5-2023

Distance Correlation Based Feature Selection in Random Forest

Jose Munoz-Lopez
California State University - San Bernardino

Distance Correlation Based Feature Selection in Random Forest

---

A Thesis

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Arts

in

Mathematics

---

by

Jose Muñoz-Lopez

May 2023

Distance Correlation Based Feature Selection in Random Forest

---

A Thesis

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Jose Muñoz-Lopez

May 2023

Approved by:

Suthakaran Ratnasingam, Committee Chair

Youngsu Kim, Committee Member

Hajrudin Fejzic, Committee Member

Dr. Madeleine Jetter, Chair, Department of Mathematics

Dr. Corey Dunn, Graduate Coordinator

Abstract

The Pearson correlation coefficient is a commonly used measure of correlation, but it has limitations as it only measures the linear relationship between two numerical variables. In 2007, Székely et al. [SRB07] introduced the distance correlation, which measures all types of dependencies between random vectors X and Y in arbitrary dimensions, not just the linear ones. In this thesis, we propose a filter method that utilizes distance correlation as a criterion for feature selection in Random Forest regression. We conduct extensive simulation studies to evaluate its performance compared to existing methods under various data settings, in terms of the prediction mean squared error. The results show that our proposed method is competitive with existing methods and outperforms all other methods in high-dimensional ($p \geq 300$) nonlinearly related data sets. The applicability of the proposed method is also illustrated by two real data applications.

Acknowledgements

Writing this thesis has been a long and challenging journey, and I could not have done it without the support and encouragement of so many people.

First and foremost, I would like to thank my thesis advisor, Suthakaran Ratnasingam, for his guidance, patience, and expertise throughout the research and writing process. I'm grateful for the feedback and insights that were invaluable in the shaping of this research. I am also grateful to my committee members, Youngsu Kim and Hajrudin Fejzic, for their helpful comments and suggestions.

Finally, I would like to extend my heartfelt thanks to my family and friends for their unwavering support and understanding throughout this journey. Their love and encouragement have been my constant source of strength. Thank you all for being a part of this journey with me!

Table of Contents

Abstract **iii**

Acknowledgements **iv**

List of Figures **vi**

1 Introduction **1**
1.1 Feature Selection . . . 3
1.1.1 Filter Method . . . 3
1.1.2 Wrapper Method . . . 4
1.1.3 Hybrid Method . . . 5
1.2 Random Forests . . . 5
1.3 Reinforcement Learning Trees . . . 11
1.4 Motivation . . . 12

2 Distance Correlation Based Feature Selection **13**
2.1 Distance Correlation (DC) . . . 13
2.2 Feature Selection Method in Random Forest . . . 15
2.3 Theoretical Results . . . 16

3 Simulation Study **18**
3.1 Analysis of the Linear Models . . . 19
3.2 Analysis of the Nonlinear Model . . . 25

4 Real Data Applications **28**
4.1 Riboflavin Data . . . 29
4.2 Boston Housing Data . . . 32

5 Conclusion and Discussion **35**
5.1 Performance of the Proposed DC-Based Filter Method . . . 35
5.2 Future Work . . . 36

Bibliography **37**

List of Figures

1.1 Filter Method 4
1.2 Wrapper Method 5
1.3 Classification Tree 7
1.4 Regression Tree 8
1.5 Possible Sample Data for Regression Tree 8

3.1 Prediction MSE Comparison for Setting 1 ($\rho = 0.5, 0.8$) and Setting 2 with $\rho = 0.5$ 24
3.2 Prediction MSE Comparison for Model 2 & 3 27

4.1 Boxplot for Prediction MSE Comparison for Riboflavin Data 30
4.2 Prediction MSE Comparison for Riboflavin Data for CC and DC-based Methods 31
4.3 Prediction MSE Comparison for Boston Housing Data for CC and DC-based Methods 33
4.4 Boxplot of Prediction MSE for Boston Housing Data 34

Chapter 1

Introduction

Feature selection is a crucial aspect of model construction in machine learning. Its main objective is to identify the most significant features and rule out less significant ones. This process involves selecting a subset of the feature space. Feature selection is widely used for various reasons, including enhancing model interpretability, reducing learning time, improving learning accuracy, and overcoming the curse of dimensionality, among other things. Feature selection is widely employed in many fields, particularly in classification tasks like bioinformatics data analysis, image recognition, change point detection, and others.

Several feature selection methods have been proposed in the literature, for example, AIC and BIC criteria are used to identify the 'best model'. One popular method is the Lasso, which was introduced by [Tib96] and employs ℓ_1 regularized linear regression model. Other Lasso-based feature selection methods have been developed since then, such as Adaptive Lasso ([Zou06]), Lars ([EHJT04]), and elastic net ([ZH05]), among others. However, when dealing with high-dimensional data, Lasso methods can face two significant problems: high computational cost and over-fitting. The correlation coefficient (CC) is a criterion, introduced by [Pea96], utilized in feature selection for multiple machine learning algorithms. The CC is represented by the symbol ρ when describing a population. However, when referring to a sample, it is usually denoted as r or $r_{x,y}$. Suppose a sample size n and vectors $X = \{x_1, ..., x_n\}$, and $Y = \{y_1, ..., y_n\}$, with sample means given by $\overline{x}$ and $\overline{y}$ respectively, and the sample standard deviations are S_x for X and S_y for Y. Then, sample CC is defined as

$$r_{x,y} = \frac{\text{Cov}(X,Y)}{S_x S_y},$$

where the sample covariance is,

$$\text{Cov}(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}), \tag{1.1}$$

and,

$$S_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}, \quad S_y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2}.$$

The CC values can range from -1 to 1, and a CC of -1 or $+1$ indicates a perfect linear relationship. In particular, the stronger the correlation, the closer the CC comes to ± 1.

Consider a set of p features, $\mathbf{X} = (X_1, \ldots, X_p)$, and the dependent variable Y. The goal is to estimate the regression function $f(x) = E(Y|X = x)$ and we assume that $Y = f(x) + \epsilon$. We observe a sample of independent and identically distributed (i.i.d.) training observations $\mathcal{D}_n = \big\{(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots, (\mathbf{X}_n, Y_n)\big\}$, where each $\mathbf{X}_i = \big(X_{i1}, \ldots, X_{ip}\big)^\top$ denotes a set of p variables from a feature space $\mathcal{X}$. Let ϵ_i's be i.i.d. with mean 0 and variance σ^2 and p^* refers to the chosen features after removing the ones that have less correlation with the response. The remaining $p - p^*$ variables have no influence on the response. We also assume that the expected value $E(Y|\mathbf{X}^*)$ is completely determined by a set of $p^* < p$ variables, which means $E(Y|\mathbf{X}^*) = E(Y|X_1, X_2, \ldots, X_{p^*})$.

[CLWY18] used the CC, amongst other measures, for feature selection in high dimensional data analysis. [HH$^+$10] made improvements to their models using the CC as well as a clustering technique to filter out less important parameters. We even see [LMC$^+$20] use the CC for detecting daily activities in smart homes, where models rely heavily on selecting the appropriate features for these daily activities, and thus on feature selection. In the study by [Won19], the CC was used as a criterion to identify features that displayed a high correlation with the response. The approach was applied to enhance random forest regression models.

Despite its usefulness, the CC has some limitations. Firstly, it only measures the linear relationships between two random variables, X and Y. Additionally, $\rho = 0$ only implies independence if X and Y have a bivariate normal joint distribution. To

remedy the shortcomings of the Pearson CC, [SRB07] introduced the distance correlation (DC), which can measure all types of dependence between two random vectors, X and Y, in any number of dimensions. In their work, [LB16] opted to use DC over CC as a feature selection measure due to its advantages. Leger provided an illustrative example, where X is a uniform random variable in $[-1, 1]$ and Y is expressed as $Y = e^{-10X^2}$. For 1000 random samples the CC is 0.02, whereas the DC is 0.50. We will explore distance correlation in more detail later. However, it is worth noting that the definition of DC is similar (at least symbolically) to that of CC, as we will see that DC is defined as the distance covariance of X and Y divided by the square root of the distance variance of X multiplied by the distance variance of Y.

1.1 Feature Selection

There are several techniques proposed in the literature to evaluate feature subsets in machine learning. The filter method, as presented by [Hal00] and [DCSL02] utilizes the intrinsic properties of data to assess feature subsets. The wrapper method, as discussed by [CF94] and [DB00] determines the best subset of features useful for the task based on the performance of the learning algorithm. Finally, the hybrid approach, as described by [Ng98],[Das01] and [XJK01], makes use of both filters and wrappers by utilizing independent criteria and learning algorithms to measure feature subsets.

1.1.1 Filter Method

A filter method assesses feature relevance from the intrinsic properties of the data. Features are typically ranked on some feature relevance score and low scoring features are removed from the feature space. Only the remaining features are then used in the classification algorithm. In essence 'filtering out' the features that do not help the classification algorithm sufficiently. Figure 1.1 below is an illustration of the filter method.

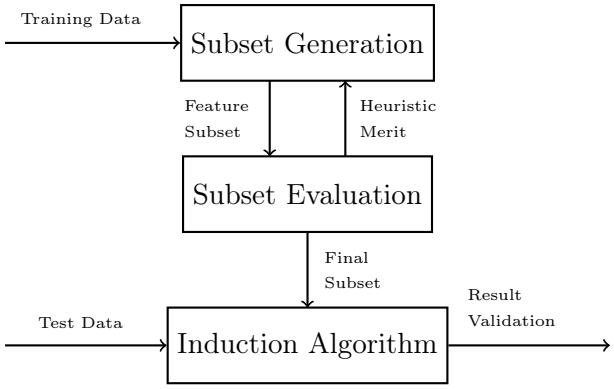

Figure 1.1: Filter Method

The techniques, although quick and scale-able, may ignore feature dependencies which may lead to worse classification performance [Pap13]. We give an example of a feature relevance definition.

Let F be the full set of features and C be the target class. Let $F_i \in F$ and $S_i = F - F_i$.

