Committee:
Dr. Sigurdur Olafsson, Co-major Professor
Dr. Lizhi Wang , Co-major Professor
Dr. Stephen Gilbert
Dr. Guiping Hu
Dr. Somak Dutta

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation/thesis. The Graduate College will ensure this dissertation/thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa
2020

DEDICATION

Dedicated to my immediate family

Contents

Page

List of Tables

List of Figures

ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this dissertation. First and foremost, Dr. Sigurdur Olafsson for his guidance, patience and support throughout this research and the writing of this dissertation. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my committee members for their efforts and contributions to this work: Dr. Lizhi Wang, Dr. Stephen Gilbert, Dr. Guiping Hu, and Dr. Somak Dutta. At the end, I would like to thank my immediate family for their kind support and encouragement.

ABSTRACT

Machine learning as a discipline has used optimization techniques and algorithms for years. Similarly, machine learning has influenced optimization by driving the development of new optimization approaches that address the major challenges raised by machine learning applications. This exchange continues to deepen, producing a growing literature in both fields by attracting researchers. Theoretical property and wide applicability of optimization approaches made them suitable method to be rooted in machine learning (Sra et al. (2012)).

One of the well-known techniques in machine learning is feature selection. Feature selection or variable selection have become the focus of much research in areas of application for which data sets with tens or hundreds of thousands of variables are available (Guyon and Elisseeff (2003)). As the dimensionality of the data rises, the amount of data required for precise forecast analysis grows exponentially (Saeys et al. (2007)). The goal of variable selection is : (1) improving the prediction performance of the predictors, (2) providing more cost-effective predictors, and (3) providing better understanding of data (John et al. (1994)).

One of the areas of study that deals with high-dimensionality data is Genetic. Studies of genes and genome, genes and environment, and genes and behavior investigate high dimensional data to address several research questions. One of the challenges associated with these studies is the phenomenon which is known as epistasis (Bateson (1909)). Epistasis refers to the incident where the interaction of multiple genes affects a certain phenotype in addition to their individual additive effects (Cordell (2002a); Xu and Jia (2007a)). Similar epistatic effects are also ubiquitous in other application areas, such as gene-environment interactions, where a certain effect is triggered only when a particular combination of genes and environmental components is present. Epistasis detection has been recognized as a major challenge in the field of genetics. Previously proposed methods either focused on finding two-gene interactions using brute force enumeration or resorted to heuristic algorithms to search only a subset of the solution space. Herein, we presented an optimization approach that can identify the number of explanatory variables responsible for the interaction effect as well as the exact combination of these variables. We explored simulation experiments using a soybean data set and concluded that the proposed approach had a 95.5% chance of correctly detecting second-order to fifth-order epistases, which was a significant improvement over two alternative approaches in the literature.

In statistics, epistasis is known as interaction effect. Consideration of interaction effect in forecast analysis leads to better predictions. As the goal of feature selection is choosing a minimal subset based on some criteria such that the original task could be achieved with good precision, consideration of interaction effects in minimal subset is crucial. Feature interaction presents a challenge in feature selection (Cotter et al. (2001); Kira and Rendell (1992)). Considering well-known feature selection methods, apparently redundant or irrelevant features have always been removed from subset selections. In our second paper, we showed removal of apparently redundant and irrelevant features is not always wise. Features could be considered irrelevant based on its correlation with the response (Hall (2000); Yu and Liu (2003)), while in combination with other features, they could be considered as relevant. Also features could be considered redundant while in the presence of other features, they can be strongly predictive of the response. To impact the importance of including interaction effect in subset selection, we followed up with a novel feature selection approach for data sets with continuous response variables and established a computational framework for selecting a subset of features with significant main effects and a subset of features with significant interaction effects. The proposed method finds features with significant main effect using well-known feature selection methods and implements a mixed integer linear programming model introduced in the first paper to recognize feature interactions and their exact combinations. The implementation of the proposed algorithm on both synthetic and real data sets concluded to the improvement in prediction analysis.

In order to predict classification of two finite point sets in n-dimensional features space, we proposed a new feature selection model, which is built on mathematical modeling that uses as few feature as possible. Proposed feature selection model finds a separating plane to classify two subsets $\mathcal{A}$ and $\mathcal{B}$ while minimizing the number of missclassifications. The purpose of our work is to establish a computational framework for selection a subset of features that are independently important for the target prediction and a subset of features that might not be considered important independently but in the presence of other features, they can strongly be predictive of the response. With consideration of feature interaction, the new feature selection model tries to find the separating plane that reduces the number of missclassifications. Computational tests of the method on the publicly available real-world databases shows an improvement on the total number of missclassifications.

Chapter1. An optimization approach to epistasis detection

Paper is published by *European Journal of Operational Research*

Lizhi Wang and Maryam Nikouei Mehr

Abstract

Epistasis refers to the phenomenon where the interaction of multiple genes affects a certain phenotype in addition to their individual additive effects. Similar epistatic effects are also ubiquitous in other application areas, such as gene-environment interactions, where a certain effect is triggered only when a particular combination of genes and environmental components is present. Epistasis detection has been recognized as a major challenge in the field of genetics. Previously proposed methods either focused on finding two-gene interactions using brute force enumeration or resorted to heuristic algorithms to search only a subset of the solution space. Herein we present an optimization approach that can identify the number of explanatory variables responsible for the epistasis as well as the exact combination of these variables. Results from simulation experiments using a soybean data set suggested that the proposed approach had a 95.5% chance of correctly detecting second-order to fifth-order epistases, which was a significant improvement over two alternative approaches in the literature.

1.1 Introduction

The epistatic effect refers to the interaction of genes (Cordell (2002a); Xu and Jia (2007a)). This term was first used by Bateson (Bateson (1909)) in 1909 to describe the phenomenon where an allele at one locus prevents the allele at another locus from manifesting its effect. More recently, geneticists and biologists have discovered higher-order epistasis for complex traits that involve five or more genes. For example, Taylor and Ehrenreich (Taylor and Ehrenreich (2014)) discovered a colony morphology trait in yeast strains that was caused by genetic interactions of five or more loci.

In statistical terms, the epistatic effect can be defined in a multiple linear regression context where each explanatory variable makes an additive contribution to the response variable. The epistatic effect is the additional (positive or negative) effect that is triggered when a certain combination of genes take certain allelic variations simultaneously. For example, the height of a certain plant species may be influenced by three alleles each with two variants (A or a, B or b, and C or c). An epistasis may be defined as the phenomenon that plants with the combination of (A, B, c) are one inch taller than the sum of the individual effects of alleles A, B, and c.

Epistasis holds the key to many scientific discoveries in genetics. Ritchie et al. (Ritchie et al. (2001)) identified three genes whose interactions are responsible for sporadic breast cancer. Combarros et al. (Combarros et al. (2009)) found 36 examples of significant epistasis in sporadic Alzheimer's disease. Witnessing the increasing capability of discovering epistatic genes in sickle-cell anemia, Nagel pointed out that (Nagel (2005)) such techniques have the potential of advancing therapeutic strategies that target epistatic genes in concert with the risks involved in individual patients.

Detecting the specific epistasis that triggers the epistatic effect is a notoriously challenging task. Mackay and Moore (Mackay and Moore (2014)) summarized three challenges for this task. *First*, commonly used parametric statistical methods were not designed to detect interactions and often struggle to provide precise parameter estimation. *Second*, it requires prohibitive computational resources to enumerate all possible combinatorial epistasis candidates. For the example with three alleles and two variants (A or a, B or b, and C or c), if we know the epistasis is triggered by two alleles, then there are 12 possible solutions: (A, B), (A, b), (A, C), (A, c), (a, B), (a, b), (a, C), (a, c), (B, C), (B, c), (b, C), and (b, c). Without the information about the complexity of the epistasis, i.e., the number of explanatory variables involved, which could be either two or three alleles, then the total number of solutions becomes 20. In general, if the total number of explanatory variables is p, and the epistasis could involve between two to p variables, then the number of possible solutions is $(3^p - 2p - 1)$. For 30 explanatory variables, this number is 206 trillion, so it would be computationally prohibitive to enumerate all possible epistases to find the true one. *Third*, it would be much more challenging to validate epistasis models through combinatorial experiments, since the process of validating even a small number of epistases candidates through field or lab experiments is usually prohibitively expensive, time-consuming, and labor intensive.

A related problem in machine learning and statistics is feature selection (Guyon and Elisseeff (2003); Unler and Murat (2010)), which has interesting similarities and differences with epistasis detection. The problem of feature selection is concerned with selecting a subset from a larger set of explanatory variables to explain the response variable, with a goal of obtaining a simplified model. Although both problems deal with variable selection, they have different assumptions and objectives. Feature selection assumes that the response variable can be sufficiently explained by a parsimonious subset of significant variables, and the objective is to make a selection that includes significant variables and excludes insignificant ones; the cost of excluding a significant variable is usually much larger than that of including an insignificant variable. In contrast, epistatic detection assumes that a set of explanatory variables are already known to have individual additive effects on the response variable, but an unknown combination of a few explanatory variables has an additional epistatic effect when they take certain values, and the objective is to find such combination in its exact form; the cost of selecting a variable that does not belong to the epistasis is as high as failing to select one that does. In fact, the study of epistatic effects requires efficient algorithms for both feature selection and epistasis detection. The genotype data sets usually contain many thousands or even millions of genes, with thousands of observed phenotype responses. Feature selection can be used first to select a small subset of genes, and then an epistasis detection algorithm can be applied to detect the epistatic interactions among these significant genes. This two-step approach was used in our simulation study to test the effectiveness of our epistasis detection algorithm.

Multiple methods have been proposed for detecting epistasis in genome-wide two- or multiple-locus interactions. Most of these methods focused on second-order epistasis through exhaustive search (Wan et al. (2010); Zhang et al. (2010); Piriyapongsa et al. (2012); Sluga et al. (2014); Gonzàlez-Domìnguez et al. (2015)). Evans et al. (Evans et al. (2006)) pointed out that even an exhaustive search of two-locus pairs across the genome would identify loci that would not have been identified using a single-locus search. Since many complex diseases have been associated with the interactions of multiple genes (Collins et al. (2013); Maher (2008); Moore et al. (2010)), several studies have targeted higher order epistasis (Ritchie et al. (2001); Kässens et al. (2002); Leem et al. (2014); Xie et al. (2012)). Almost all these studies resorted to heuristic algorithms with non-exhaustive search, due to the complexity of the problem. For example, the EDCF method (Xie et al. (2012)) is based on the clustering of relatively frequent items; the algorithm in (Leem et al. (2014)) runs k-means clustering algorithms on all single nucleotide polymorphisms (SNPs) and then applies information theory to examine the candidates from each cluster; MegaSNPHunter (Wan et al. (2009)) and SNPRuler (Wan et al. (2010)) used machine learning techniques; Ritchie et al. (Ritchie et al. (2001)) proposed the model-free and non-parametric MDR method, which first reduces the dimension of genotype predictors from n to one and then evaluates the one-dimension predictor for its ability to classify disease status through cross-validation and permutation testing; and Yang et al. (Jin-Shu and Wei-Jun (2008)) used a local search heuristic that swapped one pair of genes at a time, which was shown to outperform the Bayesian Epistasis Association Mapping method (Zhang and Liu (2007)). A summary of machine learning approaches for epistasis detection can be found in Upstill-Goddard et al. (2012).

The approach we propose in this paper was designed to detect multi-way epistasis, and it exhaustively searches all possible solutions in an efficient manner by taking advantage of combinatorial optimization modeling and solution techniques. This model can detect not only the complexity of the epistasis, but also their exact combination that triggers the epistatic effect. We also designed a heuristic algorithm for solving data sets with large numbers of genes and observations. Both the optimization model and the new algorithm were tested in a simulation study using a soybean data set. Results suggested that the new algorithm was able to find the global optimal solution for epistasis detection in 3,821 out of 4,000 independent simulation repetitions, each within 600 seconds, which was a significant improvement over two alternative algorithms compared in the simulation experiments.

1.2 Proposed Epistasis Detection Approach

In this section, we propose our approach for detecting epistatic effects from explanatory and response variables. We first formally give the problem statement in Section 2.2.2, and then present in Section 1.2.2 a local search heuristic that finds a local optimal solution by swapping one pair of genes at a time. In Section 1.2.3, we cast the epistasis detection problem as an MILP model, and in Section 1.2.4, we design an algorithm for data sets with large numbers of genes and observations by combining the MILP model with feature selection and the local search heuristic.

1.2.1 Problem Statement

We start by reviewing the multiple linear regression model, which can be used to capture additive effects of individual genes:

$$y_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + \epsilon_i, \forall i \in \{1, ..., n\}. \tag{1.1}$$

The notations used in the model include:

- p: the number of explanatory variables (genes)
- n: the number of observations (individuals)
- X_{ij}: the explanatory variable j for observation i, which is assumed to be binary
- y_i: the response variable for observation i (observed phenotype)
- β_0: the intercept coefficient
- β_j: the additive effect coefficient, which is the differential effect of $X_{ij} = 1$ over $X_{ij} = 0$ for any i
- ϵ_i: the residual effect for observation i.

To capture the epistatic effects between two genes, say j_1 and j_2, beyond the additive effects, several studies (Cordell (2002a); Zhang and Xu (2005)) have introduced interactions terms to the multiple linear regression model:

$$\begin{aligned} y_i = {} & \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + X_{ij_1}X_{ij_2}\gamma_{11} + X_{ij_1}(1 - X_{ij_2})\gamma_{10} \\ & +(1 - X_{ij_1})X_{ij_2}\gamma_{01} + (1 - X_{ij_1})(1 - X_{ij_2})\gamma_{00} + \epsilon_i, \forall i \in \{1, ..., n\}, \end{aligned} \tag{1.2}$$

where $\gamma_{k_1k_2}$ is the magnitude of the epistatic effect triggered by the combination of $X_{ij_1} = k_1$ and $X_{ij_2} = k_2$. Although Model (1.2) can quantify the effects of interactions between the two genes, it cannot determine by itself which two genes have such interactions, and it does not apply to multi-way epistatic interactions.

We now define a more generalized model to represent the epistatic effect:

$$y_i \quad = \quad \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + b \cdot I\left(\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) = \sum_{j=1}^{p} \lambda_j\right) + \epsilon_i, \forall i \in \{1, ..., n\}. \tag{1.3}$$

Here, b is the magnitude of the epistatic effect; $I(\cdot)$ is the indicator function, which is equal to 1 if the statement in the parentheses is true and 0 otherwise; and λ_j and μ_j are binary variables that define the epistasis, which is triggered if and only if $X_{ij} = 1, \forall j \in \{j : \lambda_j = 1\}$ and $X_{ij} = 0, \forall j \in \{j : \mu_j = 1\}$, for any i. In this model, for a given set of observations $X \in \mathbb{B}^{n,p}$ and $y \in \mathbb{R}^n$, we are trying to infer not only β_0 and β_j but also b, λ_j, and μ_j for all $j \in \{1, ..., n\}$. If the epistasis is present as defined in Model (3.40), then correctly inferring these parameters will lead to a smaller prediction error than a basic multiple linear regression Model (1.1) would. Compared with Model (1.2), Model (3.40) is able to reveal not only the complexity of the epistasis (the number of genes that are involved) but also the exact combination of diallelic variants that triggers the effect.

The key to solving Model (3.40) is λ and μ that represent the epistasis. Once these two variables are revealed, then solving for all other variables becomes as straightforward as solving a basic multiple linear regression model. First, we calculate $z_i = \begin{cases} 1 & \text{if } \sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) = \sum_{j=1}^{p} \lambda_j \\ 0 & \text{otherwise} \end{cases}, \forall i \in \{1, ..., n\}$, then we define $\hat{X} = [1_{n\times 1}, X, z]$, which can be used to estimate the response variables as $\hat{y} = \hat{X}(\hat{X}^\top X)^{-1}\hat{X}^\top y$; the other variables β_0, β, and b can also be calculated accordingly. We denote the root mean square error (RMSE) as a function of λ and μ as $\zeta(\lambda, \mu) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$, which evaluates the quality of the solution λ and μ. The true epistasis is expected to result in a lower RMSE than a false one.

1.2.2 Local Search Heuristic

In this section, we present a heuristic algorithm that searches for a local optimal solution to Model (3.40), which is similar with the PathSeeker algorithm (Jin-Shu and Wei-Jun (2008)) in essence but is more flexible. We will refer to this algorithm as $(\lambda^*, \mu^*) = \mathcal{L}(X, y, \lambda^0, \mu^0)$, where the input parameters of the algorithm include the data set (X, y) and an optional initial solution (λ^0, μ^0) and the output is a local optimal solution (λ^*, μ^*) that represents the epistasis. The algorithm iteratively swaps genes in and out of the incumbent solution, one pair at a time, to achieve the minimal RMSE. A major difference between our local search heuristic and the PathSeeker algorithm is that the latter assumes that the complexity of the epistasis is known, whereas the former assumes that we only know the upper bound, denoted as K, rather than the actual complexity of the epistasis. In the incumbent solution, the local search heuristic always keeps track of K indices for the K potential genes responsible for the epistasis, and uses a dummy index 0 as a place holder when fewer than K genes are found to be responsible. As such, our local search heuristic is able to take an initial solution with the wrong complexity and find its way towards a local optimal solution to the complexity and composition of the epistasis. Details of this heuristic algorithm are described as follows.

```
**Local search heuristic**
**Input parameters:** $X \in \mathbb{B}^{N\times p}$, $y \in \mathbb{R}^{N\times 1}$, and optionally $(\lambda^0, \mu^0)$.
**Output decisions:** $(\lambda^*, \mu^*)$.
**Start:** If the optional initial solution $(\lambda^0, \mu^0)$ is provided, then use it as the incumbent solution: $(\lambda^* = \lambda^0, \mu^* = \mu^0)$, otherwise initialize the incumbent solution $(\lambda^*, \mu^*)$ as two random binary vectors such that $\lambda_j^* + \mu_j^* \leq 1, \forall j \in \{1, ..., p\}$ and $\sum_{j=1}^{p} (\lambda_j^* + \mu_j^*) = K$. Evaluate its RMSE $\zeta(\lambda^*, \mu^*)$. Define $\mathcal{J} = \{j : \lambda_j^* + \mu_j^* = 1\}$ and, if necessary, extend its cardinality to $K$ by adding $K - \sum_{j=1}^{p} (\lambda_j^* + \mu_j^*)$ elements of $\{0\}$.
**while** $\mathcal{J} \neq \emptyset$ **do**
    **for** $j \in \{1, ..., p\} \backslash \mathcal{J}$ **do**
        For all $k \in \{1, ..., p\}$, define $\hat{\lambda}_k^j = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k = \mathcal{J}_1 \\ \lambda_j^* & \text{otherwise} \end{cases}$, $\hat{\mu}_k^j = \begin{cases} 0 & \text{if } k = j \\ 0 & \text{if } k = \mathcal{J}_1 \\ \mu_k^* & \text{otherwise} \end{cases}$, $\tilde{\lambda}_k^j = \begin{cases} 0 & \text{if } k = j \\ 0 & \text{if } k = \mathcal{J}_1 \\ \lambda_j^* & \text{otherwise} \end{cases}$, and $\tilde{\mu}_k^j = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k = \mathcal{J}_1 \\ \mu_k^* & \text{otherwise} \end{cases}$, where $\mathcal{J}_1$ refers to the first element in the set $\mathcal{J}$. Evaluate $\zeta(\hat{\lambda}^j, \hat{\mu}^j)$ and $\zeta(\tilde{\lambda}^j, \tilde{\mu}^j)$.
    **end for**
    Evaluate $\zeta(\lambda^0, \mu^0)$, where $\lambda_k^0 = \begin{cases} 0 & \text{if } k = \mathcal{J}_1 \\ \lambda_k^* & \text{otherwise} \end{cases}$, $\mu_k^0 = \begin{cases} 0 & \text{if } k = \mathcal{J}_1 \\ \mu_k^* & \text{otherwise} \end{cases}$, $\forall k \in \{1, ..., p\}$.
    Evaluate $\zeta(\lambda^1, \mu^1)$, where $\lambda_k^1 = \begin{cases} \mu_{\mathcal{J}_1}^* & \text{if } k = \mathcal{J}_1 \\ \lambda_k^* & \text{otherwise} \end{cases}$, $\mu_k^1 = \begin{cases} \lambda_{\mathcal{J}_1}^* & \text{if } k = \mathcal{J}_1 \\ \mu_k^* & \text{otherwise} \end{cases}$, $\forall k \in \{1, ..., p\}$.
    **if** $\min_{j \in \{1,...,p\}} \{\zeta(\hat{\lambda}^j, \hat{\mu}^j), \zeta(*, \tilde{\mu}^j), \zeta(\lambda^0, \mu^0), \zeta(\tilde{\lambda}^1, \mu^1)\} < \zeta(\lambda^*, \mu^*)$ **then**
        Update the incumbent solution $(\lambda^*, \mu^*)$ with the one that achieved the smallest RMSE.
        Update $\mathcal{J} = \{j : \lambda_j^* + \mu_j^* = 1\}$ and, if necessary, extend its cardinality to $K$ by adding $K - \sum_{j=1}^{p} (\lambda_j^* + \mu_j^*)$ elements of $\{0\}$.
    else
        Remove the first element in $\mathcal{J}$: $\mathcal{J} \leftarrow \mathcal{J}\backslash\mathcal{J}_1$.
    end if
end while
```

