Structure-Regularized Partition-Regression Models for Nonlinear System-Environment Interactions

by

Shuluo Ning

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2018 by the
Graduate Supervisory Committee:

Dr. Jing Li, Chair
Dr. Teresa Wu
Dr. Rong Pan
Dr. Tanveer A. Rafi

ARIZONA STATE UNIVERSITY

May 2018

ABSTRACT

Under different environmental conditions, the relationship between the design and operational variables of a system and the system's performance is likely to vary and is difficult to be described by a single model. The environmental variables (e.g., temperature, humidity) are not controllable while the variables of the system (e.g. heating, cooling) are mostly controllable. This phenomenon has been widely seen in the areas of building energy management, mobile communication networks, and wind energy. To account for the complicated interaction between a system and the multivariate environment under which it operates, a Sparse Partitioned-Regression (SPR) model is proposed, which automatically searches for a partition of the environmental variables and fits a sparse regression within each subdivision of the partition. SPR is an innovative approach that integrates recursive partitioning and high-dimensional regression model fitting within a single framework. Moreover, theoretical studies of SPR are explicitly conducted to derive the oracle inequalities for the SPR estimators which could provide a bound for the difference between the risk of SPR estimators and Bayes' risk. These theoretical studies show that the performance of SPR estimator is almost (up to numerical constants) as good as of an ideal estimator that can be theoretically achieved but is not available in practice. Finally, a Tree-Based Structure-Regularized Regression (TBSR) approach is proposed by considering the fact that the model performance can be improved by a joint estimation on different subdivisions in certain scenarios. It leverages the idea that models for different subdivisions may share some similarities and can borrow strength from each other. The proposed approaches are applied to two real datasets in the domain of building energy. (1) SPR is used in an application of adopting building design and operational variables, outdoor

environmental variables, and their interactions to predict energy consumption based on the Department of Energy's EnergyPlus data sets. SPR produces a high level of prediction accuracy and provides insights into the design, operation, and management of energy-efficient buildings. (2) TBSR is used in an application of predicting future temperature condition which could help to decide whether to activate or not the Heating, Ventilation, and Air Conditioning (HVAC) systems in an energy-efficient manner.

ACKNOWLEDGMENTS

During my graduate studies in Arizona State University, several persons and institutions collaborated directly and indirectly with my research. Without their support, it would be impossible for me to fulfill my graduate study.

First and foremost, I offer my sincerest gratitude to my advisor, Dr. Jing Li, who gave me the opportunity to study with her and guided me throughout my doctoral life with her patience and knowledge. Without her encouragement and effort, this dissertation would not have been completed. Besides my advisor, I would like to thank the rest of my dissertation committee members: Dr. Teresa Wu, Dr. Rong Pan and Dr. Tanveer A. Rafi for their guidance, insightful comments, and suggestions.

Members of AMIIL lab inspired me a lot through discussions, seminars, and project collaborations, and I would like to thank the following people for their valuable suggestions: Na Zou, Bing Si, Kun Wang, Hyunsoo Yoon, Xiaonan Liu, Nathan Gaw, Yinlin Fu, Can Cui, Congzhe Su, Fei Gao. I would like to offer my best wishes for their future study and career.

I would like to thank my wife, Yizhen Ji, who supports me with her care and understanding. Finally, I would like to thank my family, for their continuous support, inspiration, and love.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Background

Modeling the relationship between the design and operational variables of a system and the system's performance is of primary interest in various domains. When the system is functioning in different environments, this relationship is likely to vary. Here we give a few examples:

- In building energy management, an important topic is to model how building design and operational variables affect energy consumption. Identification of this relationship helps design and operate energy-efficient buildings. However, buildings with the same design and operation may have different levels of energy consumption/efficiency, depending on where the buildings are located. A good building design/operation needs to consider outdoor environmental conditions such as geographical location, temperature, humidity, and air flow rate (Eisenhower et al., 2012).
- In mobile communication networks, a key interest is to model how traffic volumetric variables affect Quality of Service (QoS) metrics such as packet delay and loss. Understanding this relationship helps network capacity management and optimization. It is well-known that mobile network operations are affected by environmental conditions, such as the type of landform (valley, mountain, or plain) and weather (Hardy et al., 2001).

- In wind energy industry, a persistent interest is to model how wind speed and direction affect the power output of a wind turbine. This relationship is used for a number of important tasks including prediction of wind power production and evaluation of the turbine's energy production efficiency. It has been found that this relationship varies with respect to environmental conditions such as location of the wind turbine (offshore or inland), temperature, humidity, and air pressure (Byon et al., 2015).

Particularly, in building energy management, energy spent in buildings represents more than 76% of all electricity use and 40% of all energy use in the U.S. Improving building energy efficiency is urgent for drastically reducing the consumption of scarce energy resources. To achieve this goal, wireless sensors and IoT technologies provide great promise by enabling data collection on various factors (both indoor and outdoor) that potentially affect building energy consumption. Integrated with predictive analytics, this would further allow prediction of energy consumption and timely adjustment of building parameters to minimize the consumption.

However, there are multifold challenges in developing predictive models in this arena. First, it is well-known that building energy consumption is affected by both indoor and outdoor variables. There is complicated interaction between the indoor and outdoor variables. Even buildings with the same design and operational parameters (i.e., indoor variables) may have different levels of energy efficiency, depending on where the buildings are located (i.e., the outdoor environment). Second, the outdoor variables (e.g., temperature, wind speed) are not controllable while the indoor variables (e.g., heating, cooling, lighting) are mostly controllable. Considering that our goal is to adjust controllable factors to

improve energy efficiency, an appropriate predictive model should be able to account for the different roles of indoor and outdoor variables and allow for control and adjustment. Third, buildings are complex systems for which a large number of factors could potentially affect energy consumption. This requires a predictive model with intrinsic capability for handling variable high-dimensionality.

Therefore, novel predictive models are needed to address all these challenges and provide accurate predictions for the system's outputs.

1.2 Literature Review

To account for the above challenges in predictive modeling, the existing research work can be categorized into two areas: regression-based models and tree-based models.

Let $\mathbf{X} = \left(X_1, \dots, X_p\right)^T$ denote the design and operational variables of a system (e.g., indoor variables of a building), called input variables in this research, Y denote the performance or output variable of the system (e.g., energy consumption of a building), and $\mathbf{Z} = \left(Z_1, \dots, Z_q\right)^T$ denote the environmental variables (e.g., outdoor variables of a building).

1.2.1 Regression-based Models

An intuitive approach is to concatenate the input and environmental variables into a single predictor set, which is then linked with Y by a statistical model such as a regression. To select important predictors, classic approaches are forward selection, backward elimination, and stepwise regression (Montgomery et al., 2015). With high-dimensional predictors especially under the "small-n-large-p" setting, optimization-based methods capable of selecting a *sparse* subset of the predictors have been shown to be more effective and indeed a popular research area in modern statistics and machine learning societies.

Typical sparse regression methods include lasso (Tibshirani, 1996), SCAD (Fan et al., 2001), adaptive lasso (Zou, 2006), group lasso (Yuan et al., 2006), elastic net (Zou et al., 2005), just to name a few. However, these methods do not explicitly model the interaction between **X** and **Z**. To consider the interaction in a regression model, a straightforward option is to apply the aforementioned methods to an expanded predictor set including not only the individual predictors but also their interactions up to an order of interest. However, this does not honor the well-known "hierarchical principles" in regression fitting, which state that an interaction term can only be included in a model if at least one (weak hierarchy) or all (strong hierarchy) of the individual predictors involved in the interaction term are also in the model (Hamada and Wu, 1992; Chipman, 1995).

To account for the hierarchical principles, most existing work focuses on models that involve pairwise interactions. If putting into our context, this means a model of the following format:

$$Y = \sum_{i=1}^{p} \alpha_i X_i + \sum_{j=1}^{q} \omega_j Z_j + \sum_{j=1}^{q} \sum_{i=1}^{p} \gamma_{ij} X_i Z_j + \varepsilon, \tag{1.1}$$

where α_i, ω_j, γ_{ij} and are regression coefficients. Specifically, Choi et al. (2010) proposed to re-parameterize the coefficient for each interaction term into a product, i.e., $\gamma_{ij} = \vartheta_{ij}\alpha_i\omega_j$, which enforces the strong hierarchy in the sense that whenever α_i and ω_j are zero, the coefficient for the interaction, γ_{ij}, is automatically zero. They further proposed to impose one l_1-regularization on ϑ_{ij} and another one on α_i and ω_j to enable a sparse estimation obeying the strong hierarchy. This model is non-convex and an iterative algorithm was developed for model estimation. Convex formulations enjoy better mathematical tractability and computational efficiency. Toward this end, there have been

a few developments. Yuan et al. (2009) proposed a convex optimization formulation by modifying the nonnegative garrote (Breiman, 1995) and adding linear inequality constraints to enforce hierarchy. Zhao et al. (2009) proposed the Composite Absolute Penalties (CAP) that allows given grouping and hierarchical relationships of predictors to be expressed. Bien et al. (2013) proposed to honor the strong and weak hierarchy by extending the lasso formulation to include convex constraints. However, all the aforementioned methods have the following limitations: First, they are either restricted to modeling of pairwise interactions or require the order of interactions to be pre-determined. Second, if used in our context, these methods all have to assume that the environmental variables *linearly* affect the input-output relationship, which can be violated in modeling of complex systems in practice. To see this more clearly, we can re-write (1.1) into (1.2):

$$Y = \sum_{j=1}^{q} \omega_j Z_j + \sum_{i=1}^{p} \left(\alpha_i + \sum_{j=1}^{q} \gamma_{ij} Z_j\right) X_i + \varepsilon, \tag{1.2}$$

which shows that the relationship between X_i and Y, characterized by $\alpha_i + \sum_{j=1}^{q} \gamma_{ij} Z_j$, is linearly related to the environmental variables Z_j's.