Definition 1.1 *(Irrelevance) A feature F_i is irrelevant if and only if:*

$$\boldsymbol{P}(C|F_i, S_i') = \boldsymbol{P}(C|S_i'), \quad \forall S_i' \subseteq S_i.$$

Irrelevance of a feature means that the feature is not necessary for the classification since the class distribution from any subset of other features does not change after eliminating the irrelevant feature. Relevance is not as straightforward a definition. [KJ97], however, give definitions for a strong and a weak relevance of features with a Bayes classifier. Essentially the strong relevance of a feature implies that a feature is required for an optimal set, while weak relevance implies that the feature may be required in some cases to improve the prediction.

1.1.2 Wrapper Method

Filter methods perform the search for an optimal feature subset independently of the classifier building step, while wrapper methods do not. Wrapper methods integrate the classifier hypothesis search within the feature subset search. This integration can help to identify interactions between the feature subset search and model selection that

other methods may not find. However, the drawbacks of the wrapper method include computational cost and a higher risk of overfitting. One popular example of this method in action would be genetic algorithms [Dav91]. Figure 1.2 below depicts the wrapper method.

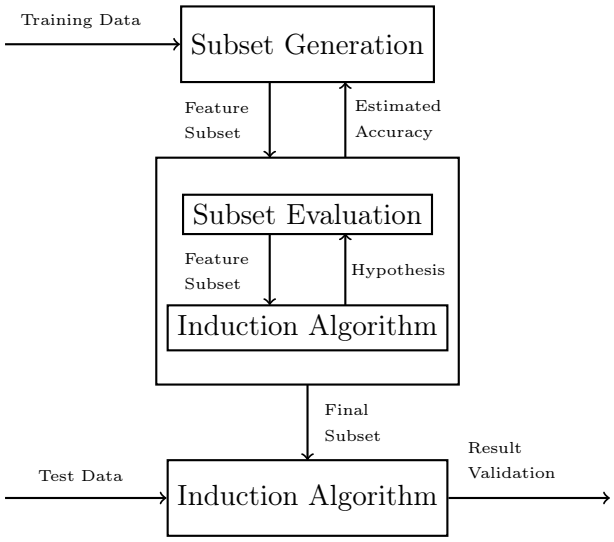

Figure 1.2: Wrapper Method

1.1.3 Hybrid Method

The hybrid method combines filter and wrapper methods. In other words, it integrates the search for an optimal subset of features into the classifier construction. This search occurs in the combined space of feature subsets and hypotheses. Hybrid methods share similarities with wrapper methods but are less computationally expensive [SIL07]. Examples of hybrid or embedded methods include support vector machine [GWBV02] and logistic regression [MH05].

1.2 Random Forests

Random Forest (RF) is an *ensemble learning algorithm* proposed by Breiman ([Bre01]) in 2001 that is widely used for both classification and regression tasks.

Aside 1.2 *An ensemble learning algorithm is a algorithm that uses many models and aggregates the models for better results.*

RF constructs a large number of decision trees during the training process and outputs the mean prediction of the individual trees. During the training process, each decision tree is grown using a randomly selected subset of the input features and a random subset of the training data. This randomness helps to reduce the overfitting problem commonly encountered in decision trees. At test time, the RF aggregates the predictions of all decision trees to provide the final prediction. RF is known for its high accuracy, robustness to noise and outliers, and scalability to large datasets. It has been widely used in various fields, including remote sensing, finance, bioinformatics, and image processing. For example, [HCCG05] used RF to classify hyperspectral data. Previously the data presented many challenges to classification algorithms as it is high dimension data with classes that are sometimes quite mixed. RF proved to be an improvement to the classification of NASA's hyperspectral data. We also see many biomedical applications such as in genomics. We see [ALX$^+$20] use RF for biomarker identification, which is one of the major goals in functional genomics.

As mentioned earlier, RF utilizes decision trees and can perform both classification and regression analyses. They achieve this by using a combination of the bootstrap aggregation method and the random subspace method to generate a collection of decision trees, which are then utilized for classification purposes. When building an RF, the best predictor from a randomly chosen subset of predictors is used to divide each node. Although this method may seem counterintuitive, it has proven to be more effective than other classifiers such as discriminant analysis, support vector machines, and neural networks. Additionally, [Bre01] showed that this approach is resistant to overfitting. Below we can see the original algorithm used by Breiman.

Algorithm: Breiman's Random Forest

Given the training data set $\mathcal{D}_n$,

1. Generate B bootstrap samples of size n from the training data.
2. For each b-th bootstrap sample, $b = 1, 2, ..., B$, grow a tree $\hat{f}_b$ by recursively repeating the following steps for each terminal node of the tree until the minimum node size n_{min} is reached;
   (a) Select m features at random out of the p feature variables.

(b) Choose the best-split feature and split among the m features.

(c) Split the node into 2 daughter nodes.

3. Output the ensemble of trees $\hat{f}_1, \hat{f}_2, ..., \hat{f}_B$ and the predicted value is obtained by combining the ensemble of trees.

To make a prediction at a new point x_0,

$$\begin{cases} \dfrac{1}{B}\displaystyle\sum_{b=1}^{B} \hat{f}_b(x_0) & \text{for regression problems} \\ \text{majority voting } \left\{\hat{f}_b(x_0)\right\}_1^B & \text{for classification problems} \end{cases}$$

An example of a classification tree can be seen below. Notice the three discrete classifications: *'eat a meal'*,*'eat a snack'*, and *'don't eat'*.

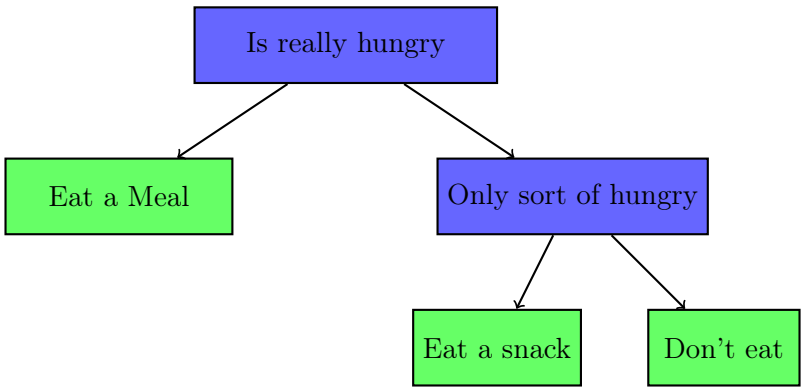

Figure 1.3: Classification Tree

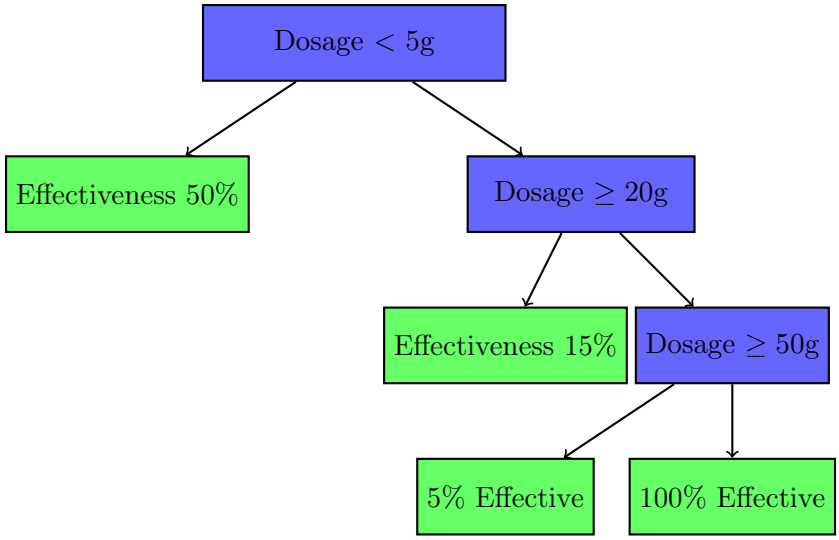

Figure 1.4: Regression Tree

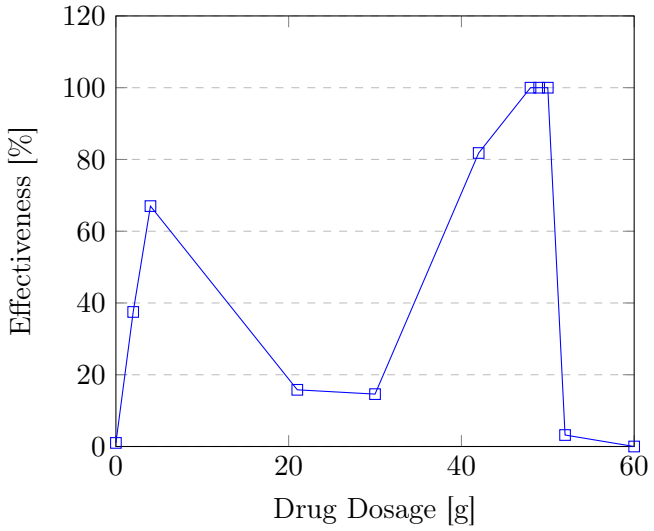

Figure 1.5: Possible Sample Data for Regression Tree

When discussing RF, it is important to mention classification and regression trees (CARTs). While classification trees predict outcomes to be in some discrete class, regression trees predict continuous outcomes to be some real number. Examples of potential classification and regression trees are shown in Figures 1.3 and 1.4, where a left arrow implies true and a right arrow implies false. In this case, a regression tree is used

to predict the effectiveness of drug dosage. The tree shows that the effectiveness of the drug dips around the 20g dosage and becomes super effective at 50g. This is a scenario where a linear regression fit cannot be used to predict effectiveness. However, the main issue with CARTs is overfitting, where the model performs well with the training data but poorly with new data. To address this issue, RFs use multiple trees and aggregate their predictions, as using just one decision tree would lead to overfitting of the training data and large errors with the test data.