1.2.3 Optimization Model

We cast Model (3.40) as the following mixed integer optimization problem.

$$\min_{\beta_0,\beta,b,\lambda,\mu,z} \quad \sum_{i=1}^{n}\left| y_i - \left(\beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + b\cdot z_i\right)\right| \tag{1.4}$$

$$\text{s. t.} \quad \sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) \geq -p(1-z_i) + \sum_{j=1}^{p}\lambda_j \quad \forall i \in \{1,...,n\} \tag{1.5}$$

$$\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) \leq \sum_{j=1}^{p}\lambda_j - 1 + p\cdot z_i \quad \forall i \in \{1,...,n\} \tag{1.6}$$

$$\lambda_j + \mu_j \leq 1 \quad \forall j \in \{1,...,p\} \tag{1.7}$$

$$\lambda_j, \mu_j, z_i \in \{0,1\} \quad \forall i \in \{1,...,n\}, \forall j \in \{1,...,p\} \tag{1.8}$$

$$\beta_0, \beta_j, b \text{ free} \quad \forall j \in \{1,...,p\}. \tag{1.9}$$

Here, the Objective (2.2) is to minimize the sum of positive and negative errors between predicted responses and actual observations. We use this objective function, as opposed to the commonly used RMSE in linear regression models (Armstrong and Collopy (1992); Montgomery et al. (2015); Chatterjee and Hadi (2015)), because it can be linearized and is computationally more tractable. Constraints (3.75) and (2.4) jointly define a binary variable z_i that checks the presence of the epistatic effect in each observation: $z_i = 1$ if and only if $\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) = \sum_{j=1}^{p}\lambda_j$, for all $i \in \{1,...,n\}$. Constraint (3.75) ensures the "only if" direction: if $z_i = 1$, then $\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) \geq \sum_{j=1}^{p}\lambda_j$; since $\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) \leq \sum_{j=1}^{p} X_{ij}\lambda_j \leq \sum_{j=1}^{p}\lambda_j$ is always true, it leads to the equation $\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) = \sum_{j=1}^{p}\lambda_j$. Conversely, Constraint (2.4) enforces the "if" direction: if $\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) \geq \sum_{j=1}^{p}\lambda_j$, then $z_i = 1$. Constraint (2.5) means that the epistasis cannot logically require the jth explanatory variable to be one ($\lambda_j = 1$) and zero ($\mu_j = 1$) at the same time. Constraints (2.6) and (2.7) define the appropriate types of the decision variables.

Due to the absolute value function and the bilinear term $b \cdot z_i$ in the objective function, Model (2.2)-(2.7) is a nonlinear non-convex combinatorial optimization problem, generally hard to solve. In the following, we reformulate Model (2.2)-(2.7) into an equivalent mixed integer linear program

(MILP) by introducing three sets of new variables. For all $i \in \{1, ..., n\}$, non-negative variables e_i^+ and e_i^- denote the positive and negative part of the prediction error for observation i, respectively, and the variable w_i is defined to be equal to $b \cdot z_i$. We also assume that we know the upper and lower bounds of the magnitude of the epistatic effect: $\underline{b} \leq b \leq \overline{b}$, which is necessary to linearize the bilinear term $b \cdot z_i$. We will refer to the following MILP (1.10)-(1.18) as $\mathcal{M}(X, y)$, with X and y being observation parameters.

$$\begin{aligned}
\min_{\beta_0,\beta,b,e_i^+,e_i^-,\lambda,\mu,z,w} \quad & \sum_{i=1}^{n}(e_i^+ + e_i^-) && && (1.10)\\
\text{s.t.} \quad & y_i - \left(\beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + w_i\right) = e_i^+ - e_i^- && \forall i \in \{1, ..., n\} && (1.11)\\
& w_i \leq \overline{b} z_i && \forall i \in \{1, ..., n\} && (1.12)\\
& w_i \geq \underline{b} z_i && \forall i \in \{1, ..., n\} && (1.13)\\
& w_i \leq b - \underline{b}(1 - z_i) && \forall i \in \{1, ..., n\} && (1.14)\\
& w_i \geq b - \overline{b}(1 - z_i) && \forall i \in \{1, ..., n\} && (1.15)\\
& \text{Constraints (3.75)-(2.6)} && && (1.16)\\
& e_i^+, e_i^- \geq 0 && \forall i \in \{1, ..., n\} && (1.17)\\
& \beta_0, \beta_j, b, w_i \text{ free} && \forall i \in \{1, ..., n\}, \forall j \in \{1, ..., p\} && (1.18)
\end{aligned}$$

Here, the Objective (1.10) is to minimize the positive and negative errors between predicted and actual observations, which is equivalent to Objective (2.2). These two error terms are defined in Constraint (1.11), at least one of which must be zero in the optimal solution (otherwise the objective value would have been improved by reducing the two error terms simultaneously). Constraints (1.12)-(1.15) are equivalent to the equation $w_i = b \cdot z_i$ when $\underline{b} \leq b \leq \overline{b}$, which is a commonly used reformulation linearization technique (Sherali and Adams (2013)). Constraints (1.17) and (1.18) define the appropriate types of the decision variables.

1.2.4 Algorithm for Large Data Sets

In theory, the MILP Model (1.10)-(1.18) can be used to detect epistasis for data sets of any dimensions. In practice, however, solving such model using the existing branch-and-bound algorithms (Lawler and Wood (1966)) and solvers becomes increasingly time-consuming as the dimensions of the problem grow. This is a particularly relevant restriction in the big data era, when high throughput genotyping and phenotyping technologies are able to collect data at increasingly high efficiency. To address this challenge, a feature selection algorithm can be used to reduce the number of genes to a few dozen (assuming the validity of a parsimonious model). In the following, we introduce a new algorithm for solving Model (1.10)-(1.18) for a few dozen genes and a large number of observations, say N.

The idea of the algorithm is to iteratively solve Model (1.10)-(1.18) with small samples of observations and refine its solutions with the local search heuristic. The algorithm starts by initializing an incumbent solution (λ^*, μ^*) and goes through two iterative steps. In Step 1, Model (1.10)-(1.18) is solved on a small subset of observations to provide a candidate epistasis solution, which is refined in Step 2 using the local search heuristic. This candidate epistasis will replace the incumbent solution if it achieves a lower RMSE. The Step 1 and Step 2 loop continues until the stopping criteria are met, which could be defined based on the RMSE of the incumbent solution, the number of iterations, or computation time. Details of this algorithm are described as follows and diagrammed in Figure 1.1.

New Algorithm
Input parameters: $X \in \mathbb{B}^{N\times p}$ and $y \in \mathbb{R}^{N\times 1}$.
Output decisions: $\lambda^* \in \mathbb{B}^{p\times 1}$ and $\mu^* \in \mathbb{B}^{p\times 1}$.
Start: Determine the sample size n. Initialize the iteration counter $t = 0$ and the incumbent solution $(\lambda^* = 0^{p\times 1}, \mu^* = 0^{p\times 1})$. Evaluated the RMSE of the incumbent solution as $\zeta(\lambda^*, \mu^*)$. Go to Step 1.
Step 1: Detection. Increase t by 1: $t \leftarrow t+1$. Randomly select a set $\mathcal{N}$ such that $\mathcal{N} \subseteq \{1, 2, ..., N\}$ and $|\mathcal{N}| = n$. Solve model $\mathcal{M}(X_{\mathcal{N}}, y_{\mathcal{N}})$, where $X_{\mathcal{N}}$ is a matrix with those rows from the input X whose indices belong to the set $\mathcal{N}$, and ditto for $y_{\mathcal{N}}$. Let (λ^t, μ^t) denote part of the optimal solution from $\mathcal{M}(X_{\mathcal{N}}, y_{\mathcal{N}})$. Go to Step 2.
Step 2: Refinement. Refine the solution using the local search heuristic $(\lambda^t, \mu^t) = \mathcal{L}(X, y, \lambda^t, \mu^t)$. If $\zeta(\lambda^t, \mu^t) < \zeta(\lambda^*, \mu^*)$, then update the incumbent (λ^*, μ^*) as (λ^t, μ^t).
Check point: If Stopping criteria are met then finish the algorithm; otherwise go back to Step 1.

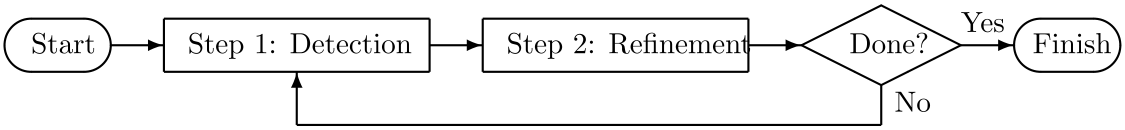


Figure 1.1: Diagram of the new algorithm for epistasis detection.

Remark 1: At the starting point, the sample size n should be small enough to make model $\mathcal{M}(X_{\mathcal{N}}, y_{\mathcal{N}})$ computationally tractable and large enough to include sufficient information about the epistasis. A sensitivity analysis of this parameter can be found in Section 1.3.2.
Remark 2: In this algorithm and in the simulation experiments, the RMSE function $\zeta(\lambda, \mu)$ was calculated for all individuals in the data set rather than a small sample.

Remark 3: In Step 1, the random selection of set $\mathcal{N}$ can be made more representative of the set $\{1, ..., N\}$ by first calculating the estimation error $|y_i - \hat{y}_i|$ for all $i \in \{1, ..., N\}$ from the previous iteration and then choosing $\mathcal{N}$ to include individuals whose estimation errors are representative of those of the set $\{1, ..., N\}$.

Remark 4: The stopping criteria could include the RMSE falling below a predetermined threshold or the number of iterations or computation time exceeding a predetermined value.

1.3 Computational Experiments

We conducted computational experiments to test the effectiveness of the proposed approach.

1.3.1 Data and Algorithm Implementation

We collected a soybean genotype data set from SoyBase, which consisted of 42,509 individuals of soybean cultivars each having 20,087 genes (represented by SNPs). In the simulation, we assumed that a parsimonious set of significant genes are responsible for a certain phenotype, and a feature selection algorithm is available to select a small set of genes that includes all significant genes and a number of insignificant ones. This assumption is reasonable because the purpose of the simulation is to test the effectiveness of the epistasis detection algorithms rather than that of feature selection algorithms.

We randomly generated ground truth values for β_0, β, b, λ, μ and ϵ and used these parameters to generate the phenotype response data y according to Model (3.40). The distributions of these random parameters are described as follows.

- β_0: normal distribution $\mathcal{N}(0, 30^2)$.
- β_i: uniform distribution $\mathcal{U}(15, 30)$, for all $j \in \{1, ..., p_0\}$.
- β_j: 0, for all $j \in \{p_0 + 1, ..., p\}$.
- b: uniform distribution $\mathcal{U}(15, 30)$.
- (λ, μ): first initialize λ and μ as zero vectors; then generate a random set $\mathcal{J} \subseteq \{1, ..., p\}$ that contains K unique indices; finally for all $j \in \mathcal{J}$, assign either $\lambda_j = 1$ or $\mu_j = 1$ with equal probability.
- ϵ_i: normal distribution $\mathcal{N}(0, \sigma^2)$, independent and identically distributed for all $i \in \{1, ..., n\}$; different values of σ were being used in the simulation.

The reason that we included a new parameter p_0 is two-fold. First, when a feature selection algorithm is used to reduce the number of explanatory variables from 20,087 to a small number, say

$p = 40$, it may include some, say $p_0 = 30$, significant variables that have positive additive effects as well as some, say $p - p_0 = 10$, insignificant variables that have negligible additive effects. Second, certain genes may contribute to an epistatic effect without having noticeable individual additive effects themselves.

The new model and algorithm proposed in Sections 1.2.3 and 1.2.4 were implemented in Octave, using CPLEX as the MILP solver for Model (1.10)-(1.18). Two alternative algorithms were also implemented in Octave for comparison: Enumeration and PathSeeker. The Enumeration algorithm evaluates $\zeta(\lambda, \mu)$ for all feasible values of (λ, μ) in a brute force manner. The PathSeeker algorithm was presented in (Jin-Shu and Wei-Jun (2008)) and is similar with our local search heuristic in Section 1.2.2. The implementation of these two algorithms was based on our understanding of the ideas reported in the literature and customized to fit the data format and simulation setting of our experiments. We made an honest effort to implement the algorithms in the most efficient manner that we were capable of, so that the performance differences reflected the differences in efficiency of algorithmic design more than the differences in efficiency of implementation.

1.3.2 Sensitivity Analysis of Model (1.10)-(1.18)

Table 1.1: Parameters for the sensitivity analysis

Parameter	Meaning	Values
K	Upper bound of the complexity of epistasis	{2, 3, 4, 5}
$p(p_0)$	Number of (significant) explanatory variables	{25(15), 50(40), 75(65), 100(90)}
n	Number of observations	{100, 200, 300, 400}
σ	Standard deviation of random error	{0, 2, 4, 6}

We conducted a sensitivity analysis of four parameters that could affect the effectiveness of the optimization Model (1.10)-(1.18), which are summarized in Table 1.1. The values for these parameters were chosen to explore the ranges of parameters within which the model is computationally tractable. A full-factorial experiment requires solving the model for $4^4 = 256$ different sets of parameters, and we ran 10 repetitions for each set. The optimization model used a small subset of the genotype and phenotype data, consisting of p genes and n individuals randomly selected from the full data set. We set 1,800 seconds as the time limit for CPLEX to solve the optimization model. We also applied the local search heuristic to refine the solution from the optimization model, which used a larger subset of the data, consisting of the p genes and all 42,509 individuals. Three values were recorded for each repetition: computation time, RMSE of the solution from Model (1.10)-(1.18), and RMSE of the solution from the local search heuristic (using the solution from

Model (1.10)-(1.18) as the starting point). Almost half of the time, the 1,800-second time limit for Model (1.10)-(1.18) was reached. The total computation time for this sensitivity analysis was approximately 40 CPU days.

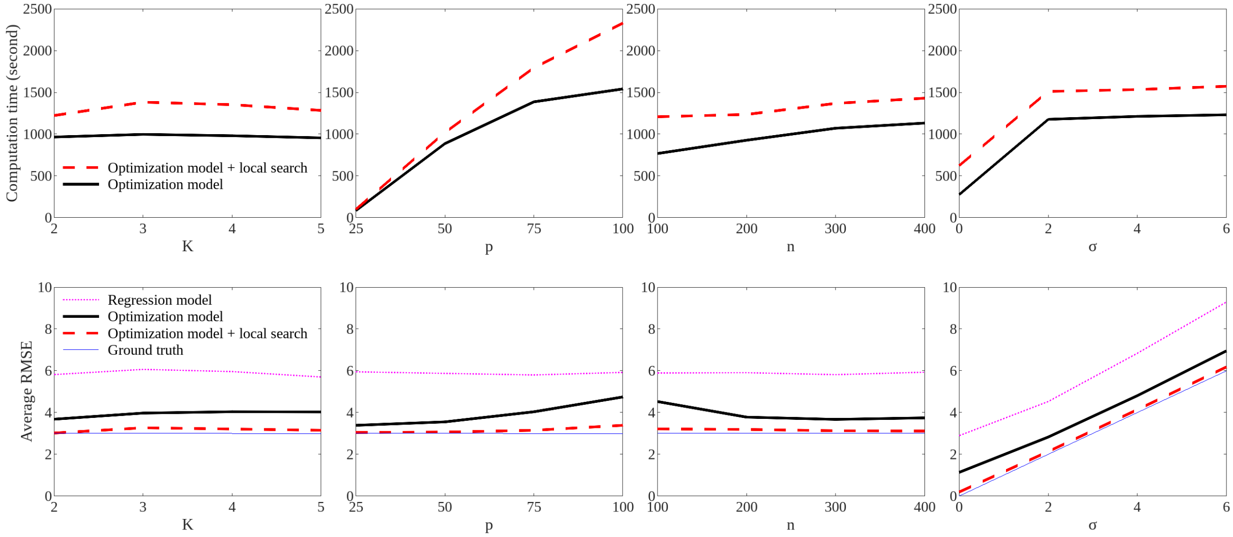


Figure 1.2: Results of sensitivity analysis. The four subfigures in the top row show the average computation times (capped at 1,800 seconds) of solving Model (1.10)-(1.18) using CPLEX and the local search heuristic for different parameter settings. The four subfigures in the bottom row show the average RMSEs using the basic multiple linear regression Model (1.1), the optimization Model (1.10)-(1.18), the optimization Model (1.10)-(1.18) and local search heuristic, and Model (3.40) with ground truth.

Results of the sensitivity analysis are summarized in Figure 1.2. We point out five noticeable observations. (1) The model is insensitive to the complexity of the epistasis, both in terms of computation time and accuracy of results. This is a desirable property, compared with exhaustive search algorithms whose computational complexity is an exponential function of the complexity of the epistasis. (2) The model is very sensitive to the number of candidate genes, p. When p exceeds 50, the computation time is likely to exceed 1,800 seconds and the accuracy of the result gets noticeably worse. (3) Computation time of the model is somewhat sensitive to the number of individuals, n. A sample size as small as $n = 100$ results in high RMSEs; when n exceeds 200 a larger sample size does not further reduce the RMSE. (4) Without random error, Model (1.10)-(1.18) can be solved efficiently to a reasonable accuracy. The computation time of the model is sensitive to the standard deviation of the random error. (5) Within 1,800 seconds, CPLEX is able to find a high quality solution that is either global optimal or in its neighborhood. In the latter case, the local search heuristic is effective and efficient in finding the global optimal solution using the solution from the optimization model as a starting point.

1.3.3 Comparison of Algorithms

We compared our new algorithm presented in Section 1.2.4 with the Enumeration and PathSeeker algorithms using a total of 4,000 independent repetitions of simulation, including 1,000 for each integer value of K between 2 and 5. In each repetition, a subset of genotype data was randomly selected from the original data set, consisting of $p = 40$ candidate genes (including $p_0 = 30$ significant ones) for all 42,509 individuals; the phenotype data was generated using the same ground truth assumptions of additive and epistatic effects described in Section 1.3.1. The same genotype and phenotype data set was used for all three algorithms for epistasis detection in each repetition.

For Model (1.10)-(1.18) in the new algorithm, we used $n = 6 \times 2^K$ as the sample size, which is small enough to keep the model tractable and large enough to make the epistasis detectable by including, on average, 6 instances of individuals that receive the epistatic effect. Rather than letting CPLEX solve the model to global optimality, we used very loose stopping critera: a 50% optimality gap or a 60-second time limit. The rationale is two-fold. First, as suggested by the sensitivity analysis results, the local search heuristic does not require a very high quality starting point to find the global optimal solution, whereas CPLEX would take much longer to arrive at global optimality. Second, since Model (1.10)-(1.18) is using a small sample of all individuals in each iteration, it may be a better use of the time to solve the model multiple iterations than to solve it once to a smaller optimality gap. For the alternative algorithms, we created a list of all possible solutions of epistases with the complexity ranging from 2 to 5. The Enumeration algorithm went through the list in a random order and checked the RMSEs of all the solutions. The PathSeeker algorithm was applied to all the solutions on the list in a random order as starting points. For all three algorithms, we used ($\lambda^* = 0^{p\times 1}, \mu^* = 0^{p\times 1}$) as the initial incumbent solution, which leads to the basic multiple linear regression Model (1.1) without consideration of the epistatic effect, and we recorded how the RMSE of the incumbent solution was improving over time. A time limit of 600 seconds was imposed for all 3 algorithms. The total computation time for the comparison experiment was approximately 63 CPU days.