To relax the linearity assumption, we may use a non-linear function, $f_i(\mathbf{Z})$ to replace the linear function $\alpha_i + \sum_{j=1}^{q} \gamma_{ij} Z_j$. Then, the model becomes a Varying Coefficient (VC) model. Various types of VC models have been developed in the literature. The estimation methods can be broadly classified into spline estimators (Hastie and Tibshirani 1993; Hoover et al., 1998; Chiang et al., 2001), kernel-type estimators (Fan and Zhang, 1999; Xia and Li, 1999), and wavelet estimators (Zhou and You, 2004). Extended work beyond these classic methods exists. For example, Cai et al. (1999) generalized the response variable of VC models to the exponential family. Fan et al. (2003) proposed an

adaptive VC model in which $\mathbf{Z}$ is assumed to be unknown and estimated as a linear combination of input variables. Zhang et al. (2002) introduced a semi-VC model considering the co-existence of varying and constant coefficients in one model. This work was further extended by Hu and Xia (2012) who added an l_1-regularization to the constant coefficients to enable sparse estimation.

A common assumption of VC models is that $f_i(\mathbf{Z})$ is a smooth function of $\mathbf{Z}$. In this research, we have a different focus by aiming to identify a partition of the space of the environmental variables $\mathbf{Z}$, such that the input-output relationship in each subdivision of the partition remains constant whereas this relationship varies across different subdivisions. From the practical point of view, each subdivision of the partition corresponds to a type of environmental condition under which the system is functioning in a specific way. When the environmental condition changes, the system may function differently. A notable difference between the proposed method and VC models is that the former relaxes the smoothness constraint, i.e., it allows unsmooth changes in the input-output relationship at *adjacent* subdivisions of the partition. This relaxation/flexibility has important practical value, because it allows for modeling of systems that are sensitive to environmental changes. For example, in mobile communication networks, the traffic volume-QoS relationship could be remarkably different even when the network is deployed at two adjacent geographical areas, e.g., when the two adjacent areas have different landforms, a valley next to a mountain. In building energy management, it has been observed that there exists tipping points in terms of the environmental conditions. That is, when the temperature, humidity, and air flow rate are within certain ranges, energy consumption is related to building design and operational variables in a specific way. This relationship

may dramatically change if the environmental conditions are outside the ranges. Indeed, VC models can be considered as a special case of the proposed method when the partition by the proposed method is sufficiently fine and the change in the input-output relationship across adjacent subdivisions of the partition satisfies smoothness constraints. Another advantage of the proposed method is that it can take both numerical and categorical environmental variables into consideration, while VC models have inherent difficulty in handling categorical variables. The difficulty is because there is no meaningful measure for the adjacency of the different categories for a categorical variable, and consequently it is meaningless to require smoothness for categorical variables. Last but not least, the proposed method is efficient, while fitting of a VC model with more than one environmental variable can be computationally very challenging.

1.2.2 Tree-based Models

To address the first challenge and model the complex interaction between input variables **X** and environmental variables **Z**, Decision Tree (DT) (Breiman et al., 1984) provides a candidate approach. However, in a DT algorithm, both **X** and **Z** are treated equally as "predictors". There is no differential treatment on their respective controllable and uncontrollable natures as pointed out in the aforementioned second challenge. As a result, DT can produce a model that is good for prediction but not easy for guiding control and interventional actions. Ideally, one would want to use the recursive partitioning scheme of DT to partition the space of environmental variables into disjoint subdivisions. The relationship between input variables and building energy is the same within each subdivision but varies across different subdivisions. In this way, building energy efficiency

can be controlled and optimized by adjusting the indoor variables within each subdivision that defines the specific outdoor environment a building resides in.

Several tree-based algorithms that integrate DT recursive partitioning and regression models have been developed in the existing studies. Some early research works considered regression models in the terminal nodes after the tree is built. Quinlan (1992) developed M5 algorithm that adds linear models to a conventional tree as part of the pruning stage. Torgo (1997) used linear models, k-nearest neighbors, or kernel regressors in terminal nodes, but also added these to a conventional tree after growing. Chaudhuri et al. (1994) proposed trees with linear models with polynomial terms and non-normal response models in the terminal nodes. Most recent tree-based algorithms encompass the idea that fits regression models during the process of tree growing and uses the fitted models to determine the next splits. These algorithms first separate input and splitting variables then fits different types of regression models on the response and input variables within each node. GUIDE (Loh, 2002), which is the first algorithm designed to avoid split variable selection bias, can provide capabilities for fitting simple linear regression models, Poisson models (Loh, 2008), and polynomial quantile regression models (Chaudhuri and Loh, 2002) in the nodes. While using χ^2 test statistics to select splitting variables as in GUIDE to eliminate biases, LOTUS (Chan and Loh, 2004) fits logistic regression in the nodes to specifically model for binary responses. Moreover, LMT (Landwehr et al., 2005), which is designed for binary or multinomial responses, employs boosted logistic model in each node and allows for multiway splits in categorical splitting variables to improve the prediction accuracy. Zeileis et al. (2008) proposed a model-based (MOB) recursive partitioning framework with an overall objective function to induce the tree structure and

fit linear regression models in the nodes. Their approach determines the variable for next splits through statistical tests for parameter instability. Rusch and Zeileis (2013) extended MOB to accommodate for generalized linear models in the nodes which exceeds the versatility of GUIDE.

A limitation of above tree-based models is that the regression model fitting at each subdivision of the partition is independent of the fittings at other subdivisions. This does not leverage the fact that models for different subdivisions may share some similarities so that a joint estimation of the models can enable the model fittings to borrow strength from each other. This is especially advantageous when the sample size of each subdivision is small relative to the dimensionality of regression models.

The idea of joint estimation has been widely applied in regularized regression models in single regression problems. Yuan and Lin (2006) proposed group lasso that allows input variables are grouped through prior knowledge and uses grouped input variables as an unit instead of individual inputs in conventional lasso (Tibshirani, 1996) to select some groups of regression coefficients are exactly zero. Specifically, group lasso applies an L_1 norm penalty over grouped inputs, while using L_2 norm for the input variables within each group, which is so-called L_1/ L_2 penalty. Kim et al. (2006) developed Blockwise Sparse Regression (BSR) that extends the idea of group lasso for general loss functions to include generalized linear models. Their works was generalized by Friedman et al. (2010), who introduced sparse group lasso that not only yields sparsity at group level but also selects variables within a group. Zhao et al. (2009) proposed the Composite Absolute Penalties (CAP) that allows combining different norms from L_1 to L_∞ to construct computationally convenient penalties to account for the nonoverlapping and

overlapping patterns on the features. Jacob et al. (2009) introduced the structural patterns by either defining a union of potentially overlapping groups of features or having a graph that describes how features are connected to each other. Jenatton et al. (2011) proposed a weighting schema that weights each group of features differently in the penalty term to correct the imbalance in estimation of overlapping groups.

It is natural to extend the ideas of group lasso in single regression model to joint estimation of models at multiple different but related domains (e.g., subdivisions in our case). This has been extensively studied in the research area of multitask learning. Multitask learning tends to integrate prediction models from several tasks in a joint manner rather than treating each task individually. To compensate the issue of inaccurate prediction due to limited samples in a single task, multitask learning carries the advantage that it can borrow or share information from other tasks to reduce the variance in model estimation. A key question here is how to define the relatedness between different tasks. One part of previous work investigates the similarity between tasks by imposing a probabilistic framework. Xiong et al. (2006) identified common set of features across tasks by introducing an automatic relevance determination prior on underlying classes with each task and regularizing the variance of model parameters. Zhang et al., (2008) proposed a unified probabilistic framework for multi-task learning in which the relatedness of tasks is characterized by the fact that the task parameters share a common structure through latent variables. Another part of related work studies the common structure between tasks by imposing penalty norms on the task parameters. L_1/ L_2 penalty is frequently used to recover a common set of features that are relevant simultaneously to all the tasks (Argyriou et al., 2008; Obozinski et al., 2009). In their studies, they defined a penalty term that employs L_1

norm over the L_2 norms of each regression parameter vector across all the tasks. The L_1 norm then enforces a group selection among these features. Although the L_1/ L_2 penalty has been shown to be effective in joint feature selection in multitask learning, it fails to enforce any structure on the features among the tasks. The structure can be expressed as the grouping and hierarchical relationships between the task features by utilizing a priori information. The extensions of group lasso mentioned previously (Zhao et al., 2009; Jacob et al., 2009; Jenatton et al., 2011) might be directly applied in multitask learning to impose structural penalty terms on the features across different tasks. Kim and Xing (2010) assumed that a subset of highly correlated tasks could share a common set of features, whereas weakly correlated tasks less likely to be affected by the same features. They encoded the hierarchy structure of the tasks as a tree where each leaf node represents individual task and each internal node serves as the root node of a subtree that includes a group of correlated tasks. Han et al. (2014) proposed a multi-component product-based decomposition for task coefficients where each component maps to one node in a given tree structure. If one component coefficient turns to zero, then the subtree rooted at the corresponding node will be removed from the model, which implies that a specific feature will not be selected in the tasks represented by the leaves in that subtree. However, all these methods either restricted all the task to share a same set of features or required a pre-defined hierarchical/tree structure to describe the relatedness of tasks.