However, a crucial aspect of Breiman's RF algorithm that we have yet to discuss is step 2b, which involves selecting the optimal split feature and splitting among the m features. While it is simple to randomly select m features out of p features in the previous step, determining the best-split feature among these randomly selected features is more complex. The choice of metric for selecting the best-split feature depends on whether the random forest is being used for classification or regression tasks, as certain metrics may be better suited for one or the other. For classification trees, the Gini impurity is a suitable metric for selecting the best node. The Gini impurity measures the effectiveness of a node split by counting the number of misclassified data points at that split over the total number of data points. In other words, it calculates the probability that a randomly chosen data point would be incorrectly labeled due to choosing this node split. By computing the Gini impurity for each feature and selecting the feature with the smallest Gini impurity, we can ensure that the best split is made for classification trees.

More precisely the Gini impurity can be defined as follows. In a classification problem with J classes and relative frequencies of these classes denoted by p_i, where $i \in \{1, 2, ..., J\}$, the probability of selecting an item with label i is p_i, and the probability of misclassification is $\sum_{k \neq i} p_k = 1 - p_i$. The Gini impurity ($I_G$) is calculated by summing the pairwise products of these two probabilities. We see below that the Gini impurity can be calculated by summing the relative frequencies squared and subtracting from 1.

$$\begin{aligned}\sum_{i=1}^{J}\left(p_i\sum_{k\neq i}p_k\right) &= \sum_{i=1}^{J}(p_i(1-p_i))\\ &= \sum_{i=1}^{J}(p_i-p_i^2)\\ &= \sum_{i=1}^{J}p_i-\sum_{i=1}^{J}p_i^2\\ &= 1-\sum_{i=1}^{J}p_i^2.\end{aligned}$$

In the case of regression trees, determining the *best* split is a matter of minimizing the sum of squared residuals. In [Won19], a widely-used algorithm for minimizing the mean squared error of all decision trees in RF is described as follows:

Algorithm: Variable Importance

---

1. For the b-th tree, $b = 1, 2, ..., B$, do the following.
   (a) Pass the out-of-bag (OOB) samples, that is the observations that are not selected in the b-th bootstrap samples down the tree and compute the mean squared error.
   (b) For each feature X_j, $j = 1, 2, ..., p$, randomly permute the values for X_j in the OOB sample, keeping all other features intact.
   (c) Pass the permuted OOB samples down the tree and recompute the mean squared error.
2. Compute the average increase in the mean squared error as a result of the randomly permuted values across all the trees.

While the Breiman RF is useful, it is not without limitations. According to [HTF01], when a data set has a small number of relevant features and a large number of irrelevant features, RF algorithms may struggle to achieve their intended predictive performance, particularly if the algorithm only selects a few features at each node. To address these limitations, several methods have been proposed in the literature to improve

the performance of Breiman's traditional RF. To address these limitations, [BDL08] establish the consistency of a special type of purely RF model where strong variables have a larger probability of selection as a splitting variable.

1.3 Reinforcement Learning Trees

Another approach to improving the Breiman random forest was proposed by [ZZK15]. They suggested a modification called Reinforcement Learning Trees (RLT), which also utilizes bootstrapped samples and ensemble methods to construct decision trees. However, the key difference lies in the process of internal node splitting. RLT uses an embedded model at each internal node to identify relevant features for splitting, muting noise variables, and prioritizing strong variables in the initial stages of tree construction, gradually decreasing the number of candidate variables towards the terminal nodes. This technique improves the selection of splitting variables, as it allows for muting variables based on relevance assessment at each internal node split. Let $VI_A(j)$ be variable importance measure for each variable $j \in \mathcal{P}$ at an internal node A. At each node, RLT constructs RF and uses it to compute the estimate of the variable importance, $\widehat{VI}_A(j)$, for every variable $j \in \mathcal{P}$ at node A. The algorithm for RLT is shown below, which is similar to Breiman's algorithm except for the node-splitting process.

Algorithm: Reinforcement Learning Trees (RLT)

Given a training data set $\mathcal{D}_n$,

1. Generate B bootstrap samples from $\mathcal{D}_n$,
2. For each b-th bootstrap sample, $b = 1, 2, ..., B$, fit a RLT model $\widehat{f}_b$ using the following:
   (a) Construct an embedded model $\widehat{f}_A^*$ to the training data in internal node A and this is done using only set of features $\{1, 2, ..., p\} \setminus P_A^d$, that is, $\mathcal{P} \setminus \mathcal{P}_A^d$, where $\mathcal{P}_A^d$ is the set of variables that are muted at node A and $\mathcal{P} = \{1, 2, \ldots, p\}$.
   (b) Compute the variable importance $\widehat{VI}_A(j)$ using the fitted embedded model $\widehat{f}_A^*$ for each variable X_j, where $j \in \mathcal{P}$.
   (c) The internal node is then split into 2 daughter nodes either using a one-dimensional split or a high dimensional split. Details are given in [ZZK15].

(d) The set of muted variable $\mathcal{P}^d$ is updated for the 2 daughter nodes by including the features with the smallest variable importance measures at the current node.

(e) Continue steps (a)-(d) on each daughter node until the minimum node size, n_{min} is reached.

3. Aggregate all B trees to obtain a final model, thus

$$\hat{f} = \begin{cases} B^{-1} \sum_{b=1}^{B} \hat{f}_b & \text{for regression problems} \\ I\left(0.5 < B^{-1} \sum_{b=1}^{B} \hat{f}_b\right) & \text{for classification problems} \end{cases}$$

1.4 Motivation

In their study, [Won19] explored the use of RF algorithms for regression problems by focusing on selecting significant features that have a strong correlation with the response variable. They utilized the filter method to find the optimal feature subset, and we are interested in applying a similar filter method using the distance correlation (DC) instead of their CC method. We anticipate that using the DC as a criterion for our filter method may provide advantages over the CC method, particularly in detecting nonlinear relationships in sample data. Additionally, we plan to evaluate the performance of our method against more sophisticated techniques, such as the RLT method. We expect our results to be comparable to the CC method's performance with linearly related data, as the DC should be sufficient to detect such relationships.

Chapter 2

Distance Correlation Based Feature Selection

As previously stated in Chapter 1, we plan to train RF models using different techniques to determine if any benefits exist using the method presented in this chapter. The method outlined in this chapter is a DC-based feature selection method in RF, as the chapter title suggests. Our approach will be compared to the existing feature selection methods, namely CC, RLT, and the conventional Breiman RF. We will use the mean-squared error (MSE) as a performance metric.

2.1 Distance Correlation (DC)

In their work, [SRB07] proposed a statistical measure called distance correlation (DC) that quantifies all forms of dependence between random vectors X and Y in arbitrary dimensions, unlike Pearson CC, which is limited to two-dimensional variables. The DC ranges from 0 to 1, and it equals 0 only when the random vectors are independent. According to [SR09a], the DC is effective in detecting nonlinear relationships that cannot be detected by the Pearson CC. The DC ($\mathcal{R}$), is a measure of dependence between two variables that measure the distance between their two characteristic functions. In the bivariate normal case, the DC becomes the Pearson product-moment correlation ρ (CC). Following [SR09a], for jointly distributed random vector $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$, let $f_{X,Y}(t, s)$ be the joint characteristic function of (X, Y) and $f_X(t)$ and $f_Y(s)$ be the corresponding

marginal characteristic functions.

Definition 2.1 *Suppose random variables X and Y have finite and positive variances. Then, the distance correlation ($\mathcal{R}$) is defined as,*

$$\mathcal{R}(X,Y) = \frac{dCov(X,Y)}{\sqrt{dCov(X,X) \cdot dCov(Y,Y)}}, \tag{2.1}$$

where $dCov(X,Y)$ is the distance covariance between random variables X and Y.

The calculation of $dCov(X,Y)$ is more complex compared to the relatively simple calculations performed when computing the covariance for the CC. However, we are fortunate that the R package "energy," authored by Rizzo, simplifies the calculation of the following definition.

Definition 2.2 *The distance covariance $dCov(X,Y)$ between X and Y is defined as the nonnegative square-root of*

$$dCov(X,Y) = \sqrt{\int_{R^{p+q}} ||f_{X,Y}(t,s) - f_X(t)f_Y(s)||^2 w(t,s)dtds}, \tag{2.2}$$

where $f_X(\cdot), f_Y(\cdot)$, and $f_{X,Y}(\cdot)$ are the characteristic and joint characteristic functions of the random variables X (p-dimensional) and Y (q-dimensional). The weight function is given by $w(t,s) = (c_p c_q ||t||_{p+1}^p ||s||_{q+1}^q)^{-1}$ and $w(t,s) > 0$ a.s. $t \in \mathbb{R}^p, s \in \mathbb{R}^q$ and c_d is the normalizing constant defined as $\pi^{(1+d)/2}/\Gamma((1+d)/2)$.

For more details, readers are referred to [SR23]. It is interesting to note that, according to [SR09b], the population distance covariance coincides with the covariance with respect to Brownian motion, the random motion of particles suspended in a medium. In the same article, the distance correlation is described as the "natural extension" of the CC, and it is clear that the DC offers certain advantages over the CC. Before proceeding, let us outline some of the properties of the DC.

1. $\mathcal{R}(X,Y)$ is defined for X and Y in arbitrary dimensions;
2. $\mathcal{R} = 0$ characterizes independence of X and Y.
3. $0 \leq \mathcal{R} \leq 1$.

4. If $\mathcal{R} = 1$, then there exists a vector a, a nonzero real number b and an orthogonal matrix C such that $Y = a + bXC$.

In terms of advantages, DC surpasses CC in several ways. For example, while CC is restricted to two-dimensional variables, DC can handle variables in any dimension. Moreover, the range of $\mathcal{R}$ is between 0 and 1, which is inclusive. It is interesting to note that when CC = 0, there is no linear correlation, but this does not indicate independence, whereas $\mathcal{R}(X, Y) = 0$ indicates independence between X and Y. Our aim is to utilize DC as a criterion for our filter method. However, having these advantages over CC does not necessarily mean that our filter method would perform better than the one presented in [Won19]. Nonetheless, there is a reason for optimism since [DKS22] employs DC as a feature selection criterion in selecting features for energy polynomials. It is worth noting that they achieved a performance that matched that of the unfiltered models using two orders of magnitude fewer parameters.