Figure 1.3 compares the three algorithms with respect to their ability to find improving incumbent solutions over time. We point out four observations. (1) In 10 out of 12 subfigures, the new algorithm reached global optimality within 200 seconds, and the other two subfigures took no more than 500 seconds; whereas the other two alternative algorithms were able to reach global optimality only at the 10th percentile of RMSE for the 1,000 repetitions when $K = 2$. (2) In 10 out of 12 subfigures, the new algorithm was able to reach global optimality in one iteration. The two exceptions were at the 90th percentile with $K = 3$ and $K = 4$, which required 2 or 3 more iterations. (3) Solution quality of the two alternative algorithms deteriorated as the epistasis complexity increases, but the new algorithm did not show such sensitivity. (4) PathSeeker outperforms the Enumeration algorithm in almost all cases, due to its effectiveness in finding local optimal solutions rather than randomly exploring the solution space. The new algorithm outperforms PathSeeker

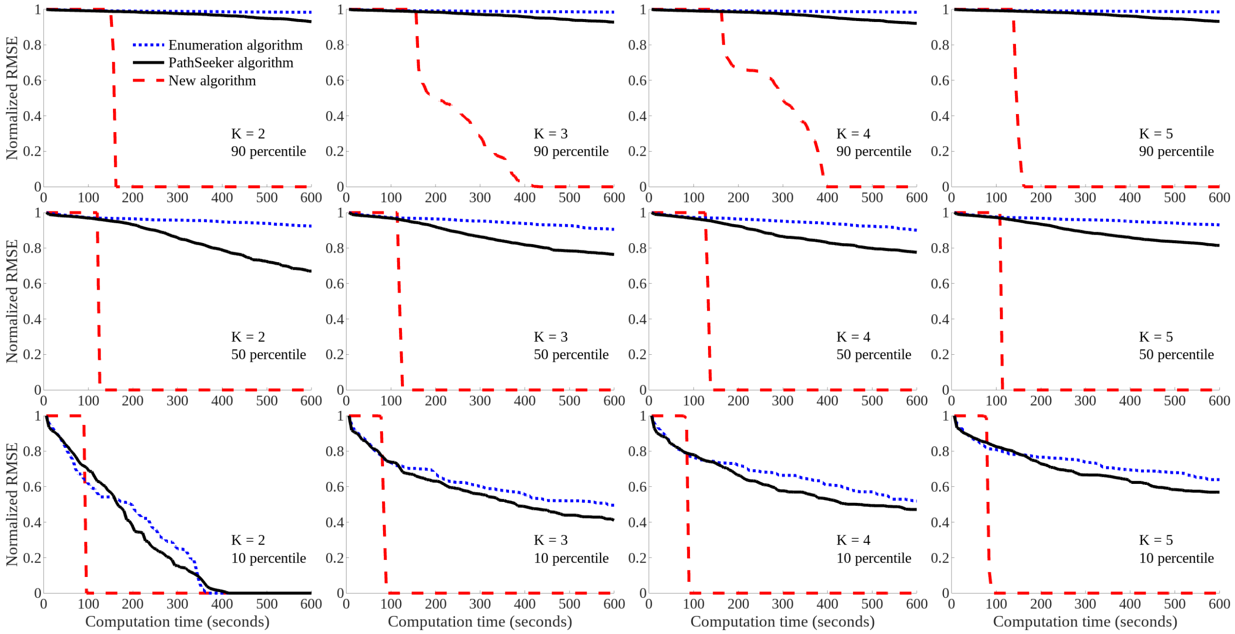


Figure 1.3: Comparison of the three algorithms' dynamic progress. The four columns of subfigures correspond to different complexity levels of the ground truth epistases, and the three rows correspond to different percentiles of results among the 1,000 repetitions. The RMSE values are normalized as $(\zeta - \zeta_0)/(\zeta_1 - \zeta_0)$, where ζ is the RMSE of the solution obtained by any of these algorithms, ζ_0 is the RMSE of the global optimal solution, and ζ_1 is the RMSE of Model (1.1) without consideration of the epistatic effect. Since the incumbent epistasis was empty at the beginning and improves over time, the normalized RMSE should start from 1 and gradually decrease to 0 or somewhere in between.

by feeding the local search heuristic with high quality incumbent solutions from the optimization model as starting points.

Table 1.2 compares the solution quality of the three algorithms at the end of the 600-second time limit. The PathSeeker algorithm slightly outperformed the Enumeration algorithm, yet they both struggled to detect the true epistatic effect, especially for $K \geq 4$. The solution quality also deteriorated for more complex epistases, which is due to the exhaustive search nature of these algorithms. On the contrary, the new algorithm had an overall 95.5% (3,821 out of 4,000) chance of finding the global optimal solution, and this success rate got even higher for more complex epistases. This can also be attributed to the nature of the new algorithm, which detects the epistasis by looking for discrepancies between individuals that receive and not receive the epistatic effect. When the epistasis is more complex, a smaller subset of individuals receive the effect, and their discrepancies from other individuals are more outstanding and harder to be diluted and explained away by additive effects.

Table 1.2: Comparison of the three algorithms' solution quality at the end of the 600-second time limit. The values are the numbers of times that the normalized RMSEs fall into the labeled ranges.

Algorithm	K	Normalized RMSE			
		$(-\infty, 0]$	$(0, 0.01]$	$(0.01, 0.1]$	$(0.1, 1]$
Enumeration algorithm	2	143	5	4	848
	3	16	3	13	968
	4	0	3	7	990
	5	0	0	2	998
PathSeeker algorithm	2	124	41	56	779
	3	4	4	24	968
	4	2	3	13	982
	5	0	1	4	995
New algorithm	2	888	112	0	0
	3	958	5	14	23
	4	976	0	3	21
	5	999	0	1	0

These results suggested that the new algorithm demonstrated substantial improvements over the two alternatives, and the main reason is the combination of the optimization Model (1.10)-(1.18) and the local search heuristic. The former explores the solution space of all epistases by taking advantage of the efficient CPLEX solver for solving Model (1.10)-(1.18), and the latter takes an incumbent solution and refines it by finding its local optimal solution, which usually ends up being globally optimal.

1.4 Conclusion

The main contribution of this paper is a new approach for detecting the epistatic effect. At the core of this approach is a combinatorial optimization model that detects both the complexity of the epistasis and the exact combination of genes that triggers the effect. Our sensitivity analysis revealed that this model is most effective and computationally efficient for a few dozen genes and a couple of hundred observed individuals. We also designed a new algorithm to detect the epistatic effect for a large data set with tens of thousands of genes and individuals. First, a feature selection algorithm can be used to reduce the number of genes to a few dozen, and then small samples of individuals are iteratively drawn to feed the optimization model, the solutions from which will be subsequently refined using a local search heuristic. We conducted computational experiments using a soybean data set, which consisted of 20,087 genes and 42,509 individuals. When compared with two popular algorithms from the literature, the Enumeration and PathSeeker algorithms, the new algorithm demonstrated significant improvement in the effectiveness and efficiency of detecting multi-way epistases.

As a caveat, the proposed approach has several limitations. For example, the validity of the new algorithm depends on three assumptions: (1) Model (3.40) is a reasonable approximation of the ground truth, (2) the phenotype of interest is determined by a small number of significant genes, and (3) an effective feature selection algorithm is available to select a small set of genes that includes all the significant ones and possibly some insignificant ones. The model also assumes that the X matrix is binary, meaning that it only applies to explanatory variables that are qualitatively categorized rather than quantitatively measured. A potentially fruitful direction of future research is to extend the proposed Model (3.40) and the new algorithm to include multiple epistatic effects simultaneously.

Acknowledgements

The authors are grateful to the Associate Editor and three anonymous reviewers for their insightful and constructive feedback. This research was partially supported by the U.S. Department of Agriculture NIFA Award 2017-67007-26175, the National Science Foundation Award 1830478, and by the Plant Sciences Institute at Iowa State University.

1.5 References

Armstrong, J. S. and Collopy, F. (1992). Error measures for generalizing about forecasting methods: Empirical comparisons. *International journal of forecasting*, 8(1):69–80.

Bateson, W. (1909). *Mendel's Principles of Heredity*. Cambridge University Press.

Chatterjee, S. and Hadi, A. S. (2015). *Regression Analysis by Example*. John Wiley & Sons.

Collins, R. L., Hu, T., and et al., C. W. (2013). Multifactor dimensionality reduction reveals a three-locus epistatic interaction associated with susceptibility to pulmonary tuberculosis. *BioData Mining*, 6:1.

Combarros, O., Cortina-Borja, M., Smith, A. D., and Lehmann, D. J. (2009). Epistasis in sporadic Alzheimer's disease. *Neurobiology of Aging*, 30:1333–11349.

Cordell, H. J. (2002). Epistasis: What it means, what it doesn't mean, and statistical methods to detect it in humans. *Human Molecular Genetics*, 11:2463–2468.

Evans, D. M., Marchini, J., Morris, A. P., and Cardon, L. R. (2006). Two-stage two-locus models in genome-wide association. *PLoS Genetics*, 2:e157.

Gonzàlez-Domìnguez, J., Wienbrandt, L., and et al., J. C. K. (2015). Parallelizing epistasis detection in GWAS on FPGA and GPU-accelerated computing systems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 12:982–994.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Jin-Shu, Y. and Wei-Jun, Y. (2008). The complete mitochondrial genome sequence of the hydrothermal vent galatheid crab shinkaia crosnieri (crustacea: Decapoda: Anomura): A novel arrangement and incomplete trna suite.

Kässens, J. C., Wienbrandt, L., González-Domínguez, J., Schmidt, B., and Schimmler, M. (2002). High-speed exhaustive 3-locus interaction epistasis analysis on FPGAs. *Journal of Computational Science*, 9:131–136.

Lawler, E. L. and Wood, D. E. (1966). Branch-and-bound methods: A survey. *Operations research*, 14(4):699–719.

Leem, S., Jeong, H. H., Lee, J., Weea, K., and Sohn, K. (2014). Fast detection of high–order epistatic interactions in genome–wide association studies using information theoretic measure. *Computational Biology and Chemistry*, 50:19–28.

Mackay, T. F. and Moore, J. H. (2014). Why epistasis is important for tackling complex human disease genetics. *Genome Medicine*, 6:1.

Maher, B. (2008). The case of the missing heritability. *Nature*, 456:18–21.

Montgomery, D. C., Peck, E. A., and Vining, G. G. (2015). *Introduction to Linear Regression Analysis*. John Wiley & Sons.

Moore, J. H., Asselbergs, F. W., and Williams, S. M. (2010). Bioinformatics challenges for genome-wide association studies. *Bioinformatics*, 26:445–455.

Nagel, R. L. (2005). Epistasis and the genetics of human diseases. *Comptes Rendus Biologies*, 328:606–615.

Piriyapongsa, J., Ngamphiw, C., and et al., A. I. (2012). iLOCi: a SNP interaction prioritization technique for detecting epistasis in genome-wide association studies. *BMC Genomics*, 13(7):1.

Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F., and Moore, J. H. (2001). Multifactor–dimensionality reduction reveals high–order interactions among estrogen–metabolism genes in sporadic breast cancer. *The American Journal of Human Genetics*, 69:138–147.

Sherali, H. D. and Adams, W. P. (2013). *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, volume 31. Springer Science & Business Media.

Sluga, D., Curk, T., Zupan, B., and Lotric, U. (2014). Heterogeneous computing architecture for fast detection of SNP-SNP interactions. *BMC Bioinformatics*, 15:216.

Taylor, M. B. and Ehrenreich, I. M. (2014). Genetic interactions involving five or more genes contribute to a complex trait in yeast. *PLoS genetics*, 10(5).

Unler, A. and Murat, A. (2010). A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research*, 206(3):528–539.

Upstill-Goddard, R., Eccles, D., Fliege, J., and Collins, A. (2012). Machine learning approaches for the discovery of gene–gene interactions in disease data. *Briefings in bioinformatics*, 14(2):251–260.

Wan, X., Yang, C., and et al., Q. Y. (2009). MegaSNPHunter: a learning approach to detect disease predisposition SNPs and high level interactions in genome wide association study. *BMC Bioinformatics*, 10:1.

Wan, X., Yang, C., and et al., Q. Y. (2010). BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies. *The American Journal of Human Genetics*, 87(3):325–340.

Xie, M., Li, J., and Jiang, T. (2012). Detecting genome-wide epistases based on the clustering of relatively frequent items. *Bioinformatics*, 28:5–12.

Xu, S. and Jia, Z. (2007). Genomewide analysis of epistatic effects for quantitative traits in barley. *Genetics*, 175:1955–1963.

Zhang, X., Huang, S., Zou, F., and Wang., W. (2010). TEAM: efficient two-locus epistasis tests in human genome-wide association study. *Bioinformatics*, 26:i217–i227.

Zhang, Y. and Liu, J. S. (2007). Bayesian inference of epistatic interactions in case-control studies. *Nature genetics*, 39(9):1167.

Zhang, Y.-M. and Xu, S. (2005). A penalized maximum likelihood method for estimating epistatic effects of qtl. *Heredity*, 95(1):96.

Chapter2. Discovering Interacting Features for Predicting a Continuous Response

Paper is submitted to *Pattern Recognition*

Maryam Nikouei Mehr and Sigurdur Olafsson

Abstract

This paper concerns a new feature selection method for multiple linear regression models. While common feature selection methods put emphasis on the removal of what appear to be redundant or irrelevant features, we show that in some cases such features are important. We establish a computational framework for selecting a subset of features that have a significant interactions with the response and a subset of features that may have little interactions with the response which might lead them to be considered as redundant, but when combined with others they can be strongly predictive of the response. Removal of these features can thus result in poor predictions. The proposed method finds features with significant main effect using a standard feature selection method and implements a mixed integer linear programming model to recognize feature interactions and their exact combinations.

2.1 Introduction

Feature selection in machine learning is used as a method for reducing dimensionality of data based on specific evaluation criteria. The motivation for feature selection is well known (Guyon and Elisseeff (2003)). Removing features that are irrelevant and redundant may increase efficiency in learning tasks, improve prediction performance, and enhance comprehensibility of learned results. By applying feature selection techniques we can gain some insight into the process and can improve the computation requirement and prediction accuracy.

Feature selection algorithms fall in to two broad categories: filter and wrapper models (Das (2001); Kohavi and John (1997)). The principle criteria for feature selection in filter method is feature ranking techniques. Simplicity and good success in report for practical applications are reasons for using ranking techniques (Guyon and Elisseeff (2003)). This method uses a suitable ranking methods to rank features and a threshold to remove feature with ranking below the threshold. Wrapper methods considers the feature subset as a black box and the performance of the features as the objective function to evaluate and choose the best subset (Guyon and Elisseeff (2003)). It would consider exhaustive search to search the space of all possible subsets and choose the right predictors if the number of variables are limited. It could become an NP-hard problem as the number of features increases. As a result, wide range of search strategies including forward selection (FS) (Guyon and Elisseeff (2003); Koller and Sahami (1996)), backward stepwise selection (BS) (Cotter et al. (2001)), branch and bound (Dash and Liu (1997)), simulated annealing (Li and Liu (2008)), forest optimization algorithm (Ghaemi and Feizi-Derakhshi (2016)), and genetic algorithms (Frohlich et al. (2003); Zhuo et al. (2008)) are developed to find local optimal subsets (Kohavi and John (1997)).

Feature selection plays an important role in pattern recognition. For example, Mo and Lai (2019) proposed a robust jointly sparse regression method to obtain more discriminative information for image recognition by combining the locality of the manifold structure of the original structure of the original data, the orthogonality and the joint sparsity of the projection. Yin et al. (2019) considered feature learning and partially constrained cluster labels learning and presented a new multi-view clustering method. A novel manifold regularized optimization framework for multi-label feature selection is introduced by Zhang et al. (2019a). Zhang et al. (2019b) also addressed a new multi-label feature selection method to consider the effect of label redundancy on the evaluation of feature relevance. Solorio-Fernández et al. (2017) proposed a new unsupervised filter feature selection method for data sets containing both numerical and non-numerical features. Zhou et al. (2019) discussed a new online streaming feature selection method based on adaptive density neighborhood relation to handle streaming features in big data problems. Using maximal information entropy between features and class, and particle swarm optimiztion-support vector machines, Zheng and Wang (2018) developed a new feature selection algorithm. Considering L_p-norm least squares support vector machine,where feature selection and prediction are implemented simultaneously, Shao et al. (2018) developed a new feature selection method for problems with sample size smaller than the number of features of a data set. Methodological Programming has also significantly contributed to feature selection (Olafsson et al. (2008)). Bradley et al. (1998) introduced the idea of using parametric objective function in mathematical programming for feature selection. They also compared 2 mathematical representation of feature selection methods (Feature selection via Concave Minimization, Feature Selection via Support Vector Machines). Feature Selection via Support Vector Machines with 1-norm and 2-norm was used to measure the distance between the two parallel bounding planes. They suggested the use of SVM 1-norm instead of SVM 2-norm to reduce computational time. Springenberg et al. (2014) formulated the hyper-parameters and feature selection problem as a stochastic global optimization problem and applied cross-entropy to solve the problem.

As noted above, variable and feature selection are beneficial for the following reasons: simplifying data understanding, reducing training time and memory utilization, reducing dimensionality and improving predictions (Guyon and Elisseeff (2003)). Generally features may be categorized as follows (John et al. (1994)):

- Relevant: These features have strong relationship with the output and could not be dismissed in the subset selection.
- Irrelevant: These features do not influence output and mostly generated due to randomness.
- Redundant: These features can be expressed by other selected features.

It appears to be common practice in many machine learning projects to attempt to exclude irrelevant (Blum and Langley (1997); Radha and Muralidhara (2016)) and redundant features from their training datasets (Appice et al. (2004); Zeng et al. (2015)). The motivation is that irrelevant and redundant features results in poor performance of the predictive models. For example Talukdar et al. (2018) discussed a new feature selection method, KPLS-mRMR, to select maximum relevant and minimum redundant features based on KPLS regression coefficients. While this is often true, we show that including apparently irrelevant or redundant features does not always result in poor predictions due to feature interactions.

Feature interaction presents a challenge in feature selection (Cotter et al. (2001); Kira and Rendell (1992)). Features could be considered irrelevant based on its correlation with the response, while in combination with other features they could be considered relevant (Cotter et al. (2001); Kira and Rendell (1992)). On the other hand, a features could be considered redundant while in the presence of other features, but could yet be strongly related to the response (Kira and Rendell (1992)). Existing feature selection methods often implicitly assume feature independence and focus on removing irrelevant and redundant features. Based on that assumption, the motivation is that irrelevant features provide no useful information for response prediction and redundant features are already represented with currently selected features. However, apparently redundant and irrelevant features could be associated in feature interaction. Handling interactions is computationally difficult since using simple filter methods may not capture these interacting features (Guyon and Elisseeff (2003)). Some wrapper methods might be able to at least partially capture feature interactions but these methods usually test each feature subset for potential interactions which is time consuming and computationally expensive.

We argue that recognizing the presence of feature interaction is crucial in feature selection. Although a few studies have highlighted the importance of feature interaction, few feature selection methods have been developed to handle feature interaction. Zhao and Liu (2009) proposed an algorithm, INTERACT, that employs consistence-contribution to handle feature interaction and efficiently select relevant features. Zeng et al. (2015) presented an Interaction Weight based Feature Selection algorithm (IWFS), a novel feature selection method that considers interaction. They used the interaction weight factor to specify whether the information of a feature is redundant or interactive. Tang et al. (2018) proposed a new feature selection method, Five-way Joint Mutual Information (FJMI), that look for two to five-way interaction between feature and the class label. A practical heuristic to find feature interaction could be studied in Jakulin and Bratko (2003). Their proposed algorithm tries to find 2-way interaction (one feature and the class) and 3-way interaction (2 features and class)using information gain.

Different feature selection methods will vary in terms of their ability to identify interacting features. Some methods will explicitly penalize including correlated features, and we are not aware of any standard method that explicitly tries to identify interacting features. However, feature selection methods that directly evaluate how well a feature works for prediction may be able to implicitly pick up on such interactions. For example, sometimes feature selection is based on a Random Forest model, where the features are ranked according to how often they are selected to be used as part of a decision tree, that is, how well they work in interaction with other features in the tree. Such an approach should be able to pick up on interacting features but will not explicitly identify which features interact. Another well known approach that should be able to pick up interactions is Relief, which ranks features based on how well they perform in subsets of features to do instance-based (nearest neighbor) prediction. Again, this should allow us to pick up on interactions without explicitly identify the interacting features, but the concern is that the final feature selection might exclude one or more interacting feature because it is unknown that it work only as part of a pair. The unique element of the proposed approach is that it explicitly identifies interacting features that should be included in pairs.

2.2 Problem Statement and Motivation

Many feature selection algorithms eliminate redundant or weakly relevant features from training datasets (Blum and Langley (1997); Appice et al. (2004)). Studies consider presence of duplicate data or irrelevant features in training datasets results in poor performance of predictive models. However scoring a feature individually and independent of each other will not grantee best subset selection. Feature could be redundant but useful in presence of other features, or could be useless by themselves but useful in presence of others. We present a series of small examples that outline the importance of including redundant or irrelevant features in a subset selection.