1.3 Summary of Original Contributions

The original contributions of my dissertation research are summarized as follows:

- I propose a new statistical method, called Sparse Partitioned-Regression (SPR), to account for the nonlinear interactions between the design/operational (input) variables of a system and multivariate environments in predicting the system's performance (output). Compared with existing sparse regression models, SPR naturally honors the hierarchical principle and can identify the order and nonlinear pattern of the interactions between input and environmental variables in a data-driven manner. Compared with VC models, SPR relaxes the smoothness constraint in the change of input-output relationship across different environmental conditions, can model both numerical and categorical environmental variables, and is computationally efficient. Additionally, SPR can select small subsets of the environmental variables together with their optimal partitions and the input variables, respectively, that are most relevant to the output variable, and therefore can handle high-dimensional problems.
- I conduct theoretical studies of SPR to derive the oracle inequalities for the SPR estimators which could provide a statistical bound for the difference between the risk of SPR estimators and Bayes' risk. These theoretical studies show that the performance of SPR estimator is almost (up to numerical constants) as good as of an ideal estimator which can be theoretically achieved but is not available in practice.
- I develop another approach, called Tree-Based Structure-Regularized Regression (TBSR), which not only can model the relationship between input variables and response variable by partitioning on environmental variables, but also jointly estimate the models within the adjacent subdivisions by considering

a structured weighting schema on the shared features. TBSR can be considered as an extension of SPR since it overcomes a limitation of SPR that the regression models fitting at each subdivision are independent from each other. In addition, compared with existing structured multitask learning approaches, TBSR doesn't require the tasks share a same set of features and doesn't pre-define a hierarchical/tree structure to describe the relatedness of tasks.

- For applications: (1) I apply SPR on Department of Energy (DOE) *EnergyPlus* datasets to predict building energy consumption. SPR has a significantly higher prediction accuracy than competing methods. The application also helps knowledge discovery for building energy management. (2) I apply TBSR on a dataset collected from a solar-powered house to predict future indoor temperatures. The aim is to establish a more efficient temperature control to reduce Heating, Ventilation, and Air Conditioning (HVAC) systems energy consumption. TBSR achieves better accuracy than competing methods and derive models with good interpretabilities.

1.4 Organization of the Dissertation

The rest of my dissertation is organized as follows: Chapter 2 presents SPR model formulation, model estimation, simulation studies, and a real application. Chapter 3 presents the theoretical studies (i.e., the oracle inequalities) of the SPR model. Chapter 4 presents TBSR model formulation, model estimation, simulation studies, and a real application. Chapter 5 concludes my research and outlines future work.

CHAPTER 2

SPARSE PARTITIONED-REGRESSION MODEL

2.1 Introduction

SPR model is formulated by minimizing an empirical risk function in which the model parameters and partition structure are unknown. SPR automatically searches for a partition of the environmental variables and fits a sparse regression within each subdivision of the partition, in order to fulfill an optimal criterion. Two estimators for the SPR model are proposed: penalized estimator and held-out estimator. Extensive simulation experiments are conducted to demonstrate the better performance of SPR compared with competing methods. Finally, an application of SPR is studied to predict energy consumption by using building design and operational variables, outdoor environmental variables, and their interactions based on Department of Energy (DOE) *EnergyPlus* datasets.

The rest of this chapter is organized as follow: Section 2.2 proposes the model formulation. Section 2.3 presents an algorithm for model estimation. Section 2.4 presents simulation studies. Section 2.5 presents an application of predicting building energy consumption using building design and operational variables together with environmental variables. Section 2.6 is the conclusion.

2.2 Formulation

Consider a training dataset with n samples. Let $\mathbf{x}_k$, $\mathbf{z}_k$, y_k be the measurement on the input, environmental, and output variables of the k-th sample, $k = 1, \dots n$. Consider a partition of the space of the environmental variables into n_S disjoint subdivisions, i.e., $\boldsymbol{\mathcal{S}} = \{\mathcal{S}^{(1)}, \dots, \mathcal{S}^{(n_S)}\}$. Each sample belongs to one and only one subdivision depending on its environmental variables. That is, the k-th sample belongs to the r-th subdivision if $\mathbf{z}_k \in \mathcal{S}^{(r)}$. Within each subdivision, the relationship between the input and output variables is characterized by a model $\mathrm{y}_k = f(\mathbf{x}_k; \boldsymbol{\theta}^{(r)})$ with parameter set $\boldsymbol{\theta}^{(r)}$. The exact form of the model is unknown but can be estimated from data. To assess how good the estimation is, we can define a risk function between the observed y_k and the estimated model $\hat{f}(\mathbf{x}_k; \boldsymbol{\theta}^{(r)})$, $L(\mathrm{y}_k, \hat{f}(\mathbf{x}_k; \boldsymbol{\theta}^{(r)}))$. Averaging over all the samples in the training dataset, we can obtain the empirical risk function as follows:

$$\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\theta}) = \frac{1}{n}\sum_{k=1}^{n}\sum_{r=1}^{n_S} L\left(y_k, \hat{f}(\mathbf{x}_k; \boldsymbol{\theta}^{(r)})\right) \cdot I(\mathbf{z}_k \in \mathcal{S}^{(r)}). \quad (2.1)$$

$\boldsymbol{\theta} = \{\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(n_S)}\}$. $I(\cdot)$ is an indicator function.

In this paper, both the partition $\boldsymbol{\mathcal{S}}$ and the model parameters $\boldsymbol{\theta}$ are treated as unknown. To estimate them, simply minimizing the empirical risk in (2.1) will cause overfitting, i.e., finer partitions would always be preferred. To address this problem, we propose two estimators, a penalized estimator and a held-out estimator.

Definition 1: The penalized estimator is defined as:

$$\widehat{\boldsymbol{\mathcal{S}}}, \widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\mathcal{S}}, \boldsymbol{\theta}}{\operatorname{argmin}}\{\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\theta}) + \lambda_{\mathcal{S}}\, pen(\boldsymbol{\mathcal{S}})\}$$

$pen(\boldsymbol{\mathcal{S}})$ is a measure for the complexity of the partition. The finer the partition, the higher the complexity. $\lambda_{\mathcal{S}}$ is a penalty parameter. Alternatively, if there are sufficient training

samples, we may divide the entire training dataset into a training set and a validation set consisting of n_1 and n_2 samples, respectively. Given $\boldsymbol{\mathcal{S}}$, we can obtain an estimate for $\boldsymbol{\theta}$ that minimizes the empirical risk evaluated on the training set alone, i.e.,

$$\widehat{\boldsymbol{\theta}}_{tr} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \hat{R}_{tr}(\boldsymbol{\mathcal{S}}, \boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n_1} \sum_{k=1}^{n_1} \sum_{r=1}^{n_S} L\left(y_k, \hat{f}\left(\mathbf{x}_k; \boldsymbol{\theta}^{(r)}\right)\right) \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right).$$

Then, this estimate is plugged into the empirical risk evaluated on the held-out validation set,

$$\hat{R}_{val}\left(\boldsymbol{\mathcal{S}}, \widehat{\boldsymbol{\theta}}_{tr}\right) = \frac{1}{n_2} \sum_{k=1}^{n_2} \sum_{r=1}^{n_S} L\left(y_k, \hat{f}\left(\mathbf{x}_k; \widehat{\boldsymbol{\theta}}_{tr}^{(r)}\right)\right) \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right),$$

which measures the generalization error of the estimate and the partition $\boldsymbol{\mathcal{S}}$. Minimizing this generalization error yields the held-out estimator.

Definition 2: The held-out estimator is defined as:

$$\widetilde{\boldsymbol{\mathcal{S}}}, \widetilde{\boldsymbol{\theta}} = \underset{\boldsymbol{\mathcal{S}}, \boldsymbol{\theta}}{\operatorname{argmin}}\, \hat{R}_{val}(\boldsymbol{\mathcal{S}}, \widehat{\boldsymbol{\theta}}_{tr}).$$

The afore-proposed framework can be used to for a numerical or a categorical output variable, resulting in a predictive model or a classification model, respectively. In the predictive model, the relationship between the input and output variables can be characterized by a linear regression, i.e., $\mathrm{y}_k = \mathbf{x}_k^T \boldsymbol{\alpha}^{(r)} + \varepsilon^{(r)}$, $\varepsilon^{(r)} \sim N\left(0, \sigma_\varepsilon^{2(r)}\right)$. A typical risk function for a linear regression is the negative log-likelihood function, using which the empirical risk function in (2.1) can be written as:

$$\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\theta}) = \hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}, \boldsymbol{\sigma}_\varepsilon^2) = \frac{1}{n} \sum_{k=1}^{n} \sum_{r=1}^{n_S} \left\{ \frac{\left(y_k - \mathbf{x}_k^T \boldsymbol{\alpha}^{(r)}\right)^2}{\sigma_\varepsilon^{2(r)}} + log \sigma_\varepsilon^{2(r)} \right\} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right), \qquad (2.2)$$

where $\boldsymbol{\alpha} = \left\{\boldsymbol{\alpha}^{(1)}, \dots, \boldsymbol{\alpha}^{(n_S)}\right\}$and $\boldsymbol{\sigma}_\varepsilon^2 = \left\{\sigma_\varepsilon^{2(1)}, \dots, \sigma_\varepsilon^{2(n_S)}\right\}$ are the model parameters. When the input variables are high-dimensional, a l_1-regularized negative log-likelihood function can be used and (2.2) can be further written as:

$$\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}, \boldsymbol{\sigma}_\varepsilon^2) = \sum_{r=1}^{n_S} \left\{ \frac{1}{n} \sum_{k=1}^{n} \left\{ \frac{\left(y_k - \mathbf{x}_k^T \boldsymbol{\alpha}^{(r)}\right)^2}{\sigma_\varepsilon^{2(r)}} + log\sigma_\varepsilon^{2(r)} \right\} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) + \lambda_\alpha^{(r)} \left\|\boldsymbol{\alpha}^{(r)}\right\|_1 \right\} . \tag{2.3}$$