2.2 Feature Selection Method in Random Forest

Our focus is on exploring how distance correlation can facilitate feature selection. To this end, we employ a feature selection algorithm to enhance our machine-learning models, particularly random forests. The goal of our feature selection algorithm is to reduce the feature space by considering the DC between each feature and the dependent variable, using a threshold value of $\mathcal{R}$, denoted by $\mathcal{R}^*$.

As outlined above, our approach involves creating a subset of this feature space using training data, which will then be employed to train a random forest model. To achieve this, we first specify a threshold value, denoted by $\mathcal{R}^*$. We then compute $\mathcal{R}(Y, X_i)$ for $i = 1, \ldots, p$. Based on the resulting distance correlation values, we identify a subset of $\mathbf{X}^*$, denoted by $\mathbf{X}^* \subseteq \mathbf{X}$, that includes any feature X_j satisfying $\mathcal{R}(Y, X_j) \geq \mathcal{R}^*$. We subsequently employ $\mathbf{X}^*$ to construct a random forest and compute the mean squared error (MSE) using test data.

Algorithm: Proposed DC based Method

Given a training data set $\mathcal{D}_n$ and the distance correlation set $\overrightarrow{\mathcal{R}^*}$ of length s,

1. Compute the distance correlation between Y and each feature X_j and rank the features using the distance correlation.

2. For each $\mathcal{R}^*$,

   (a) eliminate the less correlated variables using the specified $\mathcal{R}^*$ as a threshold.

   (b) Using the new training data with reduced feature space, construct a random forest using the Breiman RF algorithm.

3. Given the s constructed random forests, select the model with the minimum prediction error based on the value of $\mathcal{R}^*$.

2.3 Theoretical Results

In this section, we develop large sample theory for the proposed DC-based feature selection method. We assume that our features are statistically independent and that only the relevant ones have a strong correlation with the response variable. Consider the model

$$Y = f(X_i) + \epsilon_i.$$

As in [ZZK15], we assume a moment condition on the random error terms ϵ_i. Our goal is to ensure that our variable importance measure still converges and that it depends only on the filtered features. The $j-$th variable importance is calculated based on randomly permuting the values of X_j in the out-of-bag sample which is denoted by $\tilde{X}_j$. Given that we are using a regression tree and have chosen to minimize the sum of squared errors as our criterion, the resulting squared error after permutation can be calculated

$$E_{\tilde{X}_j}\left(Y - \widehat{f}\big(X_1, \ldots, \tilde{X}_j, \ldots, X_{p^*}\big)\right)^2$$

We can express the variable importance for the j-th variable as follows.

$$VI_j = \frac{E\left[\left(f\big(X_1, \ldots, \tilde{X}_j, \ldots, X_p\big) - f\big(X_1, \ldots, X_j, \ldots, X_p\big)\right)^2\right]}{E\left[\left(Y - f\big(X_1, \ldots, \tilde{X}_j, \ldots, X_p\big)\right)^2\right]}. \tag{2.3}$$

Theorem 2.3 *Under assumptions 3.1, 3.2, 3.3, and 3.4 of [ZZK15], and there exists a fixed constant* $1 < B < \infty$*, for any* $\kappa > 0$*, the estimated variable importance converges to the true variable importance at an exponential rate. That is*

$$P\left(|\widehat{VI}_j - VI_j| > \kappa\right) \leq e^{-\kappa \cdot n^{\nu(p^*)}/B},$$

where $0 < v(p^*) \leq 1$ *is a function of the dimension* p^**, which represents the reduced number of features obtained using the DC-based filter method.* VI_j *is a measure of variable importance for each variable* $j \in \mathcal{P}$*, as defined in (2.3), along with its estimate* $\widehat{VI}_j$*.*

Proof: Following the similar arguments used in [Won19], we can prove the Theorem 2.3 and are thus omitted.

Chapter 3

Simulation Study

In this section, we perform a simulation study to assess the efficacy of our proposed method. In addition to the simulation setup used in [Won19], we examine two additional settings. For each setting, we generate 200 training samples and 1000 test samples. We evaluate the performance of our approach for various numbers of features, namely $p = 80, 100, 300, 500$.

- Under settings 1 & 2, we consider the following model

$$\textbf{Model 1:}\ \ Y_i = 5\big(X_{i,1} + X_{i,2} + X_{i,3} + X_{i,4}\big) + \epsilon_i \tag{3.1}$$

  where ϵ_i's are the random errors that are normally distributed with a mean of 0 and variance of 1.

  - **Setting 1:** Generate X_i from a normal distribution: $N\big(0_{p\times 1}, \Sigma_{p\times p}\big)$, where $\Sigma_{i,j} = \rho^{|i-j|}$, with $\rho = 0.5$ and 0.8.
  - **Setting 2:** Generate X_i from a normal distribution: $N\big(0_{p\times 1}, \Sigma_{p\times p}\big)$, where $\Sigma_{i,j} = \rho^{|i-j|} + 0.2I(i \neq j)$, with $\rho = 0.5$, where $I(\cdot)$ is called the indicator function.

- Under setting 3, we consider the following model

$$\textbf{Model 2:}\ \ Y_i = X_{i,1}^2 + X_{i,20} + X_{i,33}^3 + X_{i,55}^2 + \epsilon_i \tag{3.2}$$

  where ϵ_i's are the random errors that are normally distributed with a mean of 0 and variance of 1.

  - **Setting 3:** Generate X_i from a normal distribution: $N\big(0_{p\times 1}, \Sigma_{p\times p}\big)$, where $\Sigma_{i,j} = \rho^{|i-j|}$, with $\rho = 0.8$.

- Under setting 4, we consider the following model

$$\textbf{Model 3:}\ \ Y_i = 100 \times \big(X_{i,1} - 0.5\big)^2 \times \big(X_{i,2} - 0.25\big)^+ + \epsilon_i \tag{3.3}$$

  where $(\cdot)^+$ represents the positive part and ϵ_i's are the random errors that are normally distributed with a mean of 0 and variance of 1.

  - **Setting 4:** Generate X_i from Unif$[0,1]^p$.

The first step of our method involves calculating the distance correlation between the response variable Y and each feature variable X_j for all $j = 1, \ldots, p$. Next, we use pre-defined thresholds to select significant features. These thresholds are determined based on minimum distance correlation levels between Y and X_j, which include $\overrightarrow{\mathcal{R}}^* = \big\{0.00, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60\big\}$. If $\mathcal{R}^* = 0$, then all features are selected and included in the random forest regression. Conversely, if $\mathcal{R}^* = 0.5$, then only features with a distance correlation of at least 0.5 with the response variable are selected and added to the RF at each stage. We repeated the procedure 200 times to obtain reliable results.

3.1 Analysis of the Linear Models

Table 3.1 presents the results for all methods for Model 1 and Setting 1 with $\rho = 0.5$.

Table 3.1: Prediction Mean Squared Error for Model 1 and Setting 1 with $\rho = 0.5$

Method		$p = 80$	$p = 100$	$p = 300$	$p = 500$
Traditional RF		30.4468	32.3146	37.0157	39.9092
No	RLTNo1	17.1149	18.2449	20.6827	22.2395
	RLTNo2	8.3586	9.2965	10.8636	12.1497
	RLTNo5	5.9539	6.8420	8.4067	9.5437
Moderate	RLTMod1	23.5688	24.9247	29.2962	31.4494
	RLTMod2	12.7399	13.8862	16.9476	19.1914
	RLTMod5	9.7806	10.9047	13.5140	15.6142
CC (r^*)	0	30.4568	32.3099	36.9560	39.9454
	0.1	22.8696	24.6372	29.9454	33.0442
	0.2	16.5787	16.7887	16.9566	18.2652
	0.3	15.9218	15.8904	15.7830	15.7455
	0.4	13.3106	13.4890	13.0326	13.0766
	0.5	12.5500	12.8932	12.4917	12.5678
	0.6	16.4444	16.9558	16.4051	15.5541
DC ($\mathcal{R}^*$)	0	30.5103	32.2662	36.9264	39.9739
	0.1	30.4394	32.3129	36.9792	39.9157
	0.2	30.4860	32.2304	37.0245	39.8639
	0.3	30.2126	32.1138	37.0334	39.8655
	0.4	20.8794	22.2660	27.2499	30.5149
	0.5	16.7517	16.6341	16.3208	16.6678
	0.6	13.7511	13.8123	13.5889	13.3938

One trend that is evident is that the increase in the number of parameters (p) leads to an increase in the MSE. This implies that the model's accuracy decreases as the number of parameters increases, which is expected. The RLTNo5 model, which is RLT without muting where five features are utilized in the linear combination to create a split candidate, performed significantly better than other models. On the other hand, the traditional RF had the worst performance, which is desirable since our aim is to enhance the traditional RF with our methods. The optimal r^* threshold is likely between 0.4 and 0.6, although the optimal $\mathcal{R}^*$ threshold value is inconclusive. Nonetheless, the general trend indicates that as $\mathcal{R}^*$ increases, MSE decreases. It appears that the best model has an $\mathcal{R}^* > 0.6$, but we found that this was not the case. For $\mathcal{R}^* > 0.6$, the model's accuracy decreased, and we even encountered errors for $\mathcal{R}^*$ values that were excessively high since this meant that the model was discarding all parameters, and as a result, no random forest could be generated. It is probable that for these settings, the optimal $\mathcal{R}^*$ threshold is between 0.5 and 0.7.