2.2.1 Apparently Redundant Variables

The concept of feature redundancy is usually defined using feature correlation (Guyon and Elisseeff (2003)). It is widely accepted that two features might be considered redundant if their values are highly correlated. Many feature ranking algorithms avoid including redundant features in the selected subset. It has been believed that redundant features add no relevant information to other features because they are correlated or because they can be obtained by combination of other features. However considering a feature by itself independent of others would not be always wise. We constructed an example to clarify that considering such apparently redundant features in subset selection could possibly add information to output estimation in presence of feature interactions.

Three methods that are well-known in the machine learning literature, FCBF, ReliefF, and Random Forest are considered to select feature subset. One of the well-known variable ranking algorithm is Fast Correlation Based Filter (FCBF). FCBF uses relevant analysis to determine strongly related features by removing irrelevant and weakly relevant features and implements redundancy analysis to eliminate redundant features from strongly relevant ones [16]. ReliefF was originally designed for application to binary classification problems with discrete or numerical features. RReliefF is an extension of ReliefF algorithm designed for regression problems that same as Relief calculates a feature score for each feature which can then be applied to rank and select top scoring features for feature selection (Robnik-Šikonja and Kononenko (1997)). The last considered method is Random Forest, which builds multiple decision trees and merge them together to get a more accurate and stable prediction (Liaw et al. (2002)). One big advantage of Random Forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Compared to FCBF, ReliefF and Random Forest are considered sensitive to feature interaction. We investigate how these three methods would select the best subset of features in the presence of redundancy and feature interactions.

Consider the binary matrix A_1 with 7 observations and 4 features and a response variable Y_2 generated using the formula $Y_1 = 37x_1 + 97x_2 + 60x_3x_4$. Feature-response correlation, considers features 2, 3, and 4 as significant features and feature-feature correlation determines pairs (1,3) and (3,4) as redundant features.

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}, Y_1 = \begin{bmatrix} 97 \\ 134 \\ 97 \\ 37 \\ 97 \\ 194 \\ 97 \end{bmatrix}$$

Implementing FCBF algorithm with threshold 0.4 on the generated data set determines the selected subset to include only features 2 and 3. We also considered ReliefF algorithm as a feature selection method that is notably sensitive to feature interactions (Kira and Rendell (1992)). ReliefF scores features 1, 2, 3, and 4 in order as 0.64, 0.31, 0.31, 0.33. Implementing Random Forest on matrix A_1 and response variable Y_1, gives us feature scores in order from 1 to 4 as 0.19, 0.35, 0.34, 0.09.

Considering highly correlated features with response and the significant interaction effect, the best subset should be considered as {2, 3, 4}. As we saw FCBF never considered features 4 due to the apparent redundancy. Relief and Random Forest also assigned feature 4 a low score which exclude feature 4 from the optimal subset while its interaction with feature 3 is highly significant with response Y.

2.2.2 Apparently Irrelevant Variables

Irrelevant features usually are specified by feature-response correlation. Here we came up with an example that shows removal of irrelevant features does not always give us the optimal subset. Consider the matrix A_2 with the following response variable, Y_2, generated using formula $Y_2 = 37x_1 + 97x_2 + 60x_1x_2$:

$$A_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, Y_2 = \begin{bmatrix} 97 \\ 194 \\ 194 \\ 37 \\ 97 \\ 194 \\ 37 \end{bmatrix}$$

Feature-response correlation considers features 2 as a highly correlated feature to the response variable and features 1 and 3 as irrelevant ones. Feature-feature correlation determines the existence of no redundant feature while we observe a significant interaction between features 1 and 2. Starting with FCBF algorithm and considering the threshold of 0.4, the best selected subset is $\{1\}$. Fitting ReliefF algorithm on the dataset scores features in order from 1 to 3 as 0.6, 0.6, 0.11. Random Forest also scores features as 0.4, 0.56, 0.036. As it is observable both RReliefF and Random Forest give features 1 and 2 higher scores while FCBF did not included feature 2 which is important due to interaction effect.

2.3 Feature Selection for Continuous Response Variables with Interaction (FSCI) - An Algorithm

Feature selection algorithms are the combination of search techniques for proposing a new feature subset to describe the response concept. The best subset usually consists of the features with big main effect and possibly features with interaction effects. Considering the literature, most well-known feature selection algorithms could determine the best subset of features that have a big main effect on the response, however, we are not aware of any feature selection algorithm that could explicitly try to identify the interaction features in a dataset. Here, we propose a new feature selection algorithm (FSCI) for multiple linear regression models with binary features. FSCI can specifically determine the features responsible for interaction effects along with ones having a big main effect on the response. A subset of features with big main effect on the response variable could be detected using well-known feature selection algorithms, Section 2.3.1. To select a subset of features that are interacting with each other we use a Mixed Integer Linear Programming model introduced in Section 2.3.2. In Section 2.3.3 we will discuss the steps of the algorithm in details.

2.3.1 Selecting Features with Significant Main Effect

This step of the FSCI algorithm associates with finding a subset of features having a significant main effect on the response variable. Based on literature any well-known filter or wrapper method could deliver the subset of significant features. A wrapper uses the intended learning algorithm itself to evaluate the usefulness of features, while a filter estimate the goodness of each feature using training datasets and are independent of any learning algorithms. When dealing with huge number of features, usually filter methods are prioritize due to their computational efficiency. On the other hand, the wrapper approach is generally considered to produce better feature subsets but runs much more slowly than a filter.

2.3.2 Selecting Features with Interaction Effect

To detect feature interaction, we considered an optimization method Wang and Mehr (2019) developed for detecting what is called an epistatic effect, which refers to the interaction of genes (Xu and Jia (2007b); Cordell (2002b)). They modified linear regression model by defining a new term added to a multiple linear regression model, which captures interaction effect. For completeness we briefly discuss this model represented in Equation (3.40), but refer the reader to Wang and Mehr (2019) for full details.

$$y_i \quad = \quad \beta_0 + \sum_{j=1}^{n} X_{ij}\beta_j + b \cdot I\left(\sum_{j=1}^{n} X_{ij}(\lambda_j - \mu_j) = \sum_{j=1}^{n} \lambda_j\right) + \epsilon_i, \forall i \in \{1, ..., m\}. \tag{2.1}$$

Here m denotes the number of observation and n specifies the number of features. $X_{i,j}$ represents the feature j for observation i, which is assumed to be binary and y_i corresponds to the response variable for observation i. Other notations are described as follow:

- β_0: the intercept effect for any i with $X_{i,j} = 0, \forall j \in \{1, ..., n\}$
- β_j: the partial regression coefficient, which is the differential effect of $X_{i,j} = 1$ over $X_{i,j} = 0$ for any i
- b: the magnitude of the interaction effect
- $I(\cdot)$: the indicator function, which is equal to 1 if the statement in the parentheses is true and 0 otherwise

- (λ_j, μ_j): binary variables that define the interaction, which is triggered if and only if $X_{i,j} = 1, \forall j \in \{j : \lambda_j = 1\}$ and $X_{i,j} = 0, \forall j \in \{j : \mu_j = 1\}$, for any i
- ϵ_i: the random error term for observation i

Correctly inferring λ_j and μ_j for all $j \in \{1, ..., n\}$ will lead to detection of interaction features. To solve for λ_j and μ_j, they developed a nonlinear non-convex optimization problem. However, their model can detect only one interaction effect. To consider multiple interactions, we enhanced their modeling by introducing a new index k, which represents the number of interaction effects that is observable in a dataset. The mathematical formulation of our optimization model is presented as follows.

$$\min_{\beta_0,b,\beta,\lambda,\mu,z} \quad \sum_{i=1}^{m} |y_i - (\beta_0 + \sum_{j=1}^{n} X_{ij}\beta_j + b^k z_i^k)| \tag{2.2}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} X_{ij}(\lambda_j^k - \mu_j^k) \geq -M(1 - z_i^k) + \sum_{j=1}^{n} \lambda_j^k \quad \forall i \in \{1, ..., m\}, \forall k \in \{1, ..., K\} \tag{2.3}$$

$$\sum_{j=1}^{n} X_{ij}(\lambda_j^k - \mu_j^k) \leq \sum_{j=1}^{n} \lambda_j^k - 1 + M z_i^k \quad \forall i \in \{1, ..., m\}, \forall k \in \{1, ..., K\} \tag{2.4}$$

$$\lambda_j^k + \mu_j^k \leq 1 \quad \forall j \in \{1, ..., n\}, \forall k \in \{1, ..., K\} \tag{2.5}$$

$$z_i^k, \lambda_j^k, \mu_j^k \in \{0, 1\} \quad \forall i \in \{1, ..., m\}, \forall j \in \{1, ..., n\} \\ , \forall k \in \{1, ..., K\} \tag{2.6}$$

$$b^k, w_i^k, \lambda_j^k \text{ free} \quad \forall i \in \{1, ..., m\}, j \in \{0, 1, ..., n\} \\ , \forall k \in \{1, ..., K\} \tag{2.7}$$

Here, the objective function (2.2) is minimizing the absolute errors between prediction and actual observation. Constraints (3.75) and (2.4) describe the interaction effects. The M parameter represents a sufficiently finite number, and the decision variable z_i^k indicates whether ($z_i^k = 1$) or not ($z_i^k = 0$) observation i has the k^{th} interaction effect. Constraint 2.5 means that the k^{th} interaction effect cannot require the presence ($\lambda_j^k = 1$) and absence ($\mu_j^k = 1$) of a variable at the same time. Constraints 2.6 and 2.7 define the appropriate types of the decision variables.

In this model, they minimize the absolute errors, as opposed to the commonly used root mean square error in linear regression models (Armstrong and Collopy (1992)), because it can be linearized and is more computationally tractable. Using linearizion techniques, they defined an equivalent mixed integer linear program (MILP) of the model which is used to detect interacting features (Wang and Mehr (2019)).

2.3.3 Proposed Feature Selection Algorithm

The FSCI algorithm has 3 steps. Step 0 initializes the parameters, including vector $\mathcal{SU}$, which represents a subset of features that have a big main effect on the response. As this has been studies in literature, we assume we can generate this subset by implementing any well-known variable selection method and consider the subset as the input of the algorithm. Step 1 is the main contribution of this paper. Here we try to find the features that are responsible for interaction effect. After initializing k, number of interaction effects, and determining the lower bound and the upper bound of interaction effects, we solve model $\mathcal{L}(X, y)$ on observations and detect interacting features. Interacting features will be stored in subset $\mathcal{IN}$ and the union of the two subsets $\mathcal{SU}$ and $\mathcal{IN}$ represents the final subset in Step 2.

Step 0: Take the full set of features $X \in \mathbb{B}^{m \times n}$ and the response variable $y \in \mathbb{R}^{m \times 1}$. Consider a set $\mathcal{SU}$ such that $\mathcal{SU} \subseteq \{1, 2, ..., n\}$, represents a set of features having a big main effect on the response variable.

Step 1: Select features responsible for interaction effect.

> **Step 1-A:** Initialize k, which is the number of interaction effects. Determine $\underline{b}^k$ and $\overline{b}^k$ and go to Step 1-B.
>
> **Step 1-B:** Solve model $\mathcal{L}(X, y)$. For all $j \in \{1, ..., n\}$ identify a subset $\mathcal{IN}$ such that $\mathcal{IN} \subseteq \{1, 2, ..., n\}$ and $\lambda_j^k = 1$ or $\mu_j^k = 1$. Go to Step 2.

Step 2: Determine $\mathcal{SU} \cup \mathcal{IN}$ as the final subset.

Remark: Our knowledge about a dataset could affect a right choice of k. However limitation like time and computational difficulties could affect our choice. As a result, a balance between number of interaction effects and time could be the best solution. RMSE and prediction precision could be other criteria for determining the right number of k. We can always start with a small number for k and record how RMSE changes while increasing the k. Based on the improvement, the right choice could be made.

2.4 Experiments

In this section, we first investigate the performance of our algorithm on the synthetic data and focus on the study of the impact of interaction effect on linear regression and feature selection performance. Then we conduct an experiment on real datasets found at different journals and UCI Machine Learning Repository. The purpose of the experiments is to evaluate the effectiveness of FSCI versus bench marks, highlight strengths and limitations, and provide insights into some implementation issues. In our experiments, we choose three representative feature selection algorithms (FCBF, ReliefF, and Random Forest) in comparison with FSCI algorithm. All the feature selection methods are implemented in Python sklearn package (Pedregosa et al. (2011)). Pyomo/Cplex was use to solve the model $\mathcal{L}(X, y)$. The experiments are run on a computer with 2.5 GHz Intel Core i5, 4 GB memory.

2.4.1 Simulation Experiment

In our experiment, we generated a synthetic binary datasets with $n = 9$ features and $m = 100$ observations such that the first 8 features were sampled from a random binary uniform distribution and feature X_9 was generated as a multiplication of features 2 and 5, $X_9 = X_2X_5$, which brings redundancy to our dataset.

We distributed 9 features into two groups such that features 1, 2 and 3 , were set to have the effect of β_1^j and features 4, 5, 6, 7, 8 and 9 were selected to have an effect of β_2^j on the response variable. We assumed there is an interaction effects of γ between features 7, 8 and 9. Parameters β_0 and β_2^j were generated from a uniformly distributed random integers between 0 and 25. To consider features 5 to 9 as irrelevant features, we generated β_1^j with bigger magnitudes with respect to β_2^j based on Table 2.1 and also considered different values for γ to see how changes in magnitude of β_1^j with respect to γ would affect subset selection. Intuitively values of these parameters could affect the effectiveness of feature selection methods. The distribution of β_1^j and γ are described in Table 2.1 and the response variable was generated using Model (2.8). Here we try to design different scenarios considering values of these two parameters and see how FCBF, Relief, Random Forest and FSCI would differentiate.

$$y_i \quad = \quad \beta_0 + \sum_{j=1}^{3} \beta_1^j X_{ij} + \sum_{j=4}^{9} \beta_2^j X_{ij} + \gamma X_7 X_8 X_9, \forall i \in \{1, ..., m\} \tag{2.8}$$

Table 2.1: Parameters used for simulation

Parameters	Levels
β_1^j	$\mathcal{U}(75, 100)$, $\mathcal{U}(175, 200)$, $\mathcal{U}(275, 300)$
γ	$\mathcal{U}(75, 100)$, $\mathcal{U}(175, 200)$, $\mathcal{U}(275, 300)$, $\mathcal{U}(375, 400)$, $\mathcal{U}(475, 500)$

Considering Table 2.1, we grouped our simulation studies in three categories: (1) β_1^j and γ in a same level (main effect and interaction effect are in a same level, defined as S/S), (2) β_1^j bigger than γ (main effect is bigger than the interaction effect, defined as B/L) and (3) γ bigger than β_1^j (main effect is smaller than the interaction effect , defined as L/B). For the S/S category, we considered β_1^j and γ generated from $\mathcal{U}(75, 100)$. β_1^j generated from $\mathcal{U}(75, 100)$ with all possible values of γ in Table 2.1 except $\mathcal{U}(75, 100)$ are categorized in the L/B group and γ generated from a $\mathcal{U}(75, 100)$ with β_1^j generated from $\mathcal{U}(175, 200)$ and $\mathcal{U}(275, 300)$ were considered in the B/L category. We implemented

FCBF, ReliefF, Random Forest, and FSCI on all generated scenarios in three groups, each with 100 independent repetitions, so that the total number of simulation runs was 700. The considered threshold for FCBF was set to be 0.35 and 75% of data was used to train the models and 25% was used to test. In FSCI method, the lower bound and upper bound of the magnitude of interaction effect ($\underline{b}^k$ and $\bar{b}^k$) were set to be 0.5 and 1.5 times of the true value and FCBF method was used in Step 0 to generate set $\mathcal{SU}$.

Table 2.2: Number of times interaction effect was captured

Main Effect, Interaction Effect	FCBF	ReliefF	Random Forest	FSCI
$\mathcal{U}(275, 300), \mathcal{U}(75, 100)$	0	0	0	96
$\mathcal{U}(175, 200), \mathcal{U}(75, 100)$	0	0	0	98
$\mathcal{U}(75, 100), \mathcal{U}(75, 100)$	0	0	0	98
$\mathcal{U}(75, 100), \mathcal{U}(175, 200)$	0	1	5	98
$\mathcal{U}(75, 100), \mathcal{U}(275, 300)$	0	1	38	99
$\mathcal{U}(75, 100), \mathcal{U}(375, 400)$	0	0	58	96
$\mathcal{U}(75, 100), \mathcal{U}(475, 500)$	2	2	64	99

We recorded the number of times, each feature selection method was successful in detecting interaction effect for each (Main Effect, Interaction Effect) combination in Table 2.2. Due to redundancy between features, interacting features could be the set of {7, 8, 9} or {2, 5, 7, 8}. Based on results, it seems in B/L and S/S categories (when interaction effect is as low as main effect or lower than that), all methods FCBF, ReliefF, and Random Forest act poorly in detecting the interaction effect, while FSCI performs well in all simulation runs. The deficiency we observe in FSCI result from 100 is due to dependency between features that is unavoidable while generating simulation experiments. Interestingly when interaction effect is bigger than the main effect (group L/B), Random Forest is capable to include interacting features in its selected subset and as the magnitude of interaction effect with response to the main effect increases, Random Forest performs better.

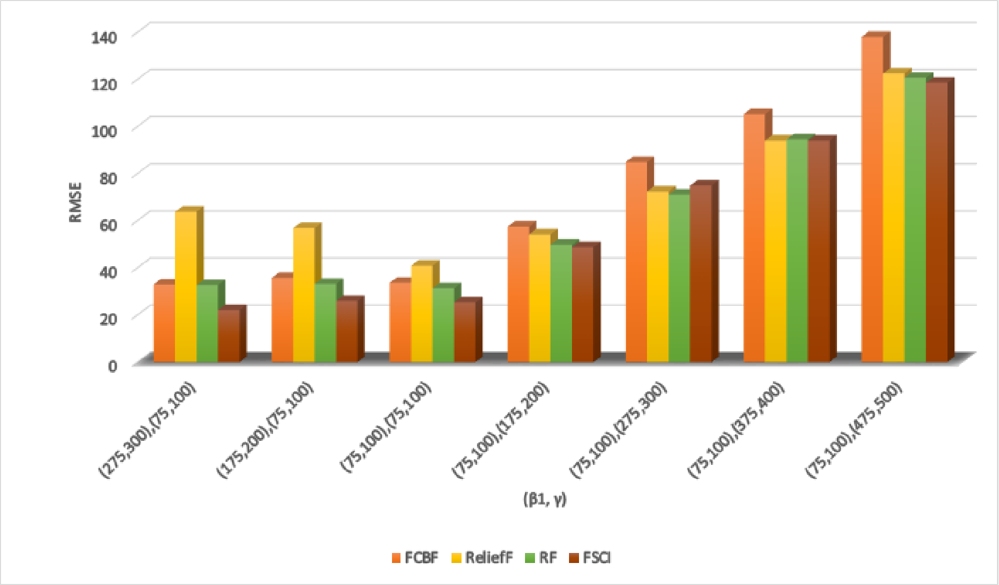
We also recorded the average number of times each feature was selected in each category for all feature selection methods in Table 2.3. In B/L and S/S categories, all methods mostly selected features with big main effects and rarely selected interacting features. However, for the L/B group we observe an improvement on the number of times each feature selection method detected interacting features. Interestingly Random Forest seems to provide more information regarding interacting features when interaction effect is bigger than the main effect. However, it is still far from FSCI performance. Looking at both Tables 2.2 and 2.3, we observe a huge improvement on interaction effect using FSCI. This method not only acts well when interaction effect is bigger than the main effect, it also performs excellently in scenarios when main effect and interaction effect are in same level or main effect is bigger than the interaction effect. An other advantage of FSCI is the ability of the algorithm in detecting the exact features responsible for interaction effect. None of the methods we could find in literature was able to exactly define the combination of features correspond to interaction effect while this information could provide valuable insight for industries to forecast and predict their processes.