$\|\cdot\|_1$ denotes the l_1-norm, which was used in lasso and is known to enforce sparsity in model estimation. Other well-known sparsity-induced regularizations such as those used in fused-lasso and group-lasso can also be adopted in (2.3) depending on the structure of the input variables. $\lambda_\alpha^{(r)}$ is a regularization parameter. Furthermore, using (2.3) in Definitions 1 and 2, a sparse penalized estimator and a sparse held-out estimator for a predictive model can be obtained, respectively.

In a classification model (i.e., when the output variable is categorical), the relationship between the input and output variables can be characterized by a multinomial logistic regression, which links the probability of the output variable being in the m-th class with the input variables in the form of $P(\mathrm{y}_k = m) = \frac{exp\left(\mathbf{x}_k^T \boldsymbol{\beta}_m^{(r)}\right)}{\sum_{l=1}^{M} exp\left(\mathbf{x}_k^T \boldsymbol{\beta}_l^{(r)}\right)}$, $m = 1, \dots, M$. A typical risk function for a multinomial logistic regression is the negative log-likelihood function, using which the empirical risk function in (2.1) can be written as:

$$\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\theta}) = \hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta}) = \frac{1}{n} \sum_{k=1}^{n} \sum_{r=1}^{n_S} \left\{ log \sum_{l=1}^{M} exp\left(\mathbf{x}_k^T \boldsymbol{\beta}_l^{(r)}\right) - \sum_{l=1}^{M} I(\mathrm{y}_k = l)\, \mathbf{x}_k^T \boldsymbol{\beta}_l^{(r)} \right\} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right), \tag{2.4}$$

where $\boldsymbol{\beta} = \left\{\boldsymbol{\beta}_1^{(1)}, \ldots, \boldsymbol{\beta}_M^{(1)}, \boldsymbol{\beta}_1^{(2)}, \ldots, \boldsymbol{\beta}_M^{(2)}, \ldots, \boldsymbol{\beta}_1^{(n_S)}, \ldots, \boldsymbol{\beta}_M^{(n_S)}\right\}$ are the model parameters. When the input variables are high-dimensional, a l_1-regularized negative log-likelihood function can be used and (2.4) can be further written as:

$$\hat{R}(\boldsymbol{S}, \boldsymbol{\beta}) = \sum_{r=1}^{n_S} \begin{Bmatrix} \frac{1}{n}\sum_{k=1}^{n}\left\{log\sum_{l=1}^{M} exp\left(\mathbf{x}_k^T\boldsymbol{\beta}_l^{(r)}\right) - \sum_{l=1}^{M} I(\mathrm{y}_k = l)\,\mathbf{x}_k^T\boldsymbol{\beta}_l^{(r)}\right\} \\ \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) + \lambda_\beta^{(r)}\sum_{l=1}^{M}\left\|\boldsymbol{\beta}_l^{(r)}\right\|_1 \end{Bmatrix}. \quad (2.5)$$

Furthermore, using (2.5) in Definitions 1 and 2, a sparse penalized estimator and a sparse held-out estimator for a classification model can be obtained, respectively.

In summary, the proposed SPR consists of four models: a sparse penalized estimator for prediction, a sparse held-out estimator for prediction, a sparse penalized estimator for classification, and a sparse held-out estimator for classification. Using the first model of SPR as an example, Proposition 1 shows that SPR obeys the weak hierarchy and Proposition 2 shows that SPR obeys the strong hierarchy under *some* conditions. These properties also hold for the other three models of SPR. The first model of SPR takes the form of $Y = \sum_{r=1}^{n_S}\left\{\mathbf{X}^T\boldsymbol{\alpha}^{(r)} + \varepsilon^{(r)}\right\}I\left(\mathbf{Z} \in \mathcal{S}^{(r)}\right)$. Let $\hat{\mathcal{S}}^{(r)}$ denote the r-th subdivision and $\widehat{\boldsymbol{\alpha}}^{(r)} = \left(\hat{\alpha}_0^{(r)}, \hat{\alpha}_1^{(r)} \ldots, \hat{\alpha}_p^{(r)}\right)^T$ denote the linear coefficients in the r-th subdivision produced by the sparse penalized estimator of the first model.

Proposition 1: SPR obeys the weak hierarchy, i.e., if there is an interaction between the i^*-th input variable, X_{i^*}, and the environmental variables in the r^*-th subdivision, $\mathbf{Z} \in \hat{\mathcal{S}}^{(r^*)}$, then the main effect of $\mathbf{Z} \in \hat{\mathcal{S}}^{(r^*)}$ must exist.

Proof:

The estimated SPR can be written as

$$\hat{Y} = \sum_{r=1}^{n_S} \mathbf{X}^T \hat{\boldsymbol{\alpha}}^{(r)} I\left(\mathbf{Z} \in \hat{\mathcal{S}}^{(r)}\right)$$

$$= \sum_{r=1}^{n_S} \hat{\alpha}_0^{(r)} I\left(\mathbf{Z} \in \hat{\mathcal{S}}^{(r)}\right) + \sum_{r=1}^{n_S} \sum_{i=1}^{p} \hat{\alpha}_i^{(r)} X_i I\left(\mathbf{Z} \in \hat{\mathcal{S}}^{(r)}\right).$$

Suppose there is an interaction between X_{i^*} and $\mathbf{Z} \in \hat{\mathcal{S}}^{(r^*)}$, which means that $\hat{\alpha}_{i^*}^{(r^*)} \neq 0$ and $I\left(\mathbf{Z} \in \hat{\mathcal{S}}^{(r^*)}\right) = 1$. Then, $\hat{\alpha}_0^{(r^*)} I\left(\mathbf{Z} \in \hat{\mathcal{S}}^{(r^*)}\right) = \hat{\alpha}_0^{(r^*)} \neq 0$, i.e., the main effect of $\mathbf{Z} \in \hat{\mathcal{S}}^{(r^*)}$ exists. Here, $\hat{\alpha}_0^{(r^*)} \neq 0$ because the l_1-penalty used in the sparse estimator follows the common practice of sparse regressions (e.g., lasso) that the intercept will not be penalized so it is always non-zero. Δ

Proposition 2: SPR obeys the strong hierarchy, i.e., if there is an interaction between X_{i^*} and $\mathbf{Z} \in \hat{\mathcal{S}}^{(r^*)}$, then both the main effects of $\mathbf{Z} \in \hat{\mathcal{S}}^{(r^*)}$ and X_{i^*} must exist, under a necessary and sufficient condition that X_{i^*} has a non-zero coefficient in every subdivision of the partition, i.e., in $\hat{\mathcal{S}}^{(r)}$ for $\forall r \in \{1, \dots, n_S\}$.

Proof:

Suppose there is an interaction between X_{i^*} and $\mathbf{Z} \in \hat{\mathcal{S}}^{(r^*)}$, which means that $\hat{\alpha}_{i^*}^{(r^*)} \neq 0$ and $I\left(\mathbf{Z} \in \hat{\mathcal{S}}^{(r^*)}\right) = 1$. Using Proposition 1, we know that the main effect of $\mathbf{Z} \in \hat{\mathcal{S}}^{(r^*)}$ must exists. Therefore, we only need to prove that the main effect of X_{i^*} exists under the condition described in Proposition 2. To prove that the condition is necessary, suppose the

main effect of X_{i^*} exists. This means that X_{i^*} always has an influence on Y regardless of the values of **Z**. In SPR, this further means that X_{i^*} always has an influence on Y regardless of the subdivisions (i.e., X_{i^*} has a non-zero coefficient in every subdivision), because the values of **Z** are different between subdivisions. To prove that the condition is sufficient, suppose X_{i^*} has a non-zero coefficient in every subdivision, i.e., $\hat{\alpha}_{i^*}^{(r)} \neq 0$ for $\forall r \in \{1, \dots, n_S\}$. Then, we can always decompose $\hat{\alpha}_{i^*}^{(r)}$ into a common component shared by all the subdivisions, $\hat{\alpha}_{i^*}$, and a subdivision-specific component, $\Delta\hat{\alpha}_{i^*}^{(r)}$, with $\hat{\alpha}_{i^*} \neq 0$ and $\Delta\hat{\alpha}_{i^*} \neq 0$. $\hat{\alpha}_{i^*} \neq 0$ means that the main effect of X_{i^*} exists. Δ

2.3 Algorithm

Among the four models proposed in the previous section, we will focus on "the held-out estimator for classification" in describing the algorithm for model estimation. The algorithms for estimating the other three models share a similar procedure. The goal of the algorithm is to find an optimal partition of the space of the environmental variables and a multinomial logistic regression between the input and output variables within each subdivision of the partition based on a training set, such that the empirical risk evaluated on a held-out validation set is minimized. To achieve this goal, an exhaustive search for the optimal partition is computationally infeasible. We propose a computationally efficient algorithm based on recursive partitioning. The basic idea is to first find a variable within the set of environmental variables $\mathbf{Z}$ and a splitting point of that variable, which best split the training set into two sub-groups ("best" in terms of optimizing a splitting criterion). Then, this process is repeated within each sub-group identified in the previous step until a stopping criterion is met.