We observed a significant improvement in the CC method's performance in the RF model when r^* increased from 0.1 to 0.2 in the $p = 500$ column. This resulted in a 44.7% decrease in MSE. Similarly, there was a 45.4% reduction in MSE when our method's threshold $\mathcal{R}^*$ increased from 0.4 to 0.5. It is possible that the similarity in the magnitude of these MSE drops is coincidental. However, we observed a similar pattern for $p = 80, 100$, and 300. To clarify, let $\text{MSE}_{\text{DC}_{\mathcal{R}^*,p}}$ represent the DC MSE at $\mathcal{R}^*$ and p. Similarly, let $\text{MSE}_{\text{CC}_{r^*,p}}$ be the CC MSE at r^* and p. We noticed the following trend:

$$\left|\frac{\text{MSE}_{\text{DC}_{.5,80}}}{\text{MSE}_{\text{DC}_{.4,80}}} - \frac{\text{MSE}_{\text{CC}_{.2,80}}}{\text{MSE}_{\text{CC}_{.1,80}}}\right| = 0.0774$$

$$\left|\frac{\text{MSE}_{\text{DC}_{.5,100}}}{\text{MSE}_{\text{DC}_{.4,100}}} - \frac{\text{MSE}_{\text{CC}_{.2,100}}}{\text{MSE}_{\text{CC}_{.1,100}}}\right| = 0.0656$$

$$\left|\frac{\text{MSE}_{\text{DC}_{.5,300}}}{\text{MSE}_{\text{DC}_{.4,300}}} - \frac{\text{MSE}_{\text{CC}_{.2,300}}}{\text{MSE}_{\text{CC}_{.1,300}}}\right| = 0.0327$$

$$\left|\frac{\text{MSE}_{\text{DC}_{.5,500}}}{\text{MSE}_{\text{DC}_{.4,500}}} - \frac{\text{MSE}_{\text{CC}_{.2,500}}}{\text{MSE}_{\text{CC}_{.1,500}}}\right| = 0.0065$$

The DC-based model accuracy eventually improves to a comparable level with the CC-based model when $\mathcal{R}^*$ reaches approximately 0.5. However, this is not the optimal $\mathcal{R}^*$ value, just as $r^* = 0.2$ is not the optimal threshold. In this case, the CC method easily identifies the more important parameters, while the DC method is more cautious and does not filter out parameters with weak linear correlations. The best prediction MSEs are achieved at $r^* = 0.5$ for the CC method and $\mathcal{R}^* = 0.6$ for the DC method. Although a higher $\mathcal{R}^*$ threshold is required for the DC method to optimize, the prediction MSE results are comparable to those of the CC method.

Table 3.2: Prediction Mean Squared Error for Model 1 and Setting 1 with $\rho = 0.8$

Method		$p = 80$	$p = 100$	$p = 300$	$p = 500$
Traditional RF		16.4542	16.8286	20.2293	21.4920
No	RLTNo1	11.1426	11.6650	13.5729	14.1749
	RLTNo2	6.8722	7.3101	8.8551	9.6527
	RLTNo5	5.4821	5.8649	7.3025	8.0649
Moderate	RLTMod1	14.9992	15.5370	18.7807	19.8693
	RLTMod2	10.3251	10.8486	13.8485	15.1718
	RLTMod5	8.4156	8.9015	11.3316	12.5533
CC (r^*)	0	16.4618	16.8028	20.2333	21.5206
	0.1	13.0510	13.2847	16.1036	17.3913
	0.2	10.7760	10.5976	10.9608	11.1928
	0.3	10.2295	10.0385	10.0109	10.0872
	0.4	9.2580	9.0398	9.0732	9.1315
	0.5	8.5590	8.4243	8.5828	8.5259
	0.6	9.1113	9.0128	9.1327	9.0838
DC ($\mathcal{R}^*$)	0	16.4589	16.8685	20.2596	21.5370
	0.1	16.4747	16.8312	20.2180	21.5444
	0.2	16.4707	16.7899	20.1973	21.5172
	0.3	16.3218	16.7368	20.2653	21.5056
	0.4	12.2518	12.5301	14.5710	15.9063
	0.5	10.3558	10.2450	10.2731	10.3228
	0.6	9.4236	9.2640	9.3533	9.3839

According to Table 3.2, we see the same optimal threshold values of r^* and $\mathcal{R}^*$. The optimal MSEs for the DC and CC methods are even closer, but the CC method still has a slight edge. The race for the best MSE is now closer with RLT, but RLTNo5 remains the best model, while the traditional RF remains the least accurate. As the correlation between parameters and the response variable increases, the MSE generally decreases compared to Table 3.1.

Table 3.3: Prediction Mean Squared Error for Model 1 and Setting 2 with $\rho = 0.5$

Method		$p = 80$	$p = 100$	$p = 300$	$p = 500$
Traditional RF		21.9640	23.6652	28.2053	30.0032
No	RLTNo1	13.0988	14.2620	16.5793	17.3747
	RLTNo2	7.3378	8.2417	10.2177	11.1712
	RLTNo5	5.5720	6.3689	8.2305	9.2038
Moderate	RLTMod1	17.9596	19.3122	23.0986	24.3520
	RLTMod2	11.4233	12.5715	16.2147	17.8654
	RLTMod5	9.1465	10.2833	13.5496	15.2372
CC (r^*)	0	21.9342	23.6987	28.1940	29.9885
	0.1	21.7451	23.6321	28.2193	29.9617
	0.2	20.9032	22.9340	27.5293	29.3341
	0.3	16.8882	18.6162	23.0721	25.1728
	0.4	11.9670	12.4959	13.4938	14.0448
	0.5	11.3873	11.7433	11.6022	11.3566
	0.6	9.0305	9.2198	9.3215	9.1254
DC ($\mathcal{R}^*$)	0	21.9021	23.7338	28.1547	29.9792
	0.1	21.8892	23.7192	28.1623	30.0492
	0.2	21.8888	23.6486	28.1887	30.0208
	0.3	21.8853	23.7239	28.2238	29.9949
	0.4	21.6011	23.4470	28.0920	29.8334
	0.5	19.3799	21.3558	26.1481	28.1744
	0.6	12.5753	13.4929	15.1863	16.6041

In Table 3.3, we observe that the CC method outperforms our method and marks the first instance where a better model than RLTNo5 is identified. It is possible that the DC method could achieve comparable results at a higher threshold, but we did not have the opportunity to optimize this threshold for the DC method.

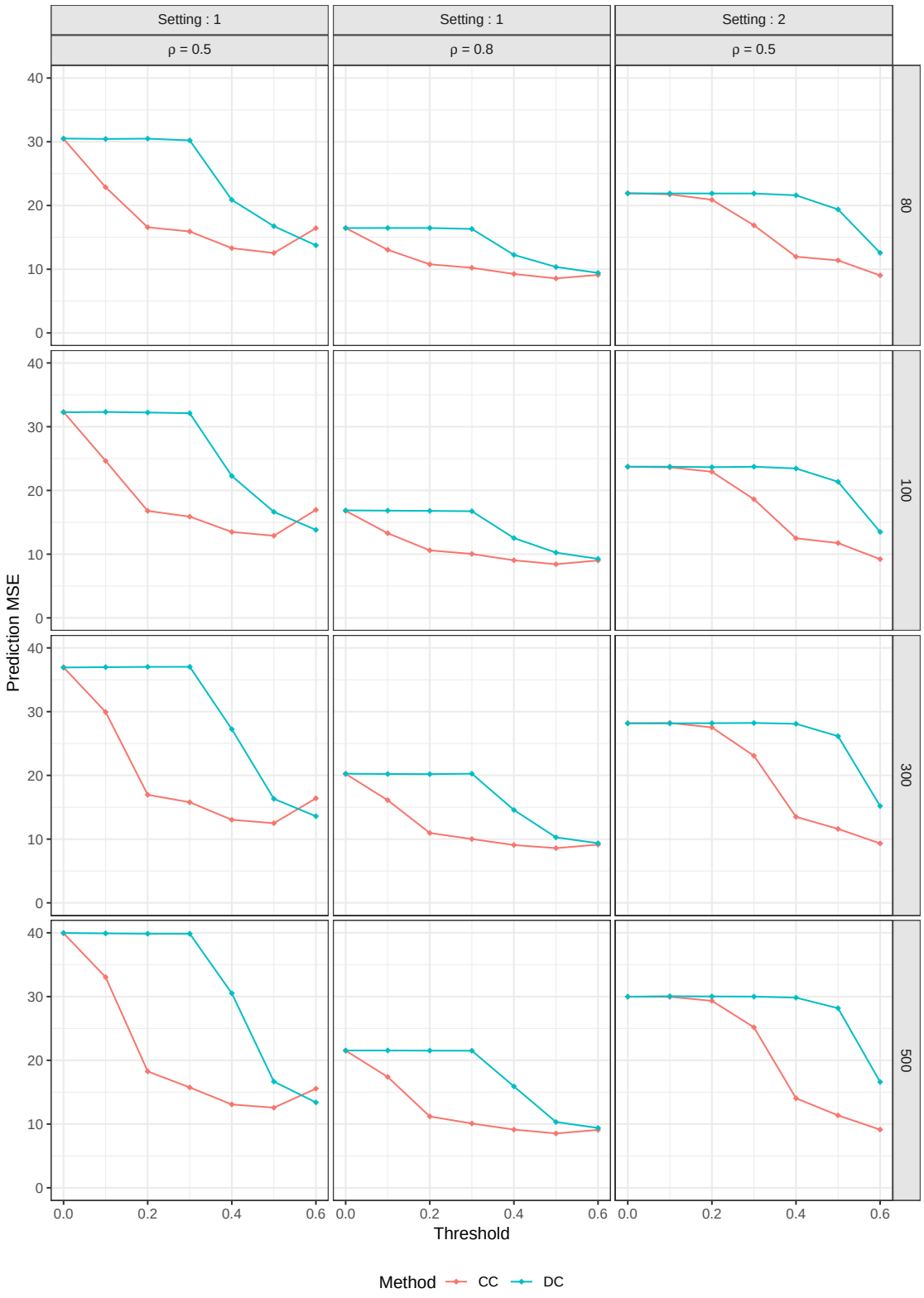

Figure 3.1: Prediction MSE Comparison for Setting 1 ($\rho = 0.5, 0.8$) and Setting 2 with $\rho = 0.5$

3.2 Analysis of the Nonlinear Model

In this section, we examine a nonlinear model as outlined in setting 3. The results are presented in Table 3.4.