Table 2.3: Average number of times each feature was selected in each category

Category	Methods	F1	F2	F3	F4	F5	F6	F7	F8	F9
B/L	FCBF	100	100	99	0	0.5	0	0	1	2
	ReliefF	91	99.5	90	0.5	0	0.5	1	0	30
	rest	100	100	100	0	0	0	0	0	2
	FSCI	100	100	99	1	32.5	3	99	99	73.5
S/S	FCBF	94	100	96	0	3	1	3	5	22
	ReliefF	80	96	78	0	5	3	2	6	73
	Random Forest	100	100	100	2	0	0	0	0	24
	FSCI	94	100	96	3	34	3	100	99	83
L/B	FCBF	55.25	100	51	0	11.25	0.25	6.75	5.75	37.5
	ReliefF	87	98	88	2	18.75	3	10.25	10.5	95.5
	Random Forest	76.75	72.5	74.25	0	0	1.5	58	60.25	83
	FSCI	55.75	100	52.25	1.75	38	3.5	99.5	98.75	91.75

We also investigated how Root Mean Square Error (RMSE) would change while implementing FCBF, ReliefF, Random Forest and FSCI. Figure (2.1) shows RMSE for all combination of β_1^j and γ. In general including interacting features decreases the RMSE. In B/L and S/S categories, ReliefF performs badly while FSCI is the best. Also, in L/B category generally we observe better RMSE for FSCI method compare to others except for the (Main Effect, Interaction Effect) : ($\mathcal{U}(75, 100)$, $\mathcal{U}(275, 300)$) combination. For this scenario Random Forest is better that FSCI method. However, it should be mentioned that FSCI considered FCBF to find the subset of features that have a big main effect on the response variable in Step 0 of the algorithm. FSCI is building on FCBF, and considering FCBF performance for this combination, we still observe a huge improvement in RMSE. The choice of Random Forest to generate set $\mathcal{SU}$ in Step 0 of the algorithm, would intuitively improve observed RMSE for all categories.

Figure 2.1: RMSE for all scenarios considering different values of main and interaction effect, using random integer uniform distribution



2.4.2 Experiment on Real-world Data

In this section, we empirically evaluate the performance of our proposed algorithm, and present the experimental result compared with the other three mentioned feature selection algorithms on real datasets. Beyond a simple evaluation of effectiveness, our primary purpose here is to provide some insights into two issues that are critical for practical implementation of our algorithm: (a) setting the bounds in the optimization problem, and (b) dealing with non-binary variables. First we consider implementation of the algorithm on categorical datasets. An extension of its application also is discussed on a continuous dataset that is transformed in to a categorical dataset having different levels for each feature.

2.4.2.1 Categorical Predictor Variables

The direct application of the proposed method is for datasets with a continuous response but categorical predictor variables. Such data sets are common in certain application areas such as factorial design experiments but we are not aware of a large set of standard test dataset that fit this criteria. While extensions to datasets with continuous and mixed predictor variables is possible, we start with a smallest of test datasets that fit the criteria. Here we considered 4 categorical datasets found on literature as follow: (1) Pulp Paper Bleaching Process dataset (PPBP): investigated the treatment of bleachingâeffluent from pulp and paper industry in a batch electro-coagulation (EC) reactor using aluminum as sacrificial electrodes (Sridhar et al. (2012)), (2) honeycomb dataset:

studies the effect of angle difference between successive prepreg layers, number of prepreg layers, and number of adhesive layers on the low-velocity impact of the honeycomb sandwich structures (Butukuri et al. (2012)), (3) concussion dataset: compared sex differences in the incidence of concussions among collegiate athletes in men's and women's soccer, lacrosse, basketball, softball/baseball, and gymnastics (Covassin et al. (2003)), and (4) Sodium Bisphosonate dataset (SB): evaluated the effect of polymer type and amount, drug amount and internal aqueous phase volume ratio on entrapment efficiency of bisphosphonate (risedronate sodium RS) (Nasr et al. (2011)). To find the features with big main effects and ones included in interaction effect, we converted categorical features in to a binary data set by introducing dummy variables. Summary of datasets before and after transformation are described in Table 2.4.

Table 2.4: Categorical data sets specifications

Dataset	Number of features	Number of binary features	Number of Observations
Pulp Paper Bleaching Process (PPBP)	4	12	29
honeycomb	3	54	9
concussion	3	10	30
Sodium Bisphosonate (SB)	4	8	48

2.4.2.2 Numerical Results of Categorical Predictor Variables

The performance of FSCI algorithm depends on the values of the lower bound and upper bound of the magnitude of the interaction effect, b^k. Usually knowledge of data sets helps to estimate proper values for these parameters. However, for these data sets we do not have clear understanding about the possible lower bound and upper bound of b^k. As a result, we developed a new heuristic method to estimate these values. The heuristic algorithm is built on stepwise linear regression method. First we find the best fitted model on the binary data set using stepwise linear regression method. Considering the selected features in stepwise linear regression model, we implement an exhaustive two-way interaction search. The coefficient of two-way interaction effects are recorded and the minimum and maximum values of these coefficients are considered as possible lower bounds and upper bounds of the magnitude of interaction effects in each data set. We conducted a sensitivity analysis of the range of the magnitude of the interaction effects that could affect the effectiveness of FSCI algorithm which are summarized in Table 2.5. The lower bound and upper bound of b^k are multiplied with ϕ. When $\phi = 1$, we have the minimum and maximum values of coefficients of exhaustive two-way interaction search in stepwise linear regression model. Other values of these parameters were chosen to explore how tightening or widening the range of b^k, would affect our selection of interacting features.

Table 2.5: Parameters for the sensitivity analysis for categorical data sets

ϕ	$\underline{b}^k_{PPBP}$	$\overline{b}^k_{PPBP}$	$\underline{b}^k_{honeycomb}$	$\overline{b}^k_{honeycomb}$	$\underline{b}^k_{concussion}$	$\overline{b}^k_{concussion}$	$\underline{b}^k_{SB}$	$\overline{b}^k_{SB}$
0.5	-6.15	16	-0.171	-0.094	-10.5	0	-9.86	17.33
1	-12.3	32	-0.342	-0.189	-21	0	-19.72	34.66
1.5	-18.45	48	-0.512	-0.283	-31.5	0	-29.58	51.99
2	-24.6	64	-0.683	0.377	-42	0	-39.44	69.33
2.5	-30.75	80	-0.854	0.471	-52.5	0	-49.3	86.66

To select the best values of $\underline{b}^k$ and $\overline{b}^k$, we ran FSCI considering all possible combinations of $\underline{b}^k$ and $\overline{b}^k$. The combination giving us the smallest RMSE, defines our selected values of $\underline{b}^k$ and $\overline{b}^k$. If multiple combinations lead to same RMSE, the tighter combination is considered. Data sets are partitioned as training (75%) and testing (25%). The possibility of having two possible interaction effect is investigated. FSCI is built on FCBF with threshold 0.1, and is compared with FCBF, Relief, and random forest in term of its effectiveness. Table 2.6 shows the desirable value of ϕ for each dataset along with number of selected features for all methods. As result shows, exhaustive two-way interaction provides desirable values for lower bound and upper bound of interaction effect. Indeed, FSCI is not sensitive to tightening or widening the range of b^k for these data sets. To measure how including interacting features improves our prediction, we calculate RMSE for FCBF, ReliefF, random forest, and FSCI. As Table 2.7 shows, for all data sets FSCI provides the lowest RMSE compare to others. The greatest reduction in RMSE is observed in Pulp Paper Bleaching Process dataset (PPBP) with %61.7 improvement compare to random forest and ReliefF, and %55.4 improvement compare to FCBF. However, honeycomb data set seems less sensitive to include interaction effect in the subset.

Table 2.6: Number of selected features for all methods

Data Set	ϕ	FCBF	FSCI	ReliefF	Random Forest
PPBP	1	5	9	2	2
honeycomb	0.5	3	5	3	2
concussion	1	2	5	2	2
SB	1	3	4	4	2

Table 2.7: RMSE for all methods

Data Set	FCBF	FSCI	ReliefF	Random Forest
PPBP	16.37	7.29	19.04	19.04
honeycomb	0.311	0.177	0.18	0.189
concussion	9.93	6.67	7.04	7.04
SB	13.77	13.14	15.51	15.51

2.4.2.3 Continuous Predictor Variables

To verify the effectiveness of our method on continuous datasets, stock portfolio performance data set from UCI machine learning repository is employed (Liu and Yeh (2017)). This data set is gathered in 4 different periods from September 1990 to June 2010. The 4th period from September 2005 to June 2010 is considered in this study. The data set has 6 features: (1) the weight of the Large B/P concept, (2) the weight of the Large ROE concept, (3) the weight of the Large S/P concept, (4) the weight of the Large Return Rate in the last quarter concept, (5) the weight of the Large Market Value concept (6) the weight of the Small systematic Risk concept. Interestingly in this study Liu and Yeh categorized all features into 7 levels described in Table 2.8. Our selective response is annual return rates. After transforming features in to binary, we have the new data set with 63 observations and 42 features. We look for two possible interaction effects in this data set. FSCI is compared with three represented feature selection algorithms.

Table 2.8: Stock portfolio performance data set

B/P	ROE	S/P	Quarterly Return(QR)	Market Value(MV)	Small Risk(SR)
0	0	0	0	0	0
0.167	0.167	0.167	0.167	0.167	0.167
0.2	0.2	0.2	0.2	0.2	0.2
0.25	0.25	0.25	0.25	0.25	0.25
0.333	0.333	0.333	0.333	0.333	0.333
0.5	0.5	0.5	0.5	0.5	0.5
1	1	1	1	1	1

2.4.2.4 Numerical Results of Continuous Predictor Variables

As mentioned before, different values of of the lower bound and upper bound of the magnitude of the interaction effect, b^k affect FSCI effectiveness. To find the desirable $\underline{b}^k$ and $\overline{b}^k$, sensitivity analysis for FSCI on the binary stock portfolio performance data set is implemented. The selected feature subsets for different range of b^k can be seen in Table 2.10. As result shows, different values of lower bound and upper bound of the magnitude of interaction effect, contributed to different selection of interacting features. To verify the best values of $\underline{b}^k$ and $\bar{b}^k$ and corresponding the best subset of interacting features, we fitted the linear regression model on the original stock portfolio performance data set. Then added the two interaction effects found with FSCI in the linear regression model. Note that interaction effects are determined by features without consideration of their level. If interaction effects reduce the RMSE of the linear regression model, the subset of interacting feature is recognized correctly and the estimation of the values of $\underline{b}^k$ and $\bar{b}^k$ is desirable. The linear regression model without consideration of interacting features has a RMSE equal to 0.0696. The RMSE of linear regression model with interacting features for all scenarios also is shown in Table 2.10. Based on RMSE values of Table 2.10, we can assure that the values of $\underline{b}^k$ and $\bar{b}^k$, when ϕ equals to 0.5, 1 or 2.5 are undesirable as they found interacting features that their consideration in the linear regression model added no valuable information to the prediction modeling. However, in scenarios where ϕ is 1.5 and 2 we observe improvements on RMSE of the linear regression model. This makes these scenarios to be desirable in generating $\underline{b}^k$ and $\bar{b}^k$. Comparison between their RMSE and computation time, also lead us to consider $\phi = 2$ as the best scenario and select $\underline{b}^k = -0.549$ and $\bar{b}^k = 1.442$.

Table 2.9: Parameters for the sensitivity analysis

ϕ	$\underline{b}^k$	$\bar{b}^k$
0.5	-0.137	0.355
1	-0.275	0.711
1.5	-0.412	1.066
2	-0.549	1.422
2.5	$-.687$	1.777

Now that the best values of $\underline{b}^k$ and $\bar{b}^k$ are determined, we run FCBF, ReliefF, and Random forest on the binary stock portfolio performance data set. We analyze the performance of these methods and compare their accuracy in prediction analysis with FSCI algorithm. Here also, the data set is partitioned as training (75%) and testing (25%). Selected features and RMSE of these methods are shown in Table 2.11. The value of RMSE shows FSCI algorithm outperforms FCBF algorithms up to 24.46% and Random Forest and ReliefF algorithms up to 16.59%. FSCI explicitly determines the complexity of interaction effects, their exact combination, and also the exact levels of the features that lead to interaction effect. As we see none of the considered method can detect complexity, the exact combination and the specific level of features responsible for interaction effect. Interestingly we observed that the result of the FSCI on binary stock portfolio performance data

Table 2.10: Sensitivity analysis on the values of $\underline{b}^k$ and $\overline{b}^k$ in FSCI

	FSCI			
ϕ	1^{st} Interaction Effect	2^{st} Interaction Effect	RMSE(linear regression)	Computation time(seconds)
0.5	$B/P = 0.2$	$B/P = 0$	0.0766	2785.46
	$ROE = 0.333$	$ROE = 1$		
	$X_4 = 0.333$	$S/P = 0$		
	$MV = 0$	$QR = 0$		
	$SR = 0$	$SR = 0.167$		
1	$B/P = 0$	$B/P = 1$	0.0766	3629.92
	$ROE = 1$	$ROE = 0.333$		
	$S/P = 0$	$QR = 0.333$		
	$QR = 0$	$MV = 0$		
	$SR = 1$	$SR = 0$		
1.5	$B/P = 0.167$	$B/P = 0$	0.0652	4061.58
	$ROE = 0.333$	$S/P = 0$		
	$S/P = 1$	$QR = 0$		
	$QR = 0.333$	$MV = 0.167$		
	$MV = 0$			
	$SR = 0$			
2	$B/P = 0$	$B/P = 0.2$	0.0639	3572.78
	$S/P = 0$	$ROE = 0.333$		
	$QR = 0$	$QR = 0.333$		
		$MV = 0$		
		$SR = 0$		
2.5	$B/P = 0$	$ROE = 0.333$	0.0787	4783.79
	$S/P = 0$	$S/P = 0.2$		
	$QR = 0$	$QR = 0.333$		
		$MV = 0$		
		$SR = 0$		

set, can be transfered to linear regression model and leads to a better prediction model with smaller RMSE. Indeed, including two interaction effects (B/P, S/P, QR) and (B/P, ROE, QR, MV, SR) reduces the RMSE of the linear regression model up to 8.14%.

It should be mentioned that here FSCI generates set $\mathcal{SU}$ in Step 0 of the algorithm based on FCBF algorithm. Therefore, the observed improvements in RMSE for both binary version of stock portfolio performance data set and the original data set is the lowest possible improvement we could achieve. If we would consider Random Forest or ReliefF in generating subset of features with big main effect, we were building FSCI in its best case scenario and would observe better improvements on RMSE.

Table 2.11: Comparison of feature selection methods on the binary stock portfolio performance data set

		FSCI			
ϕ	FCBF	1^{st} Interaction Effect	2^{st} Interaction Effect	ReliefF	Random Forest
2	$S/P = 0.5$	$B/P = 0$	$B/P = 0.2$	$S/P = 0$	$S/P = 0$
	$QR = 0.2$, 0.25, 0.333	$S/P = 0$	$ROE = 0.333$	$SR = 0$	$SR = 0$
	$SR = 0$	$QR = 0$	$QR = 0.333$		
			$MV = 0$		
			$SR = 0$		
RMSE	0.1018	0.0769		0.0922	0.0922

2.5 Conclusions and Future Work

When used with predictive models, the main goal of feature selection is to find a feature subset that has highly prediction accuracy. Well-known feature selection methods, remove redundant and apparently irrelevant features to avoid poor accuracy of predictive models. Here we proposed a new feature selection algorithm, FSCI, that emphasizes on the importance of considering redundant and apparently irrelevant features in subset selection. Redundant and apparently irrelevant features could associate in feature interaction, which exists in many applications. FSCI, is built on three steps. First it detects a subset of features having a big main effect. Then, it finds a subset of feature correspond to interaction effect. The Union of these two subset will define the final selected features. Our sensitivity analysis revealed that this algorithm is sensitive to the values of the lower bound and the upper bound of the magnitude of interaction effect. Usually knowledge of the process and data help to have a good estimation of these values. For data sets and processes we lack information about these possible values, we developed a heuristic algorithm based on stepwise linear regression model that implements exhaustive search on all possible two- way interactions.

The effectiveness of the FSCI algorithm is evaluated on the synthetics and real world data sets. Both categorical and continuous data sets are considered as case studies. The proposed method is compared with three other feature selection methods (FCBF, ReliefF, and Random Forest) in terms of ability to capture feature interaction and prediction accuracy. The experimental results of synthetic and real world data sets show that FSCI can effectively identify complexity of interaction effect and their exact combination. Interestingly based on literature no feature selection method could detect explicitly these features and their combination. We also determined FSCI superior to other three feature selection methods in prediction accuracy.

As a caveat, the proposed algorithm had several limitations. For example, the validity of detected interaction effect depends on two assumptions: (1) the lower and upper bound of the magnitude of interaction effect, and (2) number of interaction effects. The algorithm also is developed for categorical datasets with continuous response variable, which limits its application. A potentially fruitful direction of the future research is to extend the proposed method for classification data sets.

2.6 References

Appice, A., Ceci, M., Rawles, S., and Flach, P. (2004). Redundant feature elimination for multi-class problems. In *Proceedings of the twenty-first international conference on Machine learning*, page 5. ACM.

Armstrong, J. S. and Collopy, F. (1992). Error measures for generalizing about forecasting methods: Empirical comparisons. *International journal of forecasting*, 8(1):69–80.

Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271.

Bradley, P. S., Mangasarian, O. L., and Street, W. N. (1998). Feature selection via mathematical programming. *INFORMS Journal on Computing*, 10(2):209–217.

Butukuri, R. R., Bheemreddy, V., Chandrashekhara, K., and Samaranayake, V. (2012). Evaluation of low-velocity impact response of honeycomb sandwich structures using factorial-based design of experiments. *Journal of Sandwich Structures & Materials*, 14(3):339–361.

Cordell, H. J. (2002). Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Human molecular genetics*, 11(20):2463–2468.

Cotter, S. F., Kreutz-Delgado, K., and Rao, B. D. (2001). Backward sequential elimination for sparse vector subset selection. *Signal Processing*, 81(9):1849–1864.

Covassin, T., Swanik, C. B., and Sachs, M. L. (2003). Sex differences and the incidence of concussions among collegiate athletes. *Journal of athletic training*, 38(3):238.

Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. In *Icml*, volume 1, pages 74–81.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, 1(1-4):131–156.

Frohlich, H., Chapelle, O., and Scholkopf, B. (2003). Feature selection for support vector machines by means of genetic algorithm. In *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 142–148. IEEE.

Ghaemi, M. and Feizi-Derakhshi, M.-R. (2016). Feature selection using forest optimization algorithm. *Pattern Recognition*, 60:121–129.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Jakulin, A. and Bratko, I. (2003). Analyzing attribute dependencies. In *European conference on principles of data mining and knowledge discovery*, pages 229–240. Springer.

John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier.

Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In *Machine Learning Proceedings 1992*, pages 249–256. Elsevier.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324.

Koller, D. and Sahami, M. (1996). Toward optimal feature selection. Technical report, Stanford InfoLab.

Li, Y. and Liu, Y. (2008). A wrapper feature selection method based on simulated annealing algorithm for prostate protein mass spectrometry data. In *2008 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 195–200. IEEE.

Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

Liu, Y.-C. and Yeh, I.-C. (2017). Using mixture design and neural networks to build stock selection decision support systems. *Neural Computing and Applications*, 28(3):521–535.

Mo, D. and Lai, Z. (2019). Robust jointly sparse regression with generalized orthogonal learning for image feature selection. *Pattern Recognition*, 93:164–178.

Nasr, M., Awad, G. A., Mansour, S., Al Shamy, A., and Mortada, N. D. (2011). A reliable predictive factorial model for entrapment optimization of a sodium bisphosphonate into biodegradable microspheres. *Journal of pharmaceutical sciences*, 100(2):612–621.

Olafsson, S., Li, X., and Wu, S. (2008). Operations research and data mining. *European Journal of Operational Research*, 187(3):1429–1448.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830.

Radha, R. and Muralidhara, S. (2016). Removal of redundant and irrelevant data from training datasets using speedy feature selection method. *Intâl J Comp. Sci. and Mob. Comput.*, 5(7):359–364.

Robnik-Šikonja, M. and Kononenko, I. (1997). An adaptation of relief for attribute estimation in regression. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICMLâ97)*, volume 5, pages 296–304.

Shao, Y.-H., Li, C.-N., Liu, M.-Z., Wang, Z., and Deng, N.-Y. (2018). Sparse lq-norm least squares support vector machine with feature selection. *Pattern Recognition*, 78:167–181.

Solorio-Fernández, S., Martínez-Trinidad, J. F., and Carrasco-Ochoa, J. A. (2017). A new unsupervised spectral feature selection method for mixed data: a filter approach. *Pattern Recognition*, 72:314–326.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

Sridhar, R., Sivakumar, V., Immanuel, V. P., and Maran, J. P. (2012). Development of model for treatment of pulp and paper industry bleaching effluent using response surface methodology. *Environmental Progress & Sustainable Energy*, 31(4):558–565.