SPR looks similar to the recursive partitioning used to build a Classification and Regression Tree (CART) (Breiman et al., 1984). The major difference is in the splitting criterion. In CART, the splitting variable and splitting point are selected to achieve the greatest reduction in an impurity measure. A sub-group is pure if it only consists of samples belonging to the same class of the output variable Y. The higher the mix of different classes, the more impure the sub-group is. Typical impurity measures include Gini Index, entropy, and others. In our recursive partitioning algorithm, the splitting criterion considers not only Y but also the input variables $\mathbf{X}$, i.e., the relationship between Y and $\mathbf{X}$ characterized by a multinomial logistic regression. It selects the splitting variable and splitting point that

achieve the greatest reduction in the empirical risk evaluated on the validation set. Specifically, let s denote a subdivision in the current partition $\boldsymbol{\mathcal{S}}^c$. We can compute the empirical risk of this subdivision evaluated on the validation set, $\hat{R}_{val}(\boldsymbol{\mathcal{S}}^c, \hat{\boldsymbol{\theta}}_{tr}^{(s)})$. To find which environmental variable to use for further splitting s, we search through all the environmental variables in $\mathbf{Z}$. For each $Z_j \in \mathbf{Z}$, the subdivision can be split into a left region defined by $Z_j \leq z_j$ and a right region defined by $Z_j > z_j$. z_j is a candidate splitting point. Then, we compute the empirical risk of each region evaluated on the validation set, $\hat{R}_{val}(\boldsymbol{\mathcal{S}}^{c+1}, \hat{\boldsymbol{\theta}}_{tr}^{(Z_j \leq z_j)})$ and $\hat{R}_{val}(\boldsymbol{\mathcal{S}}^{c+1}, \hat{\boldsymbol{\theta}}_{tr}^{(Z_j > z_j)})$, and a reduction in the empirical risk as

$$\Delta\hat{R}_{val}^{j} = n_s\hat{R}_{val}\left(\boldsymbol{\mathcal{S}}^c, \hat{\boldsymbol{\theta}}_{tr}^{(s)}\right) - n_L\hat{R}_{val}\left(\boldsymbol{\mathcal{S}}^{c+1}, \hat{\boldsymbol{\theta}}_{tr}^{\left(Z_j \leq z_j\right)}\right) - n_R\hat{R}_{val}(\boldsymbol{\mathcal{S}}^{c+1}, \hat{\boldsymbol{\theta}}_{tr}^{(Z_j > z_j)}). \quad (2.6)$$

n_s, n_L, and n_R are sample sizes of the subdivision s, left region, and right region, respectively. The environmental variable and the splitting point with the largest reduction $\Delta\hat{R}_{val}^{j}$ are selected to split s. If no positive reduction is found, s will not be split and the algorithm stops. Another consideration in the stopping criterion is that the sample size of the subdivision s reaches a pre-defined minimum number.

Next, we discuss two technical details on the splitting criterion of our algorithm. One is regarding the selection of candidate splitting points for a categorical environmental variable. Assuming that the variable has B categories, there are $2^{(B-1)} - 1$ possible splits. Environmental variables with a large number of categories are common. For example, in studying the relationship between building design variables and energy consumption, an important environmental variable is the geographical location of a building. If using "states" in the U.S. to describe the location, $B = 50$, resulting in $2^{49} - 1$ possible splits. To reduce the computational burden, we propose an alternative transformation-based approach: First,

we transform the B categories of the environmental variable into B ordinal numbers according to the average output variable for each category computed on the training set. Then, we consider every possible binary split of the ordered sequence of the B ordinal numbers as a candidate split. This results in $(B-1)$ candidate splitting points, which compose a much smaller subset of the $2^{(B-1)}-1$ splitting points. A nice property of this approach is that it guarantees that the optimal split within the $2^{(B-1)}-1$ splitting points is included in the subset of $(B-1)$ candidate splitting points.

The other technical detail of the proposed splitting criterion is regarding the computation of the $\hat{R}_{val}(\cdot)$ in (2.6). Take $\hat{R}_{val}\left(\boldsymbol{\mathcal{S}}^c, \widehat{\boldsymbol{\theta}}_{tr}^{(s)}\right)$ as an example. Because we focus on the classification model, $\widehat{\boldsymbol{\theta}}_{tr}^{(s)}$ consists of coefficients of a multinomial logistic regression for subdivision s estimated from the training set, i.e., $\widehat{\boldsymbol{\theta}}_{tr}^{(s)} = \widehat{\boldsymbol{\beta}}_{tr}^{(s)} = \left\{\widehat{\boldsymbol{\beta}}_{tr,1}^{(s)}, \ldots, \widehat{\boldsymbol{\beta}}_{tr,M}^{(s)}\right\}$. To obtain $\widehat{\boldsymbol{\beta}}_{tr}^{(s)}$, we minimize a l_1-regularized negative log-likelihood function, i.e.,

$$\widehat{\boldsymbol{\beta}}_{tr}^{(s)} = \underset{\boldsymbol{\beta}_{tr}^{(s)}}{\operatorname{argmin}} \hat{R}_{tr}\left(\boldsymbol{\mathcal{S}}^c, \boldsymbol{\beta}_{tr}^{(s)}\right)$$

$$= \underset{\boldsymbol{\beta}_{tr}^{(s)}}{\operatorname{argmin}} \left\{ \begin{array}{c} \frac{1}{n_1}\sum_{k=1}^{n_1}\left\{log\sum_{l=1}^{M} exp\left(\mathbf{x}_k^T\boldsymbol{\beta}_l^{(s)}\right) - \sum_{l=1}^{M} I(\mathrm{y}_k = l)\,\mathbf{x}_k^T\boldsymbol{\beta}_l^{(s)}\right\} \\ \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(s)}\right) + \lambda_\beta^{(s)}\sum_{l=1}^{M}\left\|\boldsymbol{\beta}_l^{(s)}\right\|_1 \end{array} \right\}. \quad (2.7)$$

For a given $\lambda_\beta^{(s)}$, (2.7) is a convex optimization that can be solved efficiently. To find the optimal $\lambda_\beta^{(s)}$, we can conduct a line search on a series of values for $\lambda_\beta^{(s)}$. Under each value, we solve the convex optimization in (2.7), use the estimated $\widehat{\boldsymbol{\beta}}_{tr}^{(s)}$, i.e., the training model, to classify the samples in the validation set, and compute a misclassification error rate. The

optimal $\lambda_{\beta}^{(s)}$ is one under which the misclassification error rate is minimized. Furthermore, because of the l_1-regularization in (2.7), the $\widehat{\boldsymbol{\beta}}_{tr}^{(s)}$ estimated under the optimal $\lambda_{\beta}^{(s)}$ will be sparse with many zero elements, and the non-zero elements will suffer from a shrinking effect by having a magnitude smaller than what they are supposed to be. To correct this estimation bias, we re-estimate the $\widehat{\boldsymbol{\beta}}_{tr}^{(s)}$ in (2.7) without the l_1-regularization, but enforcing the sparse pattern obtained from the previous l_1-regularized estimation. Finally, we plug the re-estimated $\widehat{\boldsymbol{\beta}}_{tr}^{(s)}$ into $\hat{R}_{val}\left(\boldsymbol{\mathcal{S}}^c, \widehat{\boldsymbol{\theta}}_{tr}^{(s)}\right)$.

We conclude this section by presenting the major steps of the proposed algorithm in estimating SPR. The input to the algorithm includes a training set and a validation set on the input, environmental, and output variables, **X**, **Z**, and Y, and a minimum sample size requirement n_{min}. At the c-th step of the recursive partitioning, let $\boldsymbol{\mathcal{S}}^c$ be the partition of the space of the environmental variables. For each subdivision in the partition, $s \in \boldsymbol{\mathcal{S}}^c$, perform the following steps:

Step 1: if the sample size of the subdivision s in the training set is smaller than n_{min}, stop splitting this subdivision; otherwise, proceed to Step 2.

Step 2: fit a multinomial logistic regression model between the input and output variables and estimate the l_1-regularized regression coefficients $\widehat{\boldsymbol{\beta}}_{tr}^{(s)}$ according to (2.7) using the training set. The optimal $\lambda_{\beta}^{(s)}$ in (2.7) is selected by minimizing the misclassification error rate of applying the training model to classify the samples in the validation set.

Step 3: re-estimate the $\widehat{\boldsymbol{\beta}}_{tr}^{(s)}$ using (2.7) without the l_1-regularization, but enforcing the sparse pattern obtained from Step 2.