Table 3.4: Prediction Mean Squared Error for Model 2 and Setting 3 with $\rho = 0.8$

Method		$p = 80$	$p = 100$	$p = 300$	$p = 500$
Traditional RF		9.4389	9.5245	10.4246	10.7869
No	RLTNo1	8.6755	8.7385	9.4071	9.7955
	RLTNo2	8.5479	8.6631	9.4587	9.9032
	RLTNo5	8.6720	8.7762	9.5994	10.0118
Moderate	RLTMod1	9.6584	9.7615	10.7009	11.2133
	RLTMod2	9.7378	9.8579	10.9569	11.4871
	RLTMod5	9.8222	9.9758	11.0402	11.6132
CC (r^*)	0	10.5241	10.4354	11.7246	12.1731
	0.1	11.0046	10.9849	12.0554	12.3790
	0.2	11.3745	11.1895	11.8162	11.9509
	0.3	10.8041	10.5800	10.9673	10.8763
DC ($\mathcal{R}^*$)	0	9.4371	9.5271	10.4387	10.7732
	0.1	9.4270	9.5461	10.4322	10.7692
	0.2	9.4465	9.5276	10.4433	10.7636
	0.3	9.4336	9.5344	10.4295	10.7577
	0.4	8.9385	8.9611	9.6091	9.8364
	0.5	9.4990	9.4992	9.5010	9.4111
	0.6	10.4607	10.4244	10.3874	10.3362

These results are particularly exciting as they reveal the advantages of using DC as a feature selection criterion. It is worth noting that the CC method threshold stops at 0.3 because, as the data is not constructed under a linear model, setting a CC threshold higher than 0.3 will filter out all the parameters of the model, making it impossible to construct an RF. This is not the case with the DC method, as it is capable of detecting nonlinear correlations and allowing more parameters to survive the filter method. Although the CC method does not perform well in this case, we can see that RLT remains the best method for $p = 80, 100$, and 300. However, for the high-dimensional case, our proposed method performs best, indicating that it could be an improvement over RF in high-dimensional scenarios. In the future, it would be interesting to compare the proposed method with other machine learning techniques in high-dimensional datasets that exhibit nonlinear correlations. Additionally, we assess the benefits of the proposed

method using the simulation setting employed in a previous study [ZZK15]. The outcomes of this analysis are presented in Table 3.5.

Table 3.5: Prediction Mean Squared Error for Model 3 and Setting 4

Method		$p = 80$	$p = 100$	$p = 300$	$p = 500$
Traditional RF		6.1719	6.3132	7.0491	7.4381
No	RLTNo1	2.4868	2.4958	2.9554	3.3648
	RLTNo2	2.5882	2.6486	3.3094	3.8033
	RLTNo5	2.8512	2.8675	3.5907	4.3271
Moderate	RLTMod1	3.1720	3.1258	3.8918	4.5346
	RLTMod2	3.6176	3.5186	4.5701	5.1221
	RLTMod5	3.7851	3.7519	4.8743	5.7918
CC (r^*)	0	6.1638	6.2397	7.0040	7.4891
	0.1	8.6832	8.9644	9.0353	9.1730
	0.2	10.7540	10.7789	10.7112	10.5731
	0.3	12.2340	12.2444	11.8764	12.3109
DC ($\mathcal{R}^*$)	0	6.1879	6.1030	7.0218	7.5451
	0.1	6.1925	6.0984	6.9839	7.4811
	0.2	6.2513	6.0910	6.9863	7.4811
	0.3	6.1112	6.0962	7.0018	7.4744
	0.4	5.5324	5.5445	6.2003	6.7826
	0.5	2.6557	2.5385	2.8895	3.2704
	0.6	9.8633	9.5040	9.4988	9.9643

The performance of our DC method is outstanding compared to both traditional RF and the CC method in this nonlinear simulated dataset, similar to our other nonlinear simulated dataset. It is worth noting that the CC method has a lower threshold, as a threshold higher than 0.3 eliminates all parameters from the RF model. As we have previously observed, RLT performs exceptionally well here. However, as seen in setting 3, as the number of parameters increases, our proposed method appears to gain an advantage over RLT. Specifically, the DC-based feature selection method outperforms RLT for $p = 300$ and $p = 500$. This once again supports the notion that the DC-based method may be an excellent candidate for high-dimensional data analysis.

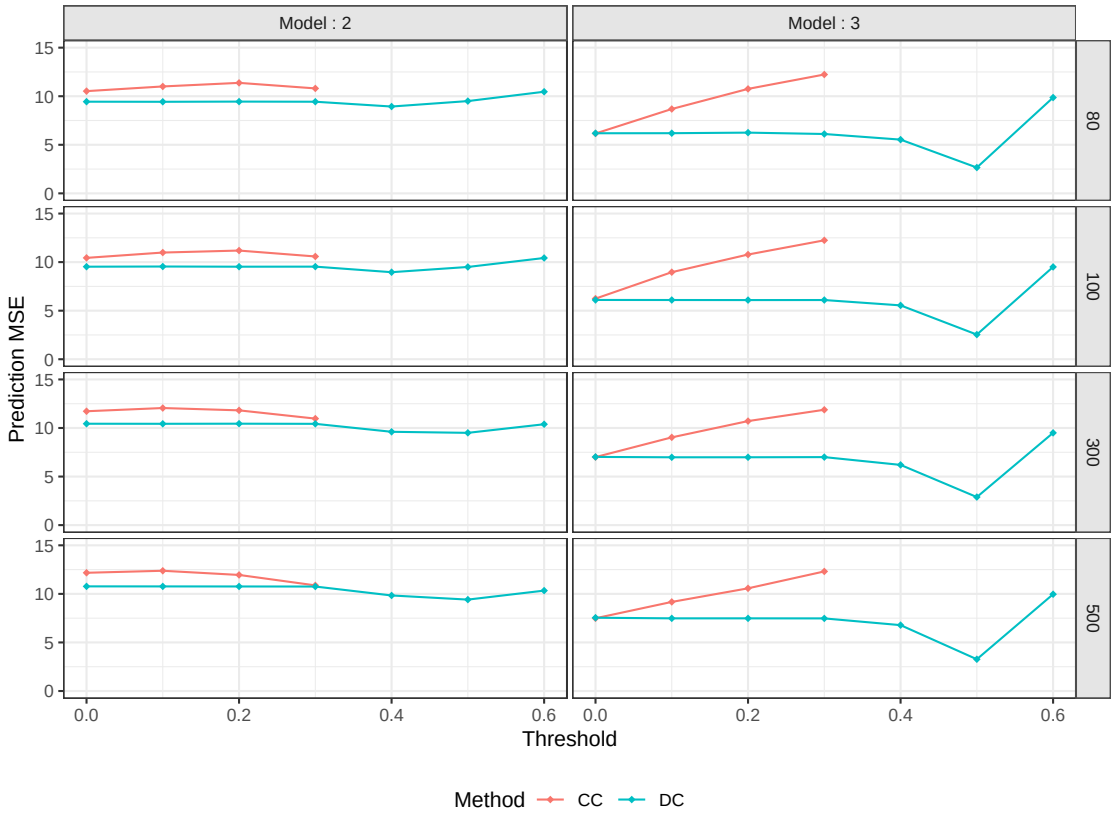

Figure 3.2: Prediction MSE Comparison for Model 2 & 3

According to Figure 3.2, it is evident that the DC-based method outperforms CC significantly. In setting 4, we observe that our optimal MSE is often less than half of the CC method's MSE.

Chapter 4

Real Data Applications

To illustrate the practical usage, we apply our proposed methods to two real datasets, which are provided below.

1. **Riboflavin Data:**

   This dataset contains riboflavin production by Bacillus subtilis. There are $n = 71$ observations of $p = 4088$ predictors (gene expressions) and a one-dimensional response variable.

2. **Boston Housing Data:**

   This dataset contains housing data for 506 census tracts of Boston from the 1970 census. There are $n = 506$ observations of $p = 14$ predictors.

4.1 Riboflavin Data

The Riboflavin dataset is a widely-used dataset found in the 'hdi' R package, provided by [BKM14]. It consists of 71 observations of 4088 predictors, representing the expression levels of 4088 genes, and a single response variable, which is the riboflavin production of Bacillus Subtilis. The objective of our study is to predict the log-transformed riboflavin production rate using gene expressions as predictors. This dataset is an example of a high-dimensional dataset, as the number of features is much larger than the number of observations, i.e., $p > n$. The results of our analysis are presented in Table 4.1.

Table 4.1: Prediction Mean Squared Error for Riboflavin Data

Traditional RF		0.5029
No	RLTNo1	0.5521
	RLTNo2	0.5459
	RLTNo5	0.5436
Moderate	RLTMod1	0.5555
	RLTMod2	0.5216
	RLTMod5	0.5623
Threshold	CC (r^*)	DC ($\mathcal{R}^*$)
0.00	0.5026	0.5071
0.05	0.4936	0.5133
0.10	0.4866	0.5049
0.15	0.4654	0.5104
0.20	0.4521	0.5130
0.25	0.4356	0.5043
0.30	0.4217	0.5063
0.35	0.4083	0.5076
0.40	0.3864	0.5100
0.45	0.4076	0.5029
0.50	0.5594	0.4990
0.55	0.4175	0.4873
0.60	0.5565	0.4628
0.65	NA	0.4358
0.70	NA	0.4126

To ensure stable results, we conduct 200 repetitions and calculate the average prediction mean squared error. The findings indicate that the CC-based feature selection method is much more precise than the RLT methods and significantly better than the traditional RF. Our proposed method comes in second place with an optimal threshold of 0.7. It is worth noting that a better $\mathcal{R}^*$ threshold may exist in the range of (6.5,7.5).