Talukdar, U., Hazarika, S. M., and Gan, J. Q. (2018). A kernel partial least square based feature selection method. *Pattern Recognition*, 83:91–106.

Tang, X., Dai, Y., and Xiang, Y. (2018). Feature selection based on feature interactions with application to text categorization. *Expert Systems with Applications*, 120.

Wang, L. and Mehr, M. N. (2019). An optimization approach to epistasis detection. *European Journal of Operational Research*, 274(3):1069–1076.

Xu, S. and Jia, Z. (2007). Genomewide analysis of epistatic effects for quantitative traits in barley. *Genetics*, 175(4):1955–1963.

Yin, Q., Zhang, J., Wu, S., and Li, H. (2019). Multi-view clustering via joint feature selection and partially constrained cluster label learning. *Pattern Recognition*, 93:380–391.

Zeng, Z., Zhang, H., Zhang, R., and Yin, C. (2015). A novel feature selection method considering feature interaction. *Pattern Recognition*, 48(8):2656–2666.

Zhang, J., Luo, Z., Li, C., Zhou, C., and Li, S. (2019a). Manifold regularized discriminative feature selection for multi-label learning. *Pattern Recognition*.

Zhang, P., Liu, G., and Gao, W. (2019b). Distinguishing two types of labels for multi-label feature selection. *Pattern Recognition*.

Zhao, Z. and Liu, H. (2009). Searching for interacting features in subset selection. *Intelligent Data Analysis*, 13(2):207–228.

Zheng, K. and Wang, X. (2018). Feature selection method with joint maximal information entropy between features and class. *Pattern Recognition*, 77:20–29.

Zhou, P., Hu, X., Li, P., and Wu, X. (2019). Ofs-density: A novel online streaming feature selection method. *Pattern Recognition*, 86:48–61.

Zhuo, L., Zheng, J., Li, X., Wang, F., Ai, B., and Qian, J. (2008). A genetic algorithm based wrapper feature selection method for classification of hyperspectral images using support vector machine. In *Geoinformatics 2008 and Joint Conference on GIS and Built Environment: Classification of Remote Sensing Images*, volume 7147, page 71471J. International Society for Optics and Photonics.

Chapter3. Discovering Interacting Features for Prediction of Binary Response

Abstract

This paper propose a new method for finding a subset of features which are important on the classification problems. A mathematical model for classification of two finite point sets in n-dimensional feature space by a separating plane that utilize as few of the feature as possible is presented. The purpose of our work is to establish a computational framework for selection a subset of features that are independently important for the target prediction and a subset of features that might not be considered important independently but in the presence of other features, they can strongly be predictive of the response. Unintended removal of these features can contribute to a poor prediction of target. The mathematical model with step function that appear in the objective function and nonlinear constraints with its equivalent linear program with equilibrium constraints (LPEC) is presented to find a separating place such that minimizes the number of misclassifications. Computational tests of the method is presented on the publicly available real-world databases.

3.1 Introduction

Feature selection is an important problem in machine learning (Bradley et al. (1998); Dash and Liu (1997); John et al. (1994)). Living in a big data era and dealing with large data sets emphasized on the importance of feature selection to reduce training time, dimensionality, avoid over-fitting to improve prediction (Das (2001)). Feature selection algorithms are categorized in two groups: filter (Santana and de Paula Canuto (2014); Abe and Kudo (2006); Wei and Billings (2006)) and wrapper models (Maldonado and Weber (2009); Guyon et al. (2002); Zhu et al. (2007)). Filter methods are developed dependant on variable ranking techniques, while wrapper methods consider the feature subset as a black box and choose the best subset based on the performance of the objective function. Based on feature selection concept, for a classification problem with n features, only a small number of features give sufficient information for the classification. These features are known as relevant features. It has been believed that irrelevant features does not affect the target concept in any way and redundant features does not add any information to the response (John et al. (1994)). As a result, most of the machine learning algorithms exclude irrelevant (Blum and Langley (1997); Radha and Muralidhara (2016); Yu and Liu ()) and redundant features from their training data sets (Appice et al. (2004); Zeng et al. (2015); Koller and Sahami (1996)) to avoid poor accuracy of predictive models.

In this research we argue that removal of redundant and irrelevant features does not always improve predictions. Features could have little correlation with the class while in presence of the other features, they could be significantly related to the class (Cotter et al. (2001); Kira and Rendell (1992)). Features could be considered redundant while in the presence of other features, they could be strongly related to the class (Kira and Rendell (1992)). This brings the idea of feature interaction in prediction modeling.

Recent studies have developed different methods for detecting feature interaction. Zhao and Liu (2009) develop an algorithm named INTERACT to select relevant features while considering interaction based on feature scoring metric. Wang et al. (2013) proposed a FOIL rule algorithm FRFS to detect relevant features and eliminate irrelevant features while considering feature interaction. Zeng et al. (2015) presented an Interaction Weight based Feature Selection algorithm (IWFS), a novel feature selection method that considers interaction. They used the interaction weight factor to specify whether the information of a feature is redundant or interactive. Tang et al. (2018) proposed a new feature selection method, Five-way Joint Mutual Information (FJMI), that look for two to five-way interaction between feature and the class label. Studies in genetic also investigate the effect of interaction of multiple genes in developing complex disease (Collins et al. (2013); Maher (2008); Moore et al. (2010)). Exhaustive search of two way interaction (Evans et al. (2006)) and heuristic algorithms with non-exhaustive search for higher order interaction (Kässens et al. (2002); Leem et al. (2014); Ritchie et al. (2001); Xie et al. (2012); Upstill-Goddard et al. (2013)) are the methods researchers developed for detecting feature interaction. For example, Xie et al. (2012) state the FSCF method that is based on the clustering of relatively frequent items.

Discovering feature interaction is a challenging task in feature selection (Zeng et al. (2015)). Filter methods do not consider classification of data in the process of feature subset evaluation (Zeng et al. (2015)). Wrapper methods "utilize the performance of a specific classifier to evaluate feature subsets by different search strategies" (Zeng et al. (2015)). Therefore, filter models does not have the ability of detecting interacting features and including them in their subset selection, while wrapper methods might be able to include feature interaction in the selected subset. However these methods require a model that tests each feature subset, which is computationally expensive (Zeng et al. (2015)). Moreover, such approaches will not explicitly identify which features interact. For example, sometimes feature selection is based on a random forest model, where the features are ranked according to how often they are selected to be used as part of a decision tree, that is, how well they work in interaction with other features in the tree. Such an approach should be able to pick up on interacting features but will not explicitly identify which features interact. Another well known approach that should be able to pick up interactions is Relief, which ranks features based on how well they perform in subsets of features to do instance-based (nearest neighbor) prediction. Again, this should allow us to pick up on interactions without explicitly identify the interacting features, but the concern is that the final feature selection might exclude one or more interacting feature because it is unknown that it work only as part of a pair. The unique element of the proposed approach is that it explicitly identifies interacting features that should be included in pairs. In this study we emphasized on selecting a good subset of features that can improve our prediction performance. We show existence of redundant or irrelevant features might add valuable information to our prediction. Our focus is finding a subset of relevant variables to build our predictors. In this research we developed a new feature selection algorithm based on a mathematical programming model with step function that appears in the objective function and nonlinear constraints. We also represent its equivalent linear program with equilibrium constraints (LPEC) to find a separating plane with consideration of feature interaction such that it minimizes the number of misclassifications.

3.2 Problem Statement and Motivation

Relevancy, redundancy, and interaction of features are the information theory concepts of feature selection algorithms. Most of previous studies focus on the feature selection algorithms designed based on the definitions of relevant and redundant features. If feature is correlated or predictive of a class, it is useful for the prediction; otherwise, it is considered irrelevant (Gennari et al. (1989)). However, some features influence the class when they are grouped with other features. To illustrate this phenomenon consider problem XOR as shown in Table 3.2.

Table 3.1: problem XOR

F_1	F_1	$Class$
1	0	1
0	1	1
1	1	0
0	0	0

In this example, the correlation between features and class is zero. As a result, none of the features F_1 and F_2 separately could determine the class independently. However, F_1, F_2 together, successfully lead to true classification. This emphasized on the importance of feature interaction in predictive modeling and point out on the fact that evaluating features independent of other features based on relevancy and redundancy is not always wise.

3.2.1 Related Work

Our new feature selection model is defined based on: (1) proposing a separating plane that discriminates between data point of sets $\mathcal{A}$ and $\mathcal{B}$, and (2) adding new dimensions to n-dimensional feature space to consider interaction effect for classification problems. Our model is an extension of two papers in the literature that we will discuss separately in the following Sections.

3.2.1.1 Notation

Here we present the notations we used in this paper. All vectors will be column vectors transposed to a row to a vector by a superscript T. For a vector x in the n-dimensional real space R^n, $|x|$ will denote a vector of absolute values of the components x_j , $j = 1, ..., n$ of x. For a vector x in R^n, x_+ denotes the vector in R^n with components $max\{0, x_i\}$. For a vector x in R^n, x denotes the vector in R^n with components $(x_\star)_i$ equal 1 if $x_i > 0$ and 0 otherwise (i.e. $x_\star$ is the result of applying the step function to the components of x). The notation $A \in R^{m\times n}$ will signify a real $m \times n$ matrix. For such a matrix A^T will denote the transpose of A and A_i will denote the i-th row of A. A vector of ones in a real space of arbitrary dimension will be denoted by e. A vector of zeros in a real space of arbitrary dimension will be denoted by 0. And for two vectors x, $y \in R^n$, $x \perp y$ denotes the scalar product $x \perp y = 0$. A separating plane, with respect to two given point sets A and B in R^n, is a plane that attempts to separate R^n into two halfspaces such that each open halfspace contains points mostly of A or B. We can also interpret the plan as a classical perception (Bradley et al. (1999)).

3.2.1.2 Classification- Mathematical Programming Formulation

Bradley et al. (1999) addressed the task of estimating a classification function which assigns a given vector $x \in R^n$ in to two disjoint point sets $\mathcal{A}$ or $\mathcal{B}$ in n-dimentioanl feature space. Considering the $\chi = R^n$ and class $y = \{0, 1\}$, the classification function is as follows.

$$g(x) \quad = \quad \begin{cases} 1 & \text{if } x \in \mathcal{A} \\ 0 & \text{if } x \in \mathcal{B} \end{cases} \tag{3.1}$$

Sets $\mathcal{A}$ and $\mathcal{B}$ in R^n consist of m and k points respectively such that $A \in R^{m\times n}$ and $B \in R^{k\times n}$. They attempted to find a plane that discriminate between the points of $\mathcal{A}$ and $\mathcal{B}$. Therefore, they defined problem P for the purpose of constructing a separating plane:

$$P \quad = \quad \{x|x \in R^n, x^T\omega = \gamma\}, \tag{3.2}$$

with normal $\omega \in R^n$ and distance $\dfrac{|\gamma|}{||\omega||_2}$ to the origin. They defined two open halfspaces $\{x|x \in R^n, x^T\omega > \gamma\}$ and $\{x|x \in R^n, x^T\omega < \gamma\}$ to respectively represent most points of sets $\mathcal{A}$ and $\mathcal{B}$ and finally presented model (3.3) to minimize the number of points misclassified by the plane P (3.2).

$$\min_{\omega,\gamma} \quad e^T(-A\omega + e\gamma + e)_* + e^T(B\omega - e\gamma + e)_* \tag{3.3}$$

And finally, they reduced (3.3) to a precise mathematical programming formulation of the nonconvex problem of minimizing the number of misclassified points by proposing a simple linear complementarity formulation of the step function (Lemma 2.1, Bradley et al. (1999)) and then transformed the result to a linear program (3.4-3.10) with equilibrium (linear complementarity) constraints (LPEC).

$$\begin{array}{rccr}
\min\limits_{\omega,\gamma,r,u,s,v} & e^Tr + e^Ts & & (3.4)\\
\text{s.t.} & u + A\omega - e\gamma - e \geq 0 & v - B\omega + e\gamma - e \geq 0 & (3.5)\\
& r \geq 0 & s \geq 0 & (3.6)\\
& r^T(u + A\omega - e\gamma - e) = 0 & s^T(v - B\omega + e\gamma - e) = 0 & (3.7)\\
& -r + e \geq 0 & -s + e \geq 0 & (3.8)\\
& u \geq 0 & v \geq 0 & (3.9)\\
& u^T(-r + e) = 0 & v^T(-s + e) = 0 & (3.10)
\end{array}$$

Model (3.4-3.10) finds the best separating plane having a minimum number of missclassifications while detecting data point of sets $\mathcal{A}$ and $\mathcal{B}$. Up to this point we presented the mathematical formulation model for computing a separating plane in a classification problem. To improve the generalization ability of the separating plane (P), we have the objective function of the model (3.11-3.17) penalized with parameter $\alpha \in [0, 1)$ for nonzero elements of the weight vector ω while weighting the original objective function by $(1 - \alpha)$ (Bradley et al. (1999)).

$$\begin{array}{rccr}
\min\limits_{\omega,\gamma,r,u,s,v} & (1-\alpha)(e^Tr + e^Ts) + \alpha e^T|\omega|_* & & (3.11)\\
\text{s.t.} & u + A\omega - e\gamma - e \geq 0 & v - B\omega + e\gamma - e \geq 0 & (3.12)\\
& r \geq 0 & s \geq 0 & (3.13)\\
& r^T(u + A\omega - e\gamma - e) = 0 & s^T(v - B\omega + e\gamma - e) = 0 & (3.14)\\
& -r + e \geq 0 & -s + e \geq 0 & (3.15)\\
& u \geq 0 & v \geq 0 & (3.16)\\
& u^T(-r + e) = 0 & v^T(-s + e) = 0 & (3.17)
\end{array}$$

Notice that the vector $|\omega|_\star$ is a step function which means based on Lemma 2.1, Bradley et al. (1999), its element are equal to 1 if the corresponding components of ω are nonzero, and equal to zero if the corresponding components of ω are zero. Therefore $e^T|\omega|_\star$ counts the number of nonzero components of ω. The equivalent formulation of model (3.11-3.17) is as follows.

$$\min_{\omega,\gamma,r,u,s,v,\phi} \quad (1-\alpha)(e^Tr+e^Ts)+\alpha e^T\phi_* \tag{3.18}$$
$$\text{s. t.} \quad u+A\omega-e\gamma-e\geq 0 \qquad v-B\omega+e\gamma-e\geq 0 \tag{3.19}$$
$$r\geq 0 \qquad s\geq 0 \tag{3.20}$$
$$r^T(u+A\omega-e\gamma-e)=0 \qquad s^T(v-B\omega+e\gamma-e)=0 \tag{3.21}$$
$$-r+e\geq 0 \qquad -s+e\geq 0 \tag{3.22}$$
$$u\geq 0 \qquad v\geq 0 \tag{3.23}$$
$$u^T(-r+e)=0 \qquad v^T(-s+e)=0 \tag{3.24}$$
$$-\phi\leq\omega\leq\phi \tag{3.25}$$

Step function $\phi_\star$ will be linearized using Lemma 2.1 in Bradley et al. (1999) and feature selection problem (3.26-3.39) will be solved for a value of $\alpha\in[0,1)$ for which the plane P generalized the best.

$$\min_{\omega,\gamma,r,u,s,v,,\phi,\psi} \quad (1-\alpha)(e^Tr+e^Ts)+\alpha e^T\psi \tag{3.26}$$
$$\text{s. t.} \quad u+A\omega-e\gamma-e\geq 0 \qquad v-B\omega+e\gamma-e\geq 0 \tag{3.27}$$
$$r\geq 0 \qquad s\geq 0 \tag{3.28}$$
$$r^T(u+A\omega-e\gamma-e)=0 \qquad s^T(v-B\omega+e\gamma-e)=0 \tag{3.29}$$
$$-r+e\geq 0 \qquad -s+e\geq 0 \tag{3.30}$$
$$u\geq 0 \qquad v\geq 0 \tag{3.31}$$
$$u^T(-r+e)=0 \qquad v^T(-s+e)=0 \tag{3.32}$$
$$-\phi\leq\omega\leq\phi \tag{3.33}$$
$$p-\phi\geq 0 \tag{3.34}$$
$$\psi\geq 0 \tag{3.35}$$
$$\psi^T(p-\phi)=0 \tag{3.36}$$
$$-\psi+e\geq 0 \tag{3.37}$$
$$p\geq 0 \tag{3.38}$$
$$p^T(-\psi+e)=0 \tag{3.39}$$

3.2.2 Proposed Feature Selection Algorithm

In this section, we present a new feature selection algorithm for discriminating two nonempty finite sets $\mathcal{A}$ and $\mathcal{B}$ in B^n in the presence of feature interaction. As already discussed, feature could be considered redundant or irrelevant to the class, while in the presence of other features, they could be highly related to the class. To capture this phenomenon, we present a new feature selection algorithm, which is similar to the feature selection algorithm presented in Bradley et al. (1999) in essence but is more flexible to consider the effect of interacting features while classifying data points.

Detecting interaction effect has been studied widely. Most of these methods looked for second order interaction effect through exhaustive search or developed heuristic algorithms for targeting higher order interaction effects. In this research, we used the technique proposed in Wang and Mehr (2019) for data sets with continuous response predictions. In this study, they added an indicator function to the linear regression model that captures the presence or absence of interaction effect. If interaction effect is observable in a data set, its effect would be added to the response prediction to reduce error and improve prediction ability, Model (3.40).

$$y_i \;=\; \beta_0 + \sum_{j=1}^{n} X_{ij}\beta_j + \sum_{l=1}^{L} b^l \cdot I\left(\sum_{j=1}^{n} X_{ij}(\lambda_j^l - \mu_j^l) = \sum_{j=1}^{n} \lambda_j^l\right) + \epsilon_i, \forall i \in \{1, ..., m\}. \quad (3.40)$$

Here m denotes the number of observation and n specifies the number of features. $X_{i,j}$ represents the feature j for observation i, which is assumed to be binary and y_i corresponds to the response variable for observation i. β_0 is the intercept effect for any i with $X_{i,j} = 0$; β_j represents the partial regression coefficient, which is the differential effect of $X_{i,j} = 1$ over $X_{i,j} = 0$; b is the magnitude of the interaction effect, and $I(\cdot)$ is the indicator function, which is equal to 1 if the statement in the parenthesis is true and 0 otherwise. λ_j^l and , μ_j^l are binary variables that define the l-th interaction effect, which is triggered if and only if $X_{ij} = 1$, $\forall j \in \{j : \lambda_j^l = 1\}$ and $X_{ij} = 0$, $\forall j \in \{j : \mu_j^l = 1\}$, for any i. ϵ_i is the random error term for observation i.