Step 4: use the re-estimated $\widehat{\boldsymbol{\beta}}_{tr}^{(s)}$ to compute the empirical risk evaluated on the validation set, $\hat{R}_{val}\left(\boldsymbol{S}^{c}, \widehat{\boldsymbol{\theta}}_{tr}^{(s)}\right)$.

Step 5: for each environmental variable $Z_j \in \mathbf{Z}$ and each candidate splitting point z_j, split the subdivision s into a left region defined by $Z_j \leq z_j$ and a right region defined by $Z_j > z_j$. If Z_j is a categorical variable, use the aforementioned transformation-based approach to select the candidate splitting points. For each region, repeat Steps 2-4 and obtain $\hat{R}_{val}\left(\boldsymbol{S}^{c+1}, \widehat{\boldsymbol{\theta}}_{tr}^{(Z_j \leq z_j)}\right)$ and $\hat{R}_{val}(\boldsymbol{S}^{c+1}, \widehat{\boldsymbol{\theta}}_{tr}^{(Z_j > z_j)})$. Use (2.6) to compute the empirical risk reduction $\Delta\hat{R}_{val}^{j}$.

Step 6: if no positive $\Delta\hat{R}_{val}^{j}$ is found, stop splitting the subdivision s; otherwise, split the subdivision using the environmental variable and splitting point with the largest $\Delta\hat{R}_{val}^{j}$.

The output from the algorithm is a partition of the space of the environmental variables, and a multinomial logistic regression model between the input and output variables for each subdivision of the partition. To classify a new sample, e.g., the $(n+1)$-th sample, we first use the environmental variables of this sample, $\mathbf{z}_{n+1}$, to find which subdivision this sample belongs to. Then, we use the input variables, $\mathbf{x}_{n+1}$, in the multinomial logistic regression of this subdivision to predict the class membership of the output variable, $\hat{y}_{n+1}$. The SPR algorithm as presented here is programmed using the R software.

It is worth mentioning that when the sample size allows, it would be better to separate the data into a training and two validation sets, with one validation set used to tune the penalty parameter and the other to find the splitting point. This would reduce the

potential risk of overfitting. When the sample size is limited, we could use a single validation set to serve the two purposes like what the current algorithm is designed to do. This may not be much of an issue as our simulation and application studies show that the algorithm grants a good accuracy on a separate test set. However, a cautious strategy for avoiding the potential overfitting with the current algorithm may be to increase the n_{min},. This issue is worthy of more in-depth future investigation.

2.4 Simulation Studies

In this section, we present the performance of our SPR as a classification and a predictive model, respectively. We focus on the held-out estimator due to its empirically better performance than the penalized estimator. In what follows, we first describe the data generation process of the environmental, input, and output variables.

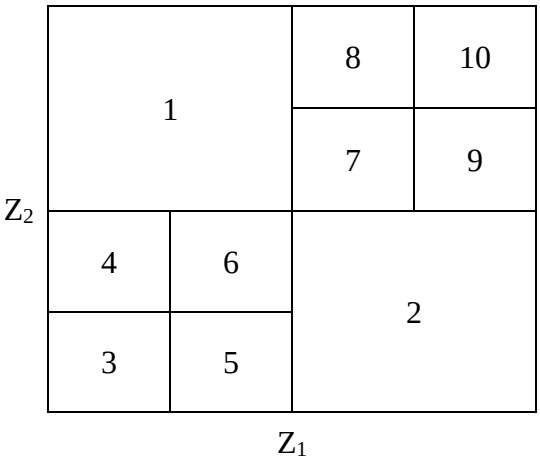

Figure 1: Subdivisions of the partition by environmental variables Z_1 and Z_2

We consider five environmental variables that are uniformly distributed on the unit hypercube $[0,1]^5$. We further assume that the first two environmental variables, Z_1 and Z_2,

are truly used in partitioning the space of the environmental variables into subdivisions, while the rest three variables are noise. Specifically, Z_1 and Z_2 partition the space into 10 subdivisions by median splits, as shown in Figure 1. In each subdivision, we consider 100 input variables and generate samples for the input variables from a multivariate normal distribution $N(\mathbf{0}, \mathbf{\Sigma}_{100\times100})$. Each element of $\mathbf{\Sigma}_{100\times100}$ is set to be $\sigma_{ij} = 0.5^{|i-j|}, i, j = 1, \ldots, 100$, to account for the potential correlation between input variables. To further generate samples for the output variable within each subdivision, we use a linear regression if the output variable is numerical and a multinomial logistic regression if the output variable is categorical. Without loss of generality, we focus on binary output variables in this section. In the linear/logistic regression, we assume that five out of the 100 input variables have non-zero coefficients sampled from $N(0,1) + 3$, while the remaining input variables have zero coefficients (i.e., they are noise). For generality, we randomly select five input variables to have non-zero coefficients in each subdivision.

Following the afore-described data generation process, we generate 5000 samples to include in a training set and another 5000 samples to include in a held-out validation set. Under this setting, the smallest subdivision includes around 300 samples in the training and validation sets, respectively, which is a limited-sample scenario compared with 100 input variables. Then, we apply the algorithm presented at the end of Section 3 to the data. The result from the algorithm is a partition of the space of the environmental variables and a fitted l_1-regularized linear/logistic regression between the input and output variables within each subdivision of the partition. This entire process from data generation to model fitting is repeatedly run for 100 times. Figure 2 shows the result from one simulation run of the SPR predictive model, in which the partition is represented by a tree whose leaf

nodes correspond to the subdivisions of the partition and internal nodes describe the recursive partitioning process. Coefficients of the fitted l_1- regularized regression for each leaf/internal node are represented by a bar chart. Furthermore, Figure 3 stacks up the coefficients of the fitted l_1 - regularized regressions for all the nodes to facilitate comparison across the nodes and discovery of patterns. Additionally, to test the performance of our algorithm under smaller sample sizes, we run another simulation with 2000 samples. Under this setting, the smallest subdivision includes around 120 samples in the training and validation sets, respectively, close to the number of input variables.

Furthermore, for comparison purposes, we apply two competing methods to the same simulation datasets as SPR: a generalized linear model with l_1-regularization (GLM-lasso) and CART. In the GLM-lasso, we include main effects of environmental and input variables as well as the two-way interactions between each environmental variable and each input variable. In CART, we put in all the environmental and input variables and let the CART algorithm decide which variables to use and the interaction structure that best fit the data. To tune the penalty parameter for GLM-lasso and the meta-parameters for CART (i.e., the minimize node size and cost parameter), we perform a grid search over a wide range of the tuning parameters and report the best performance for each method. We apply the three methods on another independently simulated test set consisting of 2000 samples.

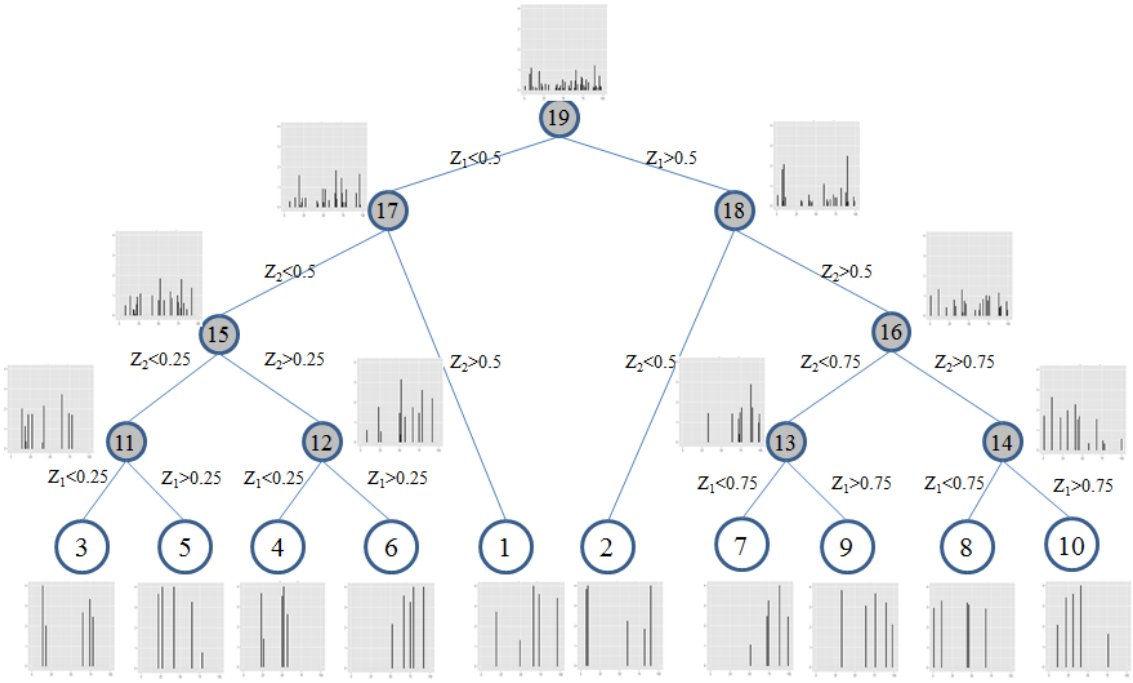

Figure 2: Result from the SPR predictive model for one simulation run (tree represents partition and bar charts represent coefficients of the l_1- regularized regressions)