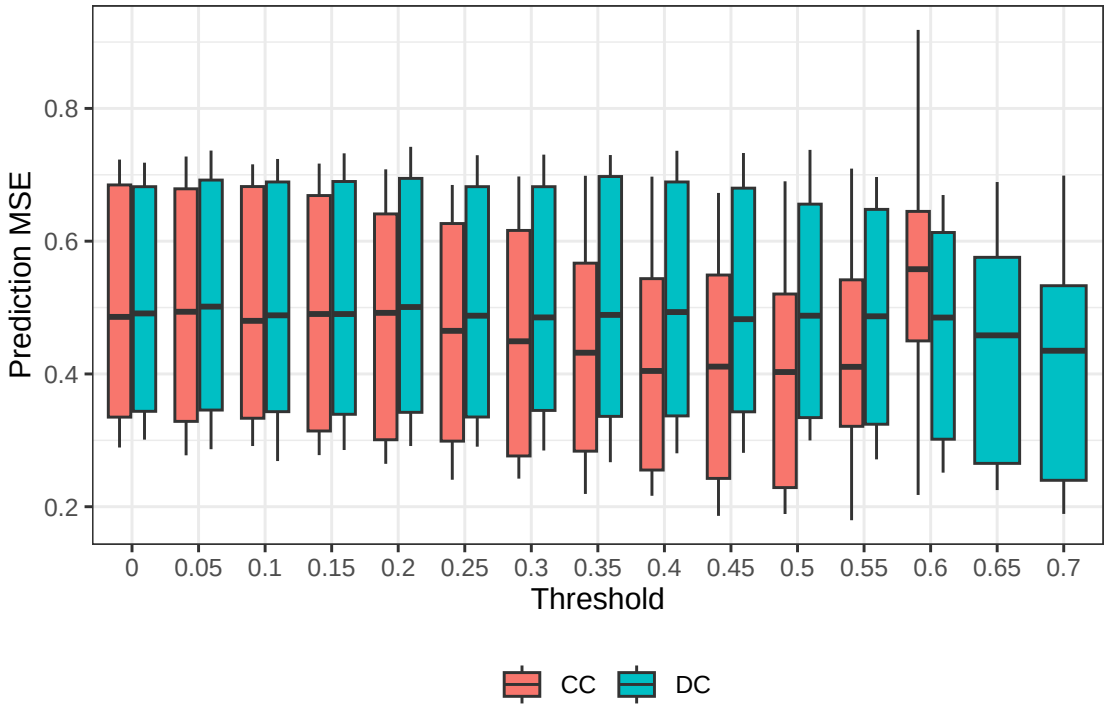

Figure 4.1: Boxplot for Prediction MSE Comparison for Riboflavin Data

The results indicate that the methods have similar accuracy, but the CC method performs better. In support of this, Figure 4.1 shows a continued decrease in MSE as the $\mathcal{R}^*$ threshold increases, suggesting that an optimal threshold may exist beyond 0.7. However, even with this potential for improvement, the results obtained with our proposed method are comparable at best to those of the CC method.

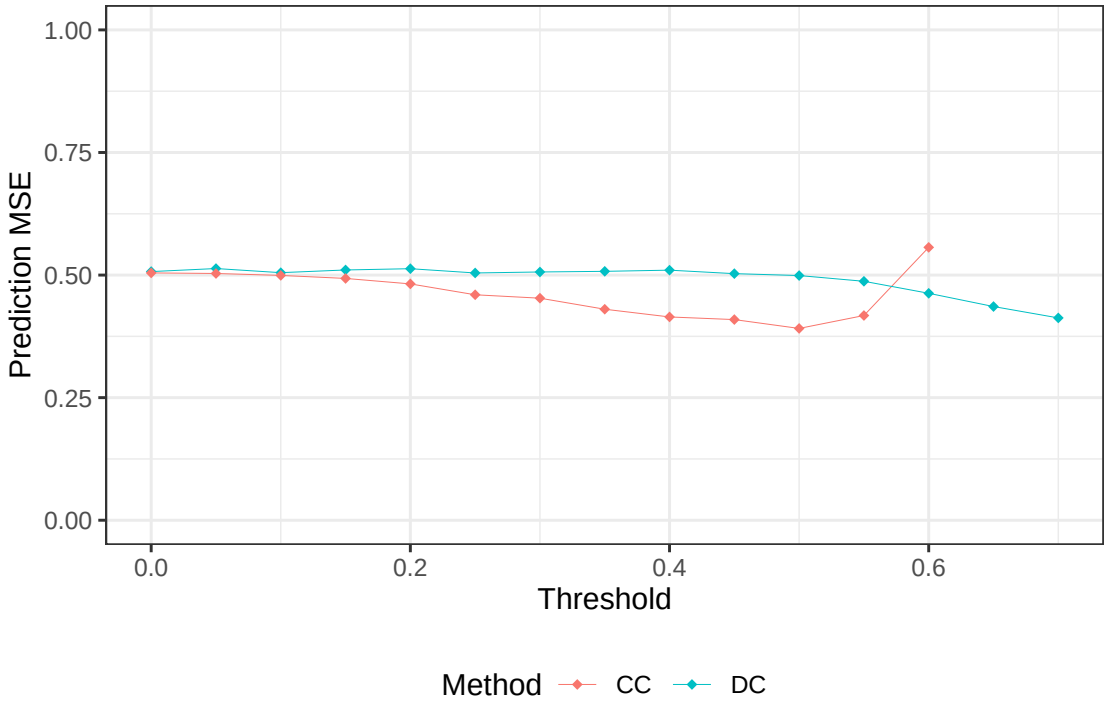

Figure 4.2: Prediction MSE Comparison for Riboflavin Data for CC and DC-based Methods

Figure 4.2 illustrates the diminishing returns of increasing the CC threshold and highlights the potential for a better prediction of mean squared error (MSE) by increasing the DC threshold.

4.2 Boston Housing Data

The Boston housing data set is provided by [HJR78] and is a built-in data set in R. Unlike the riboflavin data set, it has a lower dimensionality with only 13 predictors and a 1-dimensional response variable. The data set contains 506 observations and provides information gathered from the 1970s census. The predictors include the per capita crime rate by town, the average number of rooms per dwelling, the pupil-teacher ratio by town, and other factors. The response variable is the median value of owner-occupied homes in $1000. The objective is to use the available information, such as the per capita crime rate by town (CRIM), nitric oxides concentration (NOX), proportion of non-retail business acres per town (INDUS), and full-value property-tax rate per $10,000 (TAX), among others, to predict the median value of owner-occupied homes.

Table 4.2: Prediction Mean Squared Error for Boston Housing Data

Traditional RF		11.6123
No	RLTNo1	16.5492
	RLTNo2	16.7430
	RLTNo5	16.0898
Moderate	RLTMod1	16.0028
	RLTMod2	15.6108
	RLTMod5	15.6015
Threshold	CC (r^*)	DC ($\mathcal{R}^*$)
0.1	11.5548	11.5702
0.15	11.5674	11.5258
0.2	11.5926	11.5477
0.25	11.9115	11.5586
0.3	12.6297	11.5891
0.35	12.7505	11.5651
0.4	12.9315	11.5344
0.45	15.3672	11.5441
0.5	18.6801	11.5417
0.55	21.5029	11.5951
0.6	21.7865	11.9905
0.65	22.6410	12.5806
0.7	30.9052	13.0999

We applied the same methodology to analyze the Boston housing dataset, and the prediction MSE results are presented in Table 4.2. Similar to the Riboflavin dataset, we don't observe any improvement in the model by using the RLT method, but we see

slight improvements from the two filter methods compared to traditional RF. Furthermore, we notice that our proposed method slightly outperforms the CC method. Moreover, we observe that our proposed DC-based method has relatively stable results irrespective of the $\mathcal{R}^*$, whereas the CC method shows an increasing trend in prediction MSE and results in almost three times the MSE of the traditional RF as r^* varies from 0.1 to 0.7.

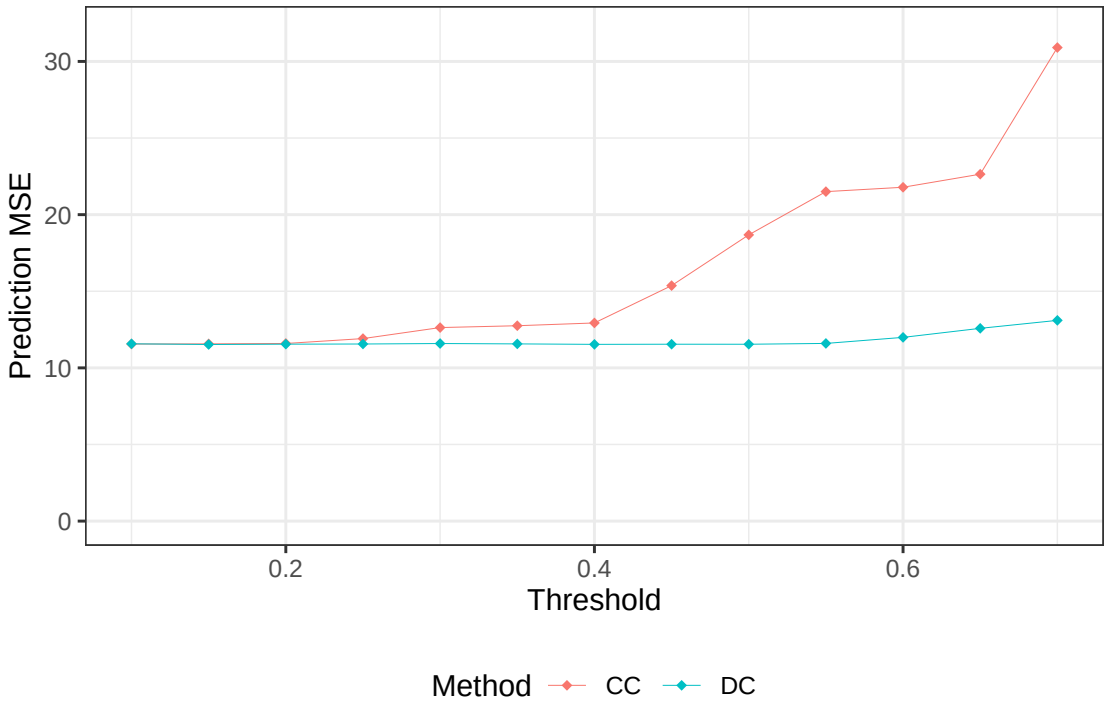

Figure 4.3: Prediction MSE Comparison for Boston Housing Data for CC and DC-based Methods

Once again, the results obtained support the notion that our DC-based feature selection method is more conservative in eliminating predictors that are relevant to the RF model compared to other methods. It is interesting to note that a similar trend as seen in Figure 3.1 can also be observed in Figure 4.3, where the change in MSE of the CC method from $r^* = 0.2$ to 0.4 is similar to that of the DC method from $\mathcal{R}^* = 0.5$ to 0.7. This change in MSE is approximately 12% for both methods as r^* and $\mathcal{R}^*$ vary within those ranges.