To develop our new feature selection model, we modify the plane P to include interaction effect. As a result, for a given vector $x \in B^n$ in to two disjoint point sets $\mathcal{A}$ or $\mathcal{B}$ in (n+l)-dimentioanl feature space, with l being the number of observable interaction effects, the new separating plane P' is:

$$P' \;=\; \{x | x \in R^{n+l}, x^T\omega + \Theta^T e = \gamma\}, \quad (3.41)$$

where, Θ is a column vector $\in R^{l\times 1}$ representing the magnitude of interaction effects and their corresponding combinations of features. Finally, we present model (3.42) to minimize the number of misclassified points by plane P'.

$$\min_{\omega,\gamma} \quad e^T(-A\omega - \Theta^T e + e\gamma + e)_* + e^T(B\omega - \Theta^T e - e\gamma + e)_* \quad (3.42)$$

Using Lemma 2.1 in Bradley et al. (1999), we will reduce (3.42) to an LPEC by representing the step function $(\cdot)_\star$ as a complimentarity condition via the the plus function $(\cdot)_+$, which is a restatement of Mangasarian (1996), Equation 2.11. And to determine the magnitude of interaction effects and interacting features, we implement the method presented by Wang and Mehr (2019).

$$\min_{\omega,\gamma,r,u,s,v,\lambda,\mu,z,b} \quad \sum_{i=1}^{m} r_i + \sum_{i=1}^{k} s_i + \sum_{i=1}^{m+k}\sum_{l=1}^{L} z_i^l \tag{3.43}$$

$$\text{s. t.} \quad u_i + \sum_{j=1}^{n} A_{ij}\omega_j + \sum_{l=1}^{L} b^l z_i^l - \gamma - 1 \geq 0 \qquad \forall i \in \{1,...,m\} \tag{3.44}$$

$$r_i \geq 0 \qquad \forall i \in \{1,...,m\} \tag{3.45}$$

$$\sum_{i=1}^{m} r_i(u_i + \sum_{j=1}^{n} A_{ij}\omega_j + \sum_{l=1}^{L} b^l z_i^l - \gamma - 1) = 0 \tag{3.46}$$

$$-r_i + 1 \geq 0 \qquad \forall i \in \{1,...,m\} \tag{3.47}$$

$$u_i \geq 0 \qquad \forall i \in \{1,...,m\} \tag{3.48}$$

$$\sum_{i=1}^{m} u_i(-r_i + 1) = 0 \tag{3.49}$$

$$v_i - \sum_{j=1}^{n} B_{ij}\omega_j - \sum_{l=1}^{L} b^l z_i^l + \gamma - 1 \geq 0 \qquad \forall i \in \{1,...,k\} \tag{3.50}$$

$$s_i \geq 0 \qquad \forall i \in \{1,...,k\} \tag{3.51}$$

$$\sum_{i=1}^{k} s_i(v_i - \sum_{j=1}^{n} B_{ij}\omega_j - \sum_{l=1}^{L} b^l z_i^l + \gamma - 1) = 0 \tag{3.52}$$

$$-s_i + 1 \geq 0 \qquad \forall i \in \{1,...,k\} \tag{3.53}$$

$$v_i \geq 0 \qquad \forall i \in \{1,...,k\} \tag{3.54}$$

$$\sum_{i=1}^{k} v_i(-s_i + 1) = 0 \tag{3.55}$$

$$\sum_{j=1}^{n} A_{ij}(\lambda_j^l - \mu_j^l) \geq -n(1 - z_i^l) + \sum_{j=1}^{n} \lambda_j^l \qquad \forall i \in \{1,...,m\}, \forall l \in \{1,...,L\} \tag{3.56}$$

$$\sum_{j=1}^{n} B_{ij}(\lambda_j^l - \mu_j^l) \geq -n(1 - z_i^l) + \sum_{j=1}^{n} \lambda_j^l \qquad \forall i \in \{1,...,k\}, \forall l \in \{1,...,L\} \tag{3.57}$$

$$\sum_{j=1}^{n} A_{ij}(\lambda_j^l - \mu_j^l) \leq \sum_{j=1}^{n} \lambda_j^l - 1 + nz_i^l \qquad \forall i \in \{1,...,m\}, \forall l \in \{1,...,L\} \tag{3.58}$$

$$\sum_{j=1}^{n} B_{ij}(\lambda_j^l - \mu_j^l) \leq \sum_{j=1}^{n} \lambda_j^l - 1 + nz_i^l \qquad \forall i \in \{1,...,k\}, \forall l \in \{1,...,L\} \tag{3.59}$$

$$\lambda_j^l + \mu_j^l \leq 1 \qquad \forall j \in \{1,...,n\}, \forall l \in \{1,...,L\} \tag{3.60}$$

$$z_i^l, \lambda_j^l, \mu_j^l \in \{0,1\} \qquad \forall i \in \{1,...,m+k\}, \forall j \in \{1,...,n\} \, , \forall l \in \{1,...,L\} \tag{3.61}$$

$$b^l, \omega_i, \gamma \text{ free} \qquad \forall i \in \{1,...,m+k\}, \forall l \in \{1,...,L\} \tag{3.62}$$

Here $\sum_{l=1}^{L} b^l z_i^l$ represents $\Theta \in R^{l\times 1}$ in plane P'. The objective (3.43) is to minimize the number of misclassifications for sets $\mathcal{A}$ and $\mathcal{B}$ in B^n. We penalize the Objective with variable z_i^l to avoid adding the l-th dimension for all observations. Constraints (3.44 - 3.55) define whether a point in set $\mathcal{A}$ or $\mathcal{B}$ in B^n is being missclassified or not. Constraints (3.56 - 3.59) jointly define a binary variable z_i^l that checks the presence of the interaction effect in each observation: $z_i^l = 1$ if an only if $\sum_{j=1}^{n} A_{ij}(\lambda_j^l - \mu_j^l) = \sum_{j=1}^{n} \lambda_j^l$, fo all $i \in \{1, ..., m\}$ or $\sum_{j=1}^{n} B_{ij}(\lambda_j^l - \mu_j^l) = \sum_{j=1}^{n} \lambda_j^l$, fo all $i \in \{1, ..., k\}$. Constraints (3.56) and (3.57) ensures the "only if" direction: if $z_i^l = 1$, then $\sum_{j=1}^{n} A_{ij}(\lambda_j^l - \mu_j^l) \geq \sum_{j=1}^{n} \lambda_j^l$ or $\sum_{j=1}^{n} B_{ij}(\lambda_j^l - \mu_j^l) \geq \sum_{j=1}^{n} \lambda_j^l$; since $\sum_{j=1}^{n} A_{ij}(\lambda_j^l - \mu_j^l) \leq \sum_{j=1}^{n} A_{ij}\lambda_j^l \leq \sum_{j=1}^{n} \lambda_j^l$ is always true, it leads to the equation $\sum_{j=1}^{n} A_{ij}(\lambda_j^l - \mu_j^l) = \sum_{j=1}^{n} \lambda_j^l$. We have the same equations for data points of set $\mathcal{B}$. Conversely, constraint (3.58) and (3.59) enforce the "if" direction: if $\sum_{j=1}^{n} A_{ij}(\lambda_j^l - \mu_j^l) \geq \sum_{j=1}^{n} \lambda_j^l$, then $z_i^l = 1$. Constraint (3.61) means that the interaction effect cannot logically require the jth feature to be one($\lambda_j^l = 1$) and zero ($\mu_j^l = 1$) at the same time. Constraints (3.61) and (3.62) define the appropriate types of decision variables. Model (3.43-3.62) finds the best separating plane having a minimum number of missclassifications while detecting data points of sets $\mathcal{A}$ and $\mathcal{B}$. Model (3.43-3.62) is a nonlinear non-convex combinatorial optimization problem, generally hard to solve. In the following, we reformulate model (3.43-3.62) into an equivalent mixed integer linear program (MILP) by introducing two sets of new variables. For $i \in \{1, ..., m+k\}$, a free variable θ_i^l is defined to be equal to $b^l \cdot z_i^l$, and variables $y_i^1, y_i^2, y_i^3, y_i^4$ are binary variables to make sure complimentarity constraints are satisfied. We also assumed that we know the upper and lower bounds of the magnitude of the interaction effects: $\underline{b} \leq b \leq \bar{b}$, which is necessary to linearize the bilinear term $b^l \cdot z_i^l$. M is a parameter needed for linearization of complimantarity constraints.

$$
\begin{align}
\min_{\omega,\gamma,r,u,s,v} \quad & \sum_{i=1}^{m} r_i + \sum_{i=1}^{k} s_i + \sum_{i=1}^{m+k} \sum_{l=1}^{L} z_i^l && \tag{3.63}\\
\text{s.t.} \quad & u_i + \sum_{j=1}^{n} A_{ij}\omega_j + \sum_{l=1}^{L} \theta^l - \gamma - 1 \geq 0 && \forall i \in \{1, ..., m\} \tag{3.64}\\
& 0 \leq r_i \leq M y_i^1 && \forall i \in \{1, ..., m\} \tag{3.65}\\
& u_i + \sum_{j=1}^{n} A_{ij}\omega_j + \sum_{l=1}^{L} \theta^l - \gamma - 1 \leq M(1 - y_i^1) && \forall i \in \{1, ..., m\} \tag{3.66}\\
& 0 \leq -r_i + 1 \leq M y_i^2 && \forall i \in \{1, ..., m\} \tag{3.67}\\
& 0 \leq u_i \leq M(1 - y_i^2) && \forall i \in \{1, ..., m\} \tag{3.68}\\
& v_i - \sum_{j=1}^{n} B_{ij}\omega_j - \sum_{l=1}^{L} \theta^l + \gamma - 1 \geq 0 && \forall i \in \{1, ..., k\} \tag{3.69}\\
& 0 \leq s_i \leq M y_i^3 && \forall i \in \{1, ..., k\} \tag{3.70}\\
& v_i - \sum_{j=1}^{n} B_{ij}\omega_j - \sum_{l=1}^{L} \theta^l + \gamma - 1 \leq M(1 - y_i^3) && \forall i \in \{1, ..., k\} \tag{3.71}\\
& 0 \leq -s_i + 1 \leq M y_i^4 && \forall i \in \{1, ..., k\} \tag{3.72}\\
& 0 \leq v_i \leq M(1 - y_i^4) && \forall i \in \{1, ..., k\} \tag{3.73}\\
& \sum_{j=1}^{n} A_{ij}(\lambda_j^l - \mu_j^l) \geq -n(1 - z_i^l) + \sum_{j=1}^{n} \lambda_j^l && \forall i \in \{1, ..., m\}, \forall l \in \{1, ..., L\} \tag{3.74}\\
& \sum_{j=1}^{n} B_{ij}(\lambda_j^l - \mu_j^l) \geq -n(1 - z_i^l) + \sum_{j=1}^{n} \lambda_j^l && \forall i \in \{1, ..., k\}, \forall l \in \{1, ..., L\} \tag{3.75}\\
& \sum_{j=1}^{n} A_{ij}(\lambda_j^l - \mu_j^l) \leq \sum_{j=1}^{n} \lambda_j^l - 1 + n z_i^l && \forall i \in \{1, ..., m\}, \forall l \in \{1, ..., L\} \tag{3.76}\\
& \sum_{j=1}^{n} B_{ij}(\lambda_j^l - \mu_j^l) \leq \sum_{j=1}^{n} \lambda_j^l - 1 + n z_i^l && \forall i \in \{1, ..., k\}, \forall l \in \{1, ..., L\} \tag{3.77}\\
& \lambda_j^l + \mu_j^l \leq 1 && \forall j \in \{1, ..., n\}, \forall l \in \{1, ..., L\} \tag{3.78}\\
& min\{0, \underline{b}\} \leq \theta_i^l \leq \bar{b} && \forall i \in \{1, ..., m+k\}, \forall l \in \{1, ..., L\} \tag{3.79}\\
& \underline{b} z_i^l \leq \theta_i^l \leq \bar{b} z_i^l && \forall i \in \{1, ..., m+k\}, \forall l \in \{1, ..., L\} \tag{3.80}\\
& b^l - (1 - z_i^l)\bar{b} \leq \theta_i^l \leq b^l - (1 - z_i^l)\underline{b} && \forall i \in \{1, ..., m+k\}, \forall l \in \{1, ..., L\} \tag{3.81}\\
& \theta_i^l \leq b^l + (1 - z_i^l)\bar{b} && \forall i \in \{1, ..., m+k\}, \forall l \in \{1, ..., L\} \tag{3.82}\\
& z_i^l, \lambda_j^l, \mu_j^l \in \{0, 1\} && \forall i \in \{1, ..., m+k\}, \forall j \in \{1, ..., n\} \nonumber\\
& && , \forall l \in \{1, ..., L\} \tag{3.83}
\end{align}
$$

$$
\begin{align}
b^l, \omega_i, \gamma, \theta_i^l \text{ free} \quad & \forall i \in \{1, ..., m+k\}, \forall l \in \{1, ..., L\} \nonumber\\
& , \forall l \in \{1, ..., L\} \tag{3.84}
\end{align}
$$

$$b^l, \omega_i, \gamma \text{ free} \qquad \forall i \in \{1, ..., m+k\}, \forall l \in \{1, ..., L\} \tag{3.85}$$

$$z_i^l, \lambda_j^l, \mu_j^l \in \{0, 1\} \quad \forall i \in \{1, ..., m+k\}, \forall j \in \{1, ..., n\} \tag{3.86}$$

The Objective (3.63) is same as Objective (3.43). Constraints (3.64)-(3.78) are equivalent to the Constraints (3.43)-(3.61) while being bounded by M to linearize complimantarity constraints. Constraints (3.79)-(3.82) are equivalent to the equation $\theta_i^l = b \cdot z_i^l$ when $\underline{b} \leq b \leq \bar{b}$, which is a commonly used linearization technique [30]. And Constraints (3.83)-(3.86) define the appropriate types of the decision variables.

To develop the feature selection model and improve the generalization ability of the separating plane P', we penalized the objective function (3.63) with parameter $\alpha \in [0, 1)$ for nonzero elements of the weight vector ω while weighting the original objective function by $(1-\alpha)$ (Bradley et al. (1999)). Feature selection model will be defined based on Model (3.26)-(3.39) and will be linearized as follows. We will refer to the following MILP(3.87)-(3.97) as $\mathcal{M}(X, y)$, with X being Observational sets $\mathcal{A}$ and $\mathcal{B}$, and y representing the class.

$$\min_{\omega,\gamma,r,u,s,v,\psi} \quad (1-\alpha)(\sum_{i=1}^{m} r_i + \sum_{i=1}^{k} s_i) + \alpha(\sum_{i=1}^{m+k} \sum_{l=1}^{L} z_i^l + \sum_{j=1}^{n} \psi_j) \tag{3.87}$$

$$\text{s. t.} \qquad \text{Constraints}(3.64) - (3.82) \tag{3.88}$$

$$-\phi_j \leq \omega_j \leq \phi_j \qquad \forall j \in \{1, ..., n\} \tag{3.89}$$

$$0 \leq \psi_j \leq M y_j^5 \qquad \forall j \in \{1, ..., n\} \tag{3.90}$$

$$0 \leq p_j - \phi_j \leq M(1 - y_j^5) \qquad \forall j \in \{1, ..., n\} \tag{3.91}$$

$$0 \leq p_j \leq M y_j^6 \qquad \forall j \in \{1, ..., n\} \tag{3.92}$$

$$0 \leq -\psi_j + 1 \leq M(1 - y_j^6) \qquad \forall j \in \{1, ..., n\} \tag{3.93}$$

$$z_i^l, \lambda_j^l, \mu_j^l, y_i^1, y_i^2, y_i^3, y_i^4, y_j^5, y_j^6 \in \{0, 1\} \qquad \forall i \in \{1, ..., m+k\}, \forall j \in \{1, ..., n\} \; , \forall l \in \{1, ..., L\} \tag{3.94}$$

$$b^l, \omega_i, \gamma, \theta_i^l \text{ free} \qquad \forall i \in \{1, ..., m+k\}, \forall l \in \{1, ..., L\} \; , \forall l \in \{1, ..., L\} \tag{3.95}$$

$$b^l, \omega_i, \gamma \text{ free} \qquad \forall i \in \{1, ..., m+k\}, \; \forall l \in \{1, ..., L\} \tag{3.96}$$

$$z_i^l, \lambda_j^l, \mu_j^l \in \{0, 1\} \qquad \forall i \in \{1, ..., m+k\}, \; \forall j \in \{1, ..., n\} \tag{3.97}$$

Here, the Objective (3.87) is to minimize the total number of missclassifications while using regularization techniques to avoid over fitting. Constraints (3.89)-(3.93) are used to define and linearize the step function, which corresponds to regilarization. Variables y_j^5, y_j^6 are binary variables to make sure complimentarity constraints are satisfied. And Constraints (3.94)-(3.97) define the appropriate types of the decision variables.

3.3 Computational Experiments

We conducted computational experiments to test the effectiveness of the proposed approach.

3.3.1 Intuitive Example

In this section, we empirically evaluate the performance of our proposed feature selection model, and present the result compare to the feature selection model Bradley et al. (1999) presented. For this reason, we present the following data set in the 3-dimensional space. Our reasoning for proposing this data set, is that 3-dimensioanl space is sensible to imagine and easy to draw in a paper. The data set is as follows.

Table 3.2: Intuitive example

F_1	F_2	F_3	$Class$
0	1	0	1
1	1	1	1
0	0	0	1
0	1	0	1
0	0	1	0
1	0	0	0
1	1	0	0

Figure (3.1) shows the data points in the 3-dimensional space. Blue data points, represents data points in class 1 and points in red, represent class 0. As it is observable, there is no separating plane that can classify data points without having misclassifications. The new feature selection model proposed in Section 3.2.2 is implemented in Python, using package Pyomo and CPLEX as the MILP solver for Model $\mathcal{M}(X, y)$. The feature selection model presented by Bradley et al. (1999) was also implemented in Python for comparison. We conducted sensitivity analysis of parameters M and α that could affect the effectiveness of the optimization Model (3.87)-(3.97). For this specific problem, we observe the same solution for different selection values of these two parameters. To select the lower bound and upper bound of magnitude of interaction effect (b), we fist ran the Bradley et al. (1999) model and found the coefficient of features in the separating plane and multiplied their values by -10 and $+10$ to have a wide range to consider. Table 3.3 shows the comparison between two feature selection models when $M = 10$ and $\alpha = 0.1$.

Based on table (3.3), both models select features x_2 and x_3 as the most important features. Bradley et al. (1999) Model finds the separating feature selection plane such that point $(0, 0, 0)$ is

Figure 3.1: Intuitive example

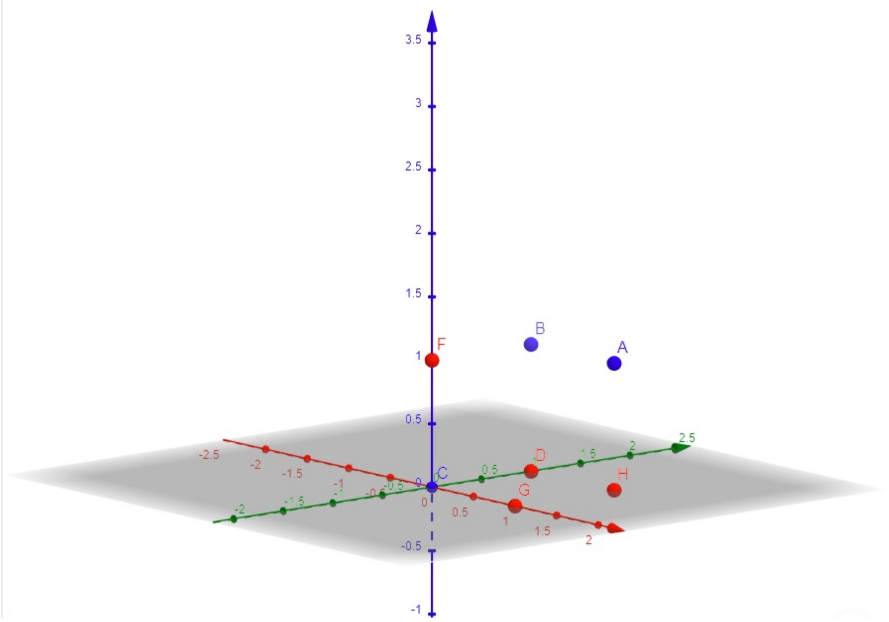


Table 3.3: Comparison of Feature Selection Models

Method	Selected Features	Interacting Features	Separating Plane	Missclassified Point
Bradley []	x_2, x_3	-	$2x_2 + 2x_3 = 3$	$(0, 0, 0)$
$\mathcal{M}(X, y)$	x_2, x_3	$x_1 = 0, x_2 = 0, x_3 = 0$	$2x_2+2x_3+4(1-x_1)(1-x_2)(1-x_3) = 3$	-

missclassified. However, Model $\mathcal{M}(X, y)$, finds an interaction effect between features x_1, x_2, and x_3. When all three features are equal to 0, adding a new dimension to the three dimensional space leads to 0 misclassifications.

3.3.2 Experiments on Real-world Data

In this section, we empirically evaluate the performance of our proposed feature selection model, and present the experimental result compared with Bradley et al. (1999) Model on real data sets. To verify the effectiveness of our method on real world problems, we consider implementation of the algorithm on Monk's and Lymphography data sets (Dua and Graff (2017)). Both data sets are appropriate for classification problems with categorical features.

3.3.2.1 MONK's Data set

The MONK's problem were the basis of a first international comparison of learning algorithms. One significant characteristic of this comparison is that it was performed by a collection of researchers, each of whom was an advocate of the technique they tested (often they were the creators of the various methods). In this sense, the results are less biased than in comparisons performed by a single person advocating a specific learning method, and more accurately reflect the generalization behavior of the learning techniques as applied by knowledgeable users. There are three different variations of Monk's data sets. Here we study Monks-1 data set. The Monk's data set has 7 features: a1:$\{1, 2, 3\}$, a2:$\{1, 2, 3\}$, a3:$\{1, 2\}$, a4:$\{1, 2, 3\}$, a5:$\{1, 2, 3, 4\}$, a6:$\{1, 2\}$, and Id. studying the data set, confirms Id as a unique symbol for each observation. Considering its unique property, it is removed from the final data set in our analysis.

To find the separating plane with features with big main effects and ones included in interaction effect, we converted categorical features in to a binary data set by introducing dummy variables. After transforming data set in to the binary using dummy variables, we have a data set with 15 features and 324 observations in the train subset and 108 observations in the test subset.