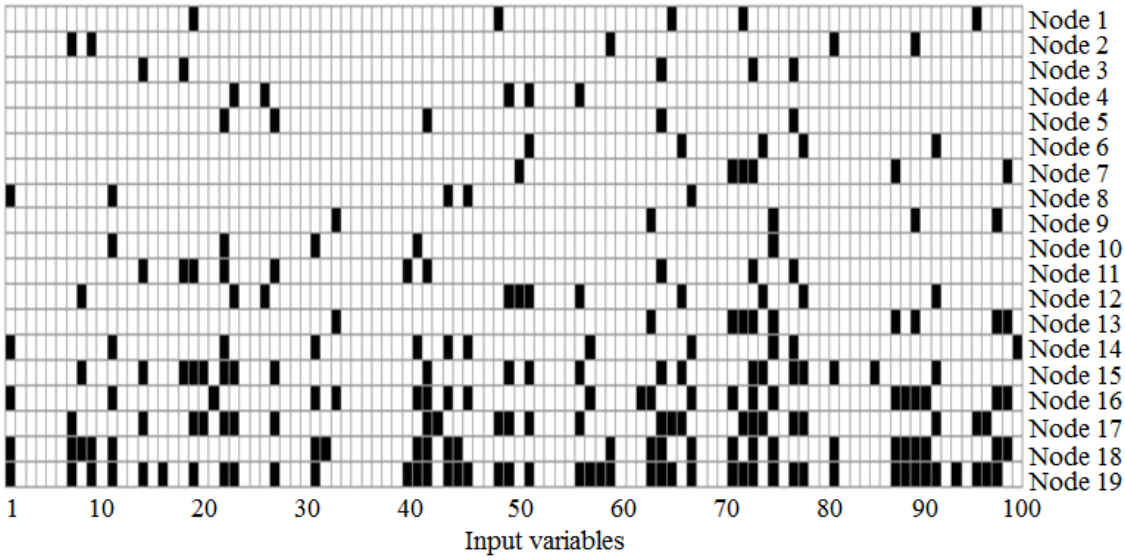

Figure 3: Coefficients of the l_1- regularized regressions fitted for all the nodes in the tree of Figure 2 (black and white represent non-zero and zero coefficients, respectively)

Our results are presented as follows: First, we compare the three methods in the accuracy of selecting the environmental variables. We use two metrics: $precision_e$ measures the fraction of environmental variables selected by a method that are the ground-truth partitioning variables; $recall_e$ measures the fraction of the ground-truth partitioning variables that are selected by a method. In GLM-lasso, we consider an environmental variable as "selected" when it is included as either a main effect or in an interaction effect. Table 1 shows the $precision_e$ and $recall_e$ under two different samples sizes for the predictive and classification models of SPR in comparison with CART and GLM-lasso.

Table 1: Accuracies of SPR, GLM-lasso, and CART in selecting the environmental variables (average (standard deviation) over 100 simulation runs)

Predictive model	Sample size = 2000		Sample size =5000	
	$precision_e$	$recall_e$	$precision_e$	$recall_e$
SPR	0.85(0.19)	1.00(0.00)	0.95(0.12)	1.00(0.00)
GLM-lasso	0.01(0.00)	1.00(0.00)	0.01(0.00)	1.00(0.00)
CART	0.18(0.06)	0.98(0.14)	0.22(0.04)	1.00(0.00)

Classification model	Sample size = 2000		Sample size =5000	
	$precision_e$	$recall_e$	$precision_e$	$recall_e$
SPR	0.81(0.20)	1.00(0.00)	0.90(0.16)	1.00(0.00)
GLM-lasso	0.08(0.05)	1.00(0.00)	0.03(0.02)	1.00(0.00)
CART	0.12(0.07)	0.69(0.33)	0.16(0.03)	0.97(0.12)

Then, we examine the variable selection accuracy of SPR in terms of the input variables. This accuracy is computed separately for each subdivision (i.e., leaf node) in Figure 1, because the subdivisions do not have the same set of input variables with non-zero coefficients. We use two common metrics for the accuracy: $precision_I$ measures the fraction of input variables found by our algorithm to have non-zero coefficients that truly have non-zero coefficients; $recall_I$ measures the fraction of the input variables with truly

non-zero coefficients that are found by our algorithm to have non-zero coefficients. Table 2 shows the subdivision-specific $precision_I$ and $recall_I$ under two different samples sizes for the predictive and classification models.

Table 2: Accuracy of SPR in selecting the input variables (average (standard deviation) over the simulation runs where the subdivisions in Figure 1 are recovered) for (a) predictive model, (b) classification model

(a**) Predictive model**	Sample size = 2000		Sample size =5000	
	$precision_I$	$recall_I$	$precision_I$	$recall_I$
Subdivision 1	0.98(0.06)	1.00(0.00)	1.00(0.02)	1.00(0.00)
Subdivision 2	0.98(0.02)	1.00(0.00)	1.00(0.02)	1.00(0.00)
Subdivision 3	0.89(0.18)	1.00(0.00)	0.97(0.07)	1.00(0.00)
Subdivision 4	0.88(0.17)	1.00(0.00)	0.98(0.06)	1.00(0.00)
Subdivision 5	0.88(0.16)	1.00(0.00)	0.97(0.08)	1.00(0.00)
Subdivision 6	0.89(0.17)	1.00(0.00)	0.98(0.08)	1.00(0.00)
Subdivision 7	0.88(0.14)	1.00(0.00)	0.97(0.08)	1.00(0.00)
Subdivision 8	0.87(0.15)	1.00(0.00)	0.98(0.07)	1.00(0.00)
Subdivision 9	0.89(0.13)	1.00(0.00)	0.97(0.07)	1.00(0.00)
Subdivision 10	0.89(0.14)	1.00(0.00)	0.97(0.07)	1.00(0.00)

(**b) Classification model**	Sample size = 2000		Sample size =5000	
	$precision_I$	$recall_I$	$precision_I$	$recall_I$
Subdivision 1	0.94(0.09)	1.00(0.00)	0.99(0.08)	1.00(0.00)
Subdivision 2	0.94(0.07)	1.00(0.00)	0.96(0.12)	1.00(0.00)
Subdivision 3	0.84(0.13)	1.00(0.00)	0.92(0.13)	1.00(0.00)
Subdivision 4	0.86(0.12)	1.00(0.00)	0.90(0.17)	1.00(0.00)
Subdivision 5	0.87(0.19)	1.00(0.00)	0.94(0.10)	1.00(0.00)
Subdivision 6	0.90(0.15)	1.00(0.00)	0.91(0.15)	1.00(0.00)
Subdivision 7	0.84(0.18)	1.00(0.00)	0.92(0.16)	1.00(0.00)
Subdivision 8	0.90(0.12)	1.00(0.00)	0.97(0.09)	1.00(0.00)
Subdivision 9	0.87(0.17)	1.00(0.00)	0.91(0.14)	1.00(0.00)
Subdivision 10	0.84(0.14)	1.00(0.00)	0.92(0.15)	1.00(0.00)

Next, we present the prediction accuracies of SPR, GLM-lasso, and CART on an independently simulated test set in Table 3. The prediction accuracy for a numerical output variable is measured by a Mean Squared Prediction Error (MSPE) and that for a categorical output variable is measured by the classification accuracy. Additionally, we would like to compare SPR with the VC model (Hastie and Tibshirani 1993). Because VC is computationally very slow, we train it on the simulation dataset with 2000 samples and only include the two true environmental variables. The MSPE of VC is 77.40, which is substantially higher than the MSPE of SRP (0.09).

Table 3: Prediction accuracy of SPR in comparison with GLM-lasso and CART

	Sample size = 2000			Sample size =5000		
	SPR	GLM-lasso	CART	SPR	GLM-lasso	CART
Predictive model (MSPE)	2.94	86.48	119.05	0.09	81.89	115.21
Classification model (classification accuracy)	0.90	0.68	0.62	0.94	0.70	0.62

Furthermore, we demonstrate the oracle properties of SPR that were discussed in Section 4 in comparison with GLM-lasso and CART. Specifically, the excess risk of each method is computed by taking the difference between the empirical risk of the method and the Bayes' risk. The Bayes' risk is computed from the ground-truth model. Therefore, the smaller the excess risk, the better oracle property a method has. To compute the empirical risk for GLM-lasso, we follow the paper by Geer (2008). Since both SPR and GLM-lasso use the negative log-likelihood function (NLLF) as the empirical risk, we would like to use NLLF for CART for consistency. However, the NLLF for CART does not exist. To tackle

this problem, we follow a similar idea to that by Friedman and Popescu (2008) and convert the tree trained by CART into an empirical regression that includes the nodes of the tree as categorical predictors. The NLLF for the empirical regression is then computed to represent the empirical risk of CART. The results are summarized in Table 4.

Table 4: Oracle properties of SPR in comparison with GLM-lasso and CART

Predictive model	Sample size = 2000			Sample size = 5000		
	SPR	GLM-lasso	CART	SPR	GLM-lasso	CART
Bayes' risk	0.72			0.73		
Empirical risk	0.78	3.78	4.38	0.74	3.76	4.25
Excess risk	0.06	3.06	3.66	0.01	3.03	3.52
Classification model	sample size = 2000			sample size = 5000		
	SPR	GLM-lasso	CART	SPR	GLM-lasso	CART
Bayes' risk	0.12			0.11		
Empirical risk	0.30	0.64	0.68	0.14	0.62	0.68
Excess risk	0.18	0.52	0.56	0.03	0.51	0.57

Also, to compare the computational efficiency of the different methods, we record the runtimes of model training by SPR, GLM-lasso, and CART, respectively, for each experiment performed in this section. On average, the runtimes for SPR, GLM-lasso, and CART are 8.82, 11.63, and 15.50 seconds, respectively.