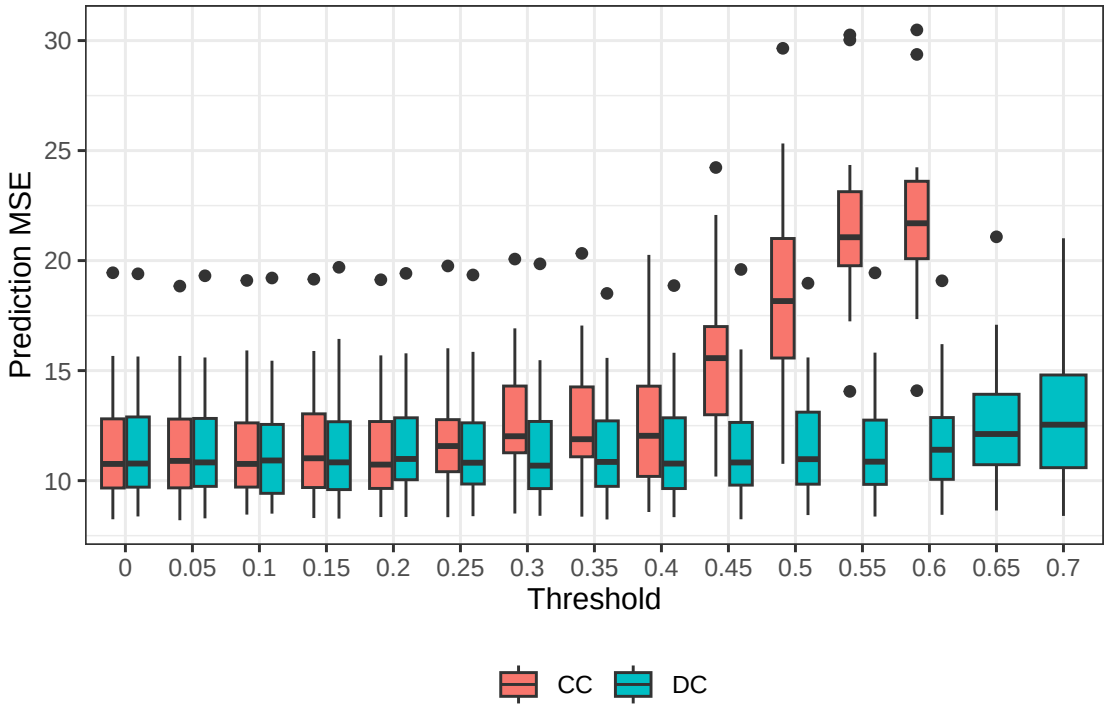

Figure 4.4: Boxplot of Prediction MSE for Boston Housing Data

Chapter 5

Conclusion and Discussion

5.1 Performance of the Proposed DC-Based Filter Method

In this thesis, we proposed a novel variable selection procedure for RF using distance correlation. We observed that the proposed DC-based method performed very well in most cases, especially in nonlinear models. Although we anticipated that our approach would perform similarly or better than the CC-based filter method, we were pleasantly surprised to find that it outperformed RLT methods under high-dimensional settings. Our approach consistently outperformed the traditional RF method, and in the case of the nonlinear models, it even outperformed the CC method.

In the linearly simulated data, we observed that the DC method performed similarly to the CC method in most cases. However, we noticed that optimizing the DC prediction MSE required a higher threshold, which is not surprising given that our method is more conservative in feature filtering. This is not a significant disadvantage, except perhaps for computational cost, as more features are retained in the RF model construction. To address this, we can adjust the threshold to a higher value.

We observed only one case where DC significantly underperformed the CC method, which was in setting 2. In this case, a strong linear correlation was simulated, and thus, the CC method was expected to perform well, which was indeed the case. However, in situations like this, we can consider increasing the DC threshold to 0.7 or 0.8 and see if the prediction MSE improves and becomes comparable to that of the CC method, as we observed previously. Our method demonstrated its superior performance in nonlin-

ear models, particularly in high-dimensional cases. This piqued our interest in exploring high-dimensional datasets. Finally, two real data applications are provided to illustrate the advantage of the proposed methods.

5.2 Future Work

As with many studies, our work leaves us with some unanswered questions, some of which may be easier to address. For instance, we are interested in finding an optimal threshold in cases where the DC method underperformed the CC method, and thus, we could decrease the step size between thresholds or devise an algorithm to identify the optimal threshold. However, to make meaningful comparisons with the CC method prediction MSE, we also need to optimize the r^* threshold. Therefore, we leave these considerations for future work.

Another area for future work, although more challenging than the previous one, would be to integrate our DC filter method with the RLT model, thereby combining the two best-performing methods to assess if a better model can be developed.

We should note that in the simulation data, we intentionally generated uncorrelated features. Therefore, an area for future research would be to conduct a simulation study of highly correlated predictors and assess the effectiveness of our proposed method in such scenarios. We can find inspiration from research such as [JNS$^+$20], which employs machine learning to investigate correlated meteorological parameters.

Finally, we are interested in applying our proposed method on high-dimensional, nonlinearly related datasets. Given the increasing demand for machine learning models that can handle high-dimensional datasets [D$^+$00], we are keen on applying our approach to such data.

Bibliography

[ALX+20] Animesh Acharjee, Joseph Larkman, Yuanwei Xu, Victor Roth Cardoso, and Georgios V. Gkoutos. A random forest based biomarker discovery and power analysis framework for diagnostics research. *BMC Medical Genomics*, 13(1), nov 2020.

[BDL08] G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9:2015–2033, 2008.

[BKM14] Peter Bühlmann, Markus Kalisch, and Lukas Meier. High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application*, 1:255–278, 2014.

[Bre01] L. Breiman. Random forest. technical report. *Stat.Dept. UCB*, 2001.

[CF94] R. Caruana and D. Freitag. Greedy attribute selection. *Proceedings of the Eleventh International Conference on Machine Learning*, page 28–36, 1994.

[CLWY18] Jie Cai, Jiawei Luo, Shulin Wang, and Sheng Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018.

[D+00] David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000.

[Das01] S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 74–81, 2001.

[Dav91] Lawrence Davis. Handbook of genetic algorithms. 1991.

[DB00] J. G. Dy and C. E. Brodley. Feature subset selection and order identification for unsupervised learning. *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 247–254, 2000.

[DCSL02] M. Dash, K. Choi, P. Scheuermann, and H. Liu. Feature selection for clustering – a filter solution. *Proceedings of the Second International Conference on Data Mining*, pages 115–122, 2002.

[DKS22] Ranit Das, Gregor Kasieczka, and David Shih. Feature selection with distance correlation. *arXiv preprint arXiv:2212.00046*, 2022.

[EHJT04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Stat.*, 32:407–499, 2004.

[GWBV02] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46:389–422, 2002.

[Hal00] M. A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 359–366, 2000.

[HCCG05] J. Ham, Yangchi Chen, M.M. Crawford, and J. Ghosh. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):492–501, 2005.

[HH+10] Hui-Huang Hsu, Cheng-Wei Hsieh, et al. Feature selection via correlation coefficient clustering. *J. Softw.*, 5(12):1371–1377, 2010.

[HJR78] David Harrison Jr and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning - data mining, inference, and prediction. 2001.

[JNS+20] Muhammad Jawad, Malik Sajjad Ahmed Nadeem, Seong-O Shim, Ishtiaq Rasool Khan, Aliya Shaheen, Nazneen Habib, Lal Hussain, and Wajid Aziz.

Machine learning based cost effective electricity load forecasting model using correlated meteorological parameters. *IEEE Access*, 8:146847–146864, 2020.

[KJ97] R Kohavi and GH John. Wrappers for feature subset selection, artificial intelligence, vol. 97, no. 1-2, 1997.

[LB16] Gildas Leger and Manuel J Barragan. Brownian distance correlation-directed search: A fast feature selection technique for alternate test. *Integration*, 55:401–414, 2016.

[LMC+20] Yaqing Liu, Yong Mu, Keyu Chen, Yiming Li, and Jinghuan Guo. Daily activity feature selection in smart homes based on pearson correlation coefficient. *Neural Processing Letters*, 51:1771–1787, 2020.

[MH05] Shuangge Ma and Jian Huang. Regularized roc method for disease classification and biomarker selection with microarray data. *Bioinformatics*, 21(24):4356–4362, 2005.

[Ng98] A. Y. Ng. On feature selection: learning with exponentially many irrelevant features as training examples. *Proceedings of the Fifteenth International Conference on Machine Learning*, page 404–412, 1998.

[Pap13] Vijay Sunder Naga Pappu. *Supervised machine learning models for feature selection and classification on high dimensional datasets.* University of Florida, 2013.

[Pea96] K. Pearson. Mathematical contributions to the theory of evolution. iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London Series*, A 187:253–318, 1896.

[SIL07] Yvan Saeys, Inaki Inza, and Pedro Larranaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[SR09a] G. J. Székely and M. L. Rizzo. Brownian distance covariance. *Ann. Appl. Stat.*, 3(4):1236–1265, 2009.

[SR09b] Gábor J. Székely and Maria L. Rizzo. Brownian distance covariance. *The Annals of Applied Statistics*, 3(4):1236 – 1265, 2009.

[SR23] G. J. Székely and M. L. Rizzo. The energy of data and distance correlation. *1st edition*, Chapman Hall, 2023.

[SRB07] G. J. Székely, M. L. Rizzo, and N. K. Bakirov. Measuring and testing independence by correlation of distances. *Annals of Statistics*, 35(6):2769–2794, 2007.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B*, 58(1):267–288, 1996.

[Won19] Y. T. Wonkye. Innovations of random forests for longitudinal data. *Bowling Green State University, Doctoral dissertation. OhioLINK Electronic Theses and Dissertations Center*, 2019.

[XJK01] E. Xing, M. Jordan, and R. Karp. Feature selection for high-dimensional genomic microarray data. *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608, 2001.

[ZH05] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B*, 67:301–320, 2005.

[Zou06] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1428, 2006.

[ZZK15] R. Zhu, D. Zeng, and M. Kosorok. Reinforcement learning trees. *Journal of the American Statistical Association*, 110(512):1770–1784, 2015.