We conducted a sensitivity analysis on the value of the α to find the best separating plane while regularizing our classification models. For this reason, we solved Bradley et al. (1999) Model and Model $\mathcal{M}(X, y)$ for different values of α equal to $0, 0.25, 0.75$ and 1. We selected these values to investigate how emphasizing more on minimizing number of missclassifications in comparison with generalization of separating plane by removing features with small coefficients would affect the performance of our model. And also in situation when more emphasize is on the generalization feature selection plane how our model would react. As already discussed in Section 3.2.2, the performance of the linearized Model $\mathcal{M}(X, y)$ depends on the values of the lower bound and upper bound of the magnitude of the interaction effect, b^k. Usually knowledge of data sets helps to estimate proper values for this parameters. However, for these data sets we do not have clear understanding about the possible lower bound and upper bound of b^k. As a result, we developed a new heuristic method to estimate these values. In this method, using Bradley et al. (1999) Model, we find the best separating plane for the categorical data set and considering the selected features, we set the lower bound and upper bound of b^k equal to the 5 times of the minimum and maximum value of their coefficients in each scenario. Another parameter that is needed for linearization of complimentarity costraints in our Model is M, which is the maximum distance of a classified points from the selected plane. For this data set, we investigate how M equals to 10 versus 100, or 1000 would affect the performance of our model. The new model proposed in Section 3.2.2 is implemented in Python (R Core Team (2014)), Pyomo (Hart et al. (2011)), using CPLEX as the MILP solver for Model $\mathcal{M}(X, y)$. The Bradley et al. (1999) Model also is implemented in Python for comparison.

First we start with Bradley et al. (1999) Model to find the best value of M. Table 3.4 shows the output of their Model for different values of M and α. Separating plane, number of missclassifications in both train and test subsets along with computational time is recorded. As expected, different values of α and M lead to different separating planes with different number of missclassifications. Result shows Bradley et al. (1999) Model acts robust in term of M for the total number of missclassifications in the test subset, while we observe more missclassified points in the train data set when $M = 1000$. Consequently M equal to 10 or 100 are the better selections. Studying the computational time also suggests $M = 10$ to be a better choice leading to the same result with less computational effort. As a result, we select $M = 10$ as the best value of M to propose our comparison for the two feature selection models.

We compare our new feature selection model with Bradley et al. (1999) Model with consideration of possibility of having two interaction effects. Running Model $\mathcal{M}(X, y)$ on Monk's data set for two interaction effects, shows a significant increase in the computational difficulty. For instance, for $\alpha = 0.75$, Model $\mathcal{M}(X, y)$ was ran for 42756.60 seconds and still we were %9.52 away from the optimal solution. To overcome this issue, we developed a new heuristic algorithm that is built on the repetitive implementation of Model $\mathcal{M}(X, y)$ for finding the separating plane with consideration of multiple interaction effects. The algorithm works such that in each round, we always look for one interaction effect. The detected interaction effect will be added to the data set as a new feature, which represents the presence or absence of the interaction effect. The updated data set will be given to the Model $\mathcal{M}(X, y)$ for detecting a second interaction effect. We continue this loop to reach the desired number of interaction effects. The algorithm is implemented to detect two interaction effect on the Monk's data set.

Table 3.5 compares the two Model with respect to their ability to find the separating plane that minimizes the number of missclassifications for all scenarios. For the scenario $\alpha = 0$, we specifically look for minimizing the total number of missclassifications. Both Models find the separating plane with 13 features that leads to 48/24 missclassified points in the training/test subset based on Bradley et al. (1999) Model and Model $\mathcal{M}(X, y)$. When $\alpha = 1$, we emphasis on the separating plane having as few feature as possible. Since we completely ignore minimizing the total number of missclassifications, it leads to the selection of a plane having all its features' coefficients equal to 0. Here all data pints are assigned to the same group $\mathcal{B}$ and we would have 165 data points missclassified in the train subset and 51 missclassifications in the test subset. For the scenarios when α is 0.25 or 0.75 , Model $\mathcal{M}(X, y)$ was able to find the interacting features and reduce the number of missclassifications in both train and test subsets. Compare to Bradley et al. (1999) Model, we observe %100 improvement on the number of missclassified points in the training subset and upto %45 decrease on the missclassifications in the test subset. As expected compare to Bradley et al. (1999) Model, the computatioanl difficulty increases using Model $\mathcal{M}(X, y)$ and interestingly increase in α leads to more computational effort. The result suggests that $\alpha = 0.25$ is the best scenario, which leads to the lowest possible number of missclassifications in both train and test subsets, while computationally is less time consuming compare to $\alpha = 0.75$.

Table 3.4: Result of Bradley et al. (1999) feature selection model for different values of M, Monks's Data set

M	α	Separating Plane	Train Missclassifications	Test Missclassifications	Computational Time(sec)
10	0	$-10x_3-8x_4-6x_5-8x_6-10x_7-6x_8-10x_9-10x_{10}-10x_{11}-10x_{12}-4x_{13}-4x_{14}-4x_{15}=-31$	48	24	39.43
	0.25	$2x_4+4x_5-2x_7+2x_8-6x_{12}=1$	48	24	350.11
	0.75	$-2x_4-8x_7-10x_{12}=-9$	51	21	163.74
	1	-	165	51	0.68
100	0	$-100x_3-98x_4-96x_5-98x_6-100x_7-96x_8-100x_9-100x_{10}-100x_{11}-100x_{12}-94x_{13}-94x_{14}-94x_{15}=-391$	48	24	111.18
	0.25	$-2x_3+2x_5-2x_6-4x_7-6x_{12}=-3$	48	24	526.75
	0.75	$-2x_4-98x_7-100x_{12}=-99$	51	21	385.6
	1	-	165	51	0.88
1000	0	$-1000x_3-998x_4-996x_5+998x_6+996x_7+1000x_8-1000x_9-1000x_{10}-1000x_{11}-100x_{12}-994x_{13}-994x_{14}-994x_{15}=-1995$	90	24	273.25
	0.25	$-996x_3-994x_4-2x_7+2x_8-998x_{12}=-995$	67	24	388.23
	0.75	$-2x_4-2x_7-4x_{12}=-3$	68	21	1494.43
	1	-	165	51	0.95

3.3.2.2 Lymphography Data set

The Lymphography (ly) data set is one of three domains provided by the Oncology Institute that has repeatedly appeared in the machine learning literature. This lymphography domain was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. It contains 148 instances with 18 nominal features. The task is to distinguish healthy patients from those with metastases or malignant lymphoma. Studying its features, we have 10 nominal with two variations, which can easily be transformed to binary. The remaining features are categorical with different levels, which are converted in to binary by introducing dummy variables. We filtered instances to distinguish metastases from malignant lymphoma patients. The new data set has 142 instances with 47 features. Data set is partitioned into %75 training and %25 test subsets.
To find the separating plane that minimizes the number of missclassifications, we conduct sensitivity

Table 3.5: Comparison of feature selection models, Monks's Data set

Model	α	Separating Plane	Train Missclassifications	Test Missclassifications	Computational Time(sec)
Bradley et. al.	0	$-10x_3 - 8x_4 - 6x_5 - 8x_6 - 10x_7 - 6x_8 - 10x_9 - 10x_{10} - 10x_{11} - 10x_{12} - 4x_{13} - 4x_{14} - 4x_{15} = -31$	48	24	39.43
	0.25	$2x_4+4x_5-2x_7+2x_8-6x_{12} = 1$	48	24	350.11
	0.75	$-2x_4 - 8x_7 - 10x_{12} = -9$	51	21	163.74
	1	-	165	51	0.68
$\mathcal{M}(X, y)$	0	$-10x_3 - 8x_4 - 6x_5 - 8x_6 - 10x_7 - 6x_8 - 10x_9 - 10x_{10} - 10x_{11}+4x_{12}+10x_{13}+10x_{14}+10x_{15} = -17$	48	24	39.23
	0.25	$-4x_3-2x_4+2x_6+4x_8-6x_{12}-6x_5x_8(1 - x_{12}) + 4x_3x_7x_{12} = -1$	0	13	1204.21
	0.75	$-2x_4 - 2x_7 - 410x_{12} - 4x_5x_8(1-x_{12})(1-x_{13}) = -3$	33	15	2586.81
	1	-	165	51	3.97

analysis on the values of α equal to $0, 0.25, 0.75$, and 1. Same as Monk's data set, we have no clear understanding about the possible lower bound and upper bound of b^k. Therefore, we follow the same heuristic algorithm that is already discussed in Section 3.3.2.1 to find these values. Although we already discussed how selection of M could affect on the effectiveness of both Model $\mathcal{M}(X, y)$ and Bradley et al. (1999) Model, here we will not implement the same analysis we presented in Section 3.3.2.1 to find the best M. The purpose of this paper is presenting a new feature selection model using mathematical modeling that separates data points in two different classes while considering interaction effect and also demonstrating a substantial improvement over Bradley et al. (1999) Model. Therefore, for any selection of M, this comparison could be applicable. Here we fix M to be equal to 10.

The comparison between two feature selection models with possibility of two interaction effects was investigated. Result detects the same interacting features with same magnitude of interaction effects for possibilities of two interaction effects. This redundancy lead us to the conclusion that data set only has one interaction effect. Result of Bradley et al. (1999) Model and Model $\mathcal{M}(X, y)$ for all values of α are presented in Table 3.6.

Number of selected features, complexity of interaction effect, number of missclassified data points in train and test subsets, along with computational time for both models are shown in Table 3.6. For both $\alpha = 0$ and $\alpha = 1$, we observe the same behaviour from both Bradley et al. (1999)

Table 3.6: Comparison of feature selection models, Lymphography Data set

Model	α	Number of Selected Features	Number of Interacting features	Train Miss-classifications	Test Missclas-sifications	Computational Time(sec)
Bradley et. al.	0	47	-	0	9	3.12
	0.25	13	-	0	9	44.12
	0.75	13	-	0	9	44.06
	1	0	-	49	12	0.07
$\mathcal{M}(X, y)$	0	47	0	0	12	0.24
	0.25	10	11	0	7	1614.13
	0.75	4	6	9	7	6.31
	1	0	-	49	12	0.1

Model and Model $\mathcal{M}(X, y)$. The only difference is the test missclassifications and computational time. Because of estimating lower bound and upper bound of b^k based on the heuristic algorithm we already discussed, we allow b^k having values different than 0, and due to including θ^l in the constraints and the value of M, θ^l turns out to get a value, which leads to an increase in the number of missclassifications in the test subset. A better estimation of mentioned parameters, would avoid the issue. For the scenario when $\alpha = 0.25$, Bradley et al. (1999) Model selects a separating plane having 13 features with total number of missclassifications for the test data set equal to 9 and we have no missclassified points in the train subset. However, Model $\mathcal{M}(X, y)$ found a different separating plane with 10 features and 1 interaction effect such that it reduces the number of missclassifications to 7 in the test subset, while keeping no missclassified point in the train subset. When $\alpha = 0.75$, we observe a separating plane resulted from Model $\mathcal{M}(X, y)$ that not only has fewer number of missclassifications in the test subset, but also has fewer selected features in the plane (4 features). Although it shows 9 missclassifications in the train subset, due to generalization format of the separating plane, it leads to fewer number of missclassified test data points. Both $\alpha = 0.25$ and $\alpha = 0.75$ shows an improvement on classifying data points by reducing the number of missclassifications up to %22.22. Finally consideration of computational time, would suggest $\alpha = 0.75$ as the best scenario which reduces the number of missclassifications while computationally acts efficient. These results suggest that the Model $\mathcal{M}(X, y)$ demonstrates substantial improvements over Bradley et al. (1999) Model in the presence of interacting features.

3.3.3 Conclusions and Future Work

The main goal of feature selection is to find a feature subset that is highly predictive of the response variable. While well-known feature selection methods, emphasize on the removal of apparently redundant and irrelevant features to avoid poor predictive models, we showed redundant and apparently irrelevant features could be useful for prediction. We proposed a new feature selection model that emphasizes on the importance of considering redundant or apparently irrelevant features that are associated in feature interaction. The model is an advanced version of the feature selection model Bradley et al. (1999) proposed. Using mathematical modeling, we tried to find a separating plane to categorize data points in two different sets of $\mathcal{A}$ and $\mathcal{B}$ while considering interaction effect. Our model is specifically designed for categorical data sets with binary features. It can also be used on categorical data sets with Continuous features, by transforming features in to different categories. Sensitivity analysis revealed that the performance of our new feature selection model depends on our selection of α and lower bound and upper bound of b^l. The effectiveness of the new feature selection model is evaluated on the synthetics and real world data sets. The proposed method is compared with Bradley et al. (1999) feature selection model in terms of ability to minimize number of misclassifications and computation time. Interestingly based on literature no feature selection method could detect explicitly interacting features and their combination, while our model can specifically tells which features are interacting with each others. The proposed feature selection model has several limitations: The validity of the detected interaction effect depends on two assumptions: (1) the lower and upper bound of the magnitude of interaction effect, and (2) number of interaction effects.

3.4 References

Abe, N. and Kudo, M. (2006). Non-parametric classifier-independent feature selection. *Pattern recognition*, 39(5):737–746.

Appice, A., Ceci, M., Rawles, S., and Flach, P. (2004). Redundant feature elimination for multi-class problems. In *Proceedings of the twenty-first international conference on Machine learning*, page 5. ACM.

Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271.

Bradley, P. S., Fayyad, U. M., and Mangasarian, O. L. (1999). Mathematical programming for data mining: Formulations and challenges. *INFORMS Journal on Computing*, 11(3):217–238.

Bradley, P. S., Mangasarian, O. L., and Street, W. N. (1998). Feature selection via mathematical programming. *INFORMS Journal on Computing*, 10(2):209–217.

Collins, R. L., Hu, T., and et al., C. W. (2013). Multifactor dimensionality reduction reveals a three-locus epistatic interaction associated with susceptibility to pulmonary tuberculosis. *BioData Mining*, 6:1.

Cotter, S. F., Kreutz-Delgado, K., and Rao, B. D. (2001). Backward sequential elimination for sparse vector subset selection. *Signal Processing*, 81(9):1849–1864.

Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. In *Icml*, volume 1, pages 74–81.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, 1(1-4):131–156.

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

Evans, D. M., Marchini, J., Morris, A. P., and Cardon, L. R. (2006). Two-stage two-locus models in genome-wide association. *PLoS Genetics*, 2:e157.

Gennari, J. H., Langley, P., and Fisher, D. (1989). Models of incremental concept formation. *Artificial intelligence*, 40(1-3):11–61.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422.

Hart, W. E., Watson, J.-P., and Woodruff, D. L. (2011). Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3(3):219–260.

John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier.

Kässens, J. C., Wienbrandt, L., González-Domínguez, J., Schmidt, B., and Schimmler, M. (2002). High-speed exhaustive 3-locus interaction epistasis analysis on FPGAs. *Journal of Computational Science*, 9:131–136.

Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In *Machine Learning Proceedings 1992*, pages 249–256. Elsevier.

Koller, D. and Sahami, M. (1996). Toward optimal feature selection. Technical report, Stanford InfoLab.

Leem, S., Jeong, H. H., Lee, J., Weea, K., and Sohn, K. (2014). Fast detection of high–order epistatic interactions in genome–wide association studies using information theoretic measure. *Computational Biology and Chemistry*, 50:19–28.

Maher, B. (2008). The case of the missing heritability. *Nature*, 456:18–21.

Maldonado, S. and Weber, R. (2009). A wrapper method for feature selection using support vector machines. *Information Sciences*, 179(13):2208–2217.

Mangasarian, O. L. (1996). Mathematical programming in machine learning. In *Nonlinear Optimization and Applications*, pages 283–295. Springer.

Moore, J. H., Asselbergs, F. W., and Williams, S. M. (2010). Bioinformatics challenges for genome-wide association studies. *Bioinformatics*, 26:445–455.

R Core Team (2014). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Radha, R. and Muralidhara, S. (2016). Removal of redundant and irrelevant data from training datasets using speedy feature selection method. *Intâl J Comp. Sci. and Mob. Comput.*, 5(7):359–364.

Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F., and Moore, J. H. (2001). Multifactor–dimensionality reduction reveals high–order interactions among estrogen–metabolism genes in sporadic breast cancer. *The American Journal of Human Genetics*, 69:138–147.

Santana, L. E. A. d. S. and de Paula Canuto, A. M. (2014). Filter-based optimization techniques for selection of feature subsets in ensemble systems. *Expert Systems with Applications*, 41(4):1622–1631.

Tang, X., Dai, Y., and Xiang, Y. (2018). Feature selection based on feature interactions with application to text categorization. *Expert Systems with Applications*, 120.

Upstill-Goddard, R., Eccles, D., Fliege, J., and Collins, A. (2013). Machine learning approaches for the discovery of gene–gene interactions in disease data. *Briefings in bioinformatics*, 14(2):251–260.

Wang, G., Song, Q., Xu, B., and Zhou, Y. (2013). Selecting feature subset for high dimensional data via the propositional foil rules. *Pattern Recognition*, 46(1):199–214.

Wang, L. and Mehr, M. N. (2019). An optimization approach to epistasis detection. *European Journal of Operational Research*, 274(3):1069–1076.

Wei, H.-L. and Billings, S. A. (2006). Feature subset selection and ranking for data dimensionality reduction. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):162–166.

Xie, M., Li, J., and Jiang, T. (2012). Detecting genome-wide epistases based on the clustering of relatively frequent items. *Bioinformatics*, 28:5–12.

Yu, L. and Liu, H. Efficient feature selection via analysis of relevance and redundancy.

Zeng, Z., Zhang, H., Zhang, R., and Yin, C. (2015). A novel feature selection method considering feature interaction. *Pattern Recognition*, 48(8):2656–2666.

Zhao, Z. and Liu, H. (2009). Searching for interacting features in subset selection. *Intelligent Data Analysis*, 13(2):207–228.

Zhu, Z., Ong, Y.-S., and Dash, M. (2007). Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognition*, 40(11):3236–3248.

SUMMARY OF CONTRIBUTIONS

In this dissertation, we proposed novel feature selection models for response prediction with consideration of interacting features. In the first paper we addressed a novel application of operations research in detecting epistatic effect. In the paper, we addressed a Mixed Integer Linear Programming model that detects both the complexity of the epistasis and the exact combination of genes the triggers the effect. Sensitivity analysis revealed that model is most effective and computationally efficient when we have a few dozen genes and a couple of hundreds observations. We also developed a local heuristics search to detect the epistasis for a large data set. In presence of large number of features, first a feature selection algorithm can be used to reduce the dimensionality, and then small sample of observations are iteratively drawn to feed the optimization model. We also conducted experimental studies on a soybean data set with 20,087 genes and 42,509 individuals. In comparison with popular algorithms in literature, the new algorithm demonstrated a significant improvement in detecting epistasis effects. In the second paper, we proposed a novel feature selection algorithm for data sets with continuous response predictions that detects interactive features. While common well-known feature selection algorithm mostly are structured based on the removal of redundant and apparently irrelevant features, we show redundant and apparently irrelevant features might be important in subset selections. Considering interaction effect, a feature could be redundant or apparently irrelevant by itself, while it could be important in the presence of others. We showed unintentional removal of these features could lead to poor predictions. Proposed method finds features with significant main effect using well-known feature selection methods and implements a mixed integer linear programming model to recognize feature interactions and their exact combinations. Therefore, we modified the MILP modeling introduced in Chapter 1, to be able to detect multiple interaction effects and conducted simulation experiments on synthetic and real data sets. We proposed a comparison between the result of the new algorithm, FSCI, with three existing feature selection methods (FCBF, ReliefF and Random forest) and showed how effective it finds interacting features. We also demonstrated the prediction improvement we observed in synthetic and real data sets by implementing the FSCI algorithm. The third paper discussed a new feature selection model for datasets with binary response variables. The new feature selection model emphasizes on the importance of considering redundant or apparently irrelevant features that are associated in feature interaction. Using mathematical modeling, we tried to find a separating plane to categorize data points in two different sets of $\mathcal{A}$ and $\mathcal{B}$ while considering interaction effect. Our model is specifically designed for categorical datasets with binary features. It can also be used on categorical datasets with Continuous features, by transforming features in to different categories. Study confirms the performance of new feature selection model depends on the value of α. The effectiveness of the new feature selection model is evaluated on the synthetics and real world data sets. The proposed method is compared with Bradley et al. (1999) feature selection model in terms of ability to minimize number of misclassifications and computation time. Interestingly based on literature no feature selection method could detect explicitly interacting features and their combination, while our model can specifically tells which features are interacting with each others. The proposed feature selection models have limitations: the validity of the detected interaction effect depends on two assumptions: (1) the lower and upper bound of the magnitude of interaction effect, and (2) number of interaction effects.