Finally, we summarize our observations from the results: (i) SPR achieves high precision and recall in selecting the environmental variables (Table 1) and the input variables (Table 2). A smaller sample size slightly affects the precision but not the recall. (ii) The precision in selecting the input variables varies across the subdivisions (Table 2). Specifically, subdivisions 1 and 2 have the highest precision, while the other subdivisions have slightly lower precisions. This is because subdivisions 1 and 2 have the largest sample size. (iii) In comparison with the competing methods, SPR has significantly higher

precision in selecting the environmental variables and prediction accuracy than GLM-lasso and CART (Tables 1 and 3). SPR also significantly outperforms VC in prediction accuracy. GLM-lasso performs worse because it uses a single model to fit all the data which are known to be a mixture of 10 different distributions. CART performs worse, which is somewhat surprising because CART is known to be a flexible approach for modeling complex variable relationships with good performance. Its underperformance may be a result that it does not respect the (generalized) linear relationship between the input and output variables within each subdivision. VC performs worse because its smoothness assumption for the input-output relationship across adjacent subdivisions of the partition does not hold in our simulation settings. (iv) Table 4 shows that SPR has a substantially smaller empirical/excess risk than GLM-lasso and CART under a fixed sample size. When the sample size increases from 2000 to 5000, the excess risk for SPR shrinks dramatically while those for GLM-lasso and CART have little change. These provide evidence that SPR has a better oracle property. (v) In terms of computational efficiency, SPR on average only needs 76% and 57% of the runtimes by GLM-lasso and CART to complete model training, respectively.

Lastly in this section, we would like to discuss the pattern of the recursive partitioning process of SPR, as revealed by Figures 2 and 3. The pattern holds across all the simulation runs. Specifically, we observe that the regression fitted at the root node is the densest (i.e., having the most non-zero coefficients). As the recursive partitioning proceeds, the fitted regressions become sparser and sparser. Eventually at the leaf nodes, the sparsest regressions are obtained, which are consistent with the ground truth that each subdivision has only five out of 100 input variables with non-zero coefficients. The reason

for this trend is that the data used to fit a regression at an earlier stage of the recursive partitioning (i.e., closer to the root node) are more mixed with different distributions. Therefore, more input variables with non-zero coefficients are needed to fit the data. Even so, the fitting is still not good, which keeps the recursive partitioning going until reaching the leaf nodes. Another evidence of the less adequate fitting of the regression at earlier stages of the partitioning is that the magnitude of non-zero coefficients is generally smaller than that of the leaf nodes. All of these support the necessity and adequacy of the recursive partitioning in SPR.

2.5 Application

In this section, we present an application of using building design variables, outdoor environmental variables, and their interactions to predict building energy consumption. To obtain relevant data, we use *EnergyPlus*, a building energy simulation platform developed by the DOE (https://energyplus.net/). DOE developed *EnergyPlus* with a goal of making substantial progress toward improving energy efficiency for commercial and residential buildings in the U.S. Since its establishment, *EnergyPlus* has been used by numerous researchers, engineers, and architects to model energy consumption in various types of buildings. 16 building types can be simulated by *EnergyPlus*. In this study, we focus on "Big Offices", which is the most prevalent building type that *EnergyPlus* has been used for.

Table 5: Abbreviations and physical meanings of input, environmental, and output variables in building energy consumption modeling

	Variable abbreviation [unit]	Physical meaning
Input variables	people_occ	total number of people within the building zone
	air_temp [℃]	indoor air temperature
	air_re_hum [%]	indoor air relative humidity
	therm_set_temp [℃]	thermostat cooling setpoint temperature
	equip_sch	building equipment schedule; 0 and 1 for equipment off and on, respectively
	window_rad [W]	window total transmitted solar radiation rate
	window_sunlit_frac	fraction of window surface illuminated by unreflected beam solar radiation
	window_heat_gain [W]	surface window heat gain rate
	window_heat_loss [W]	surface window heat loss rate
	water_tank_temp [℃]	water heater tank temperature
	heat_coil [W]	average total heating capacity provide by heat pump
	cool_coil [W]	average total cooling load provided by heat pump
	people_air_temp [℃]	thermal comfort temperature that determines the balance between people heat gain and loss
Environmental variables	out_temp [℃]	outdoor air drybulb temperature
	out_re_hum [%]	outdoor air relative humidity
	out_airflow_frac	outdoor air flow fraction
	solar_diffuse [W/m^2]	diffuse solar radiation rate
	solar_direct [W/m^2]	direct solar radiation rate
	loc: {SFO, BLD, BWI, MIA, IAH, PHX}	airport codes of the six cities
Output variable	electricity [kw]	electricity consumption

Specifically, based on domain knowledge and existing literature (Eisenhower et al., 2012), we choose to include 13 input variables on the operational features of Big Offices that potentially affect building energy consumption. We further include six outdoor environmental variables, among which there is one categorical variable of building locations. Six locations across different climate zones of the U.S. are included: San Francisco (CA), Boulder (CO), Phoenix (AZ), Houston (TX), Miami (FL), and Baltimore (MD). The output variable is building energy consumption. Abbreviations and physical meanings of all the variables are given in Table 5. We run *EnergyPlus* and generate a dataset of one month (July) with a sampling frequency of every 30 minutes, which results in a total of 8922 samples.

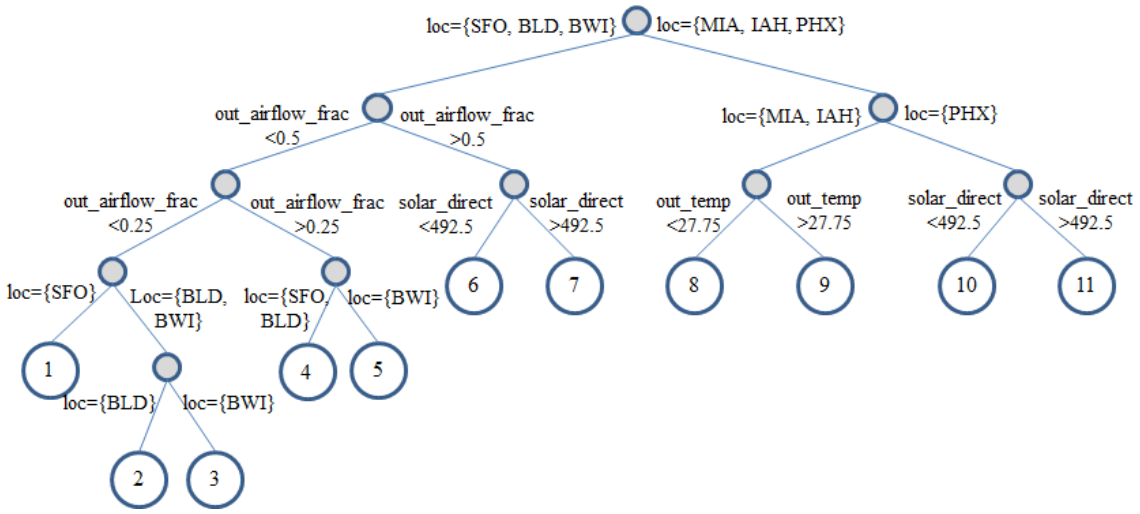

Figure 4: Partition of the space of environmental variables found by SPR

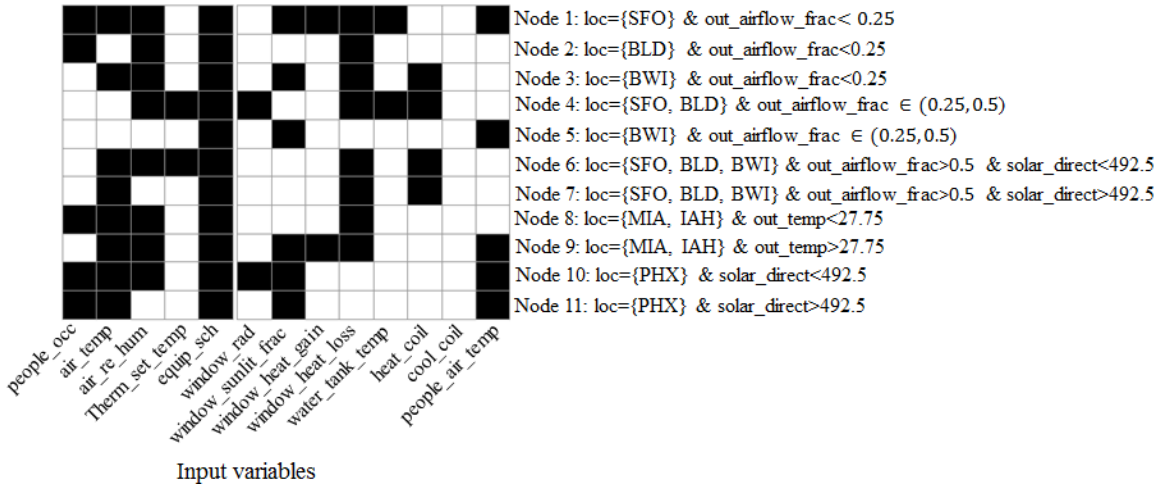

Figure 5: Zero (white) and non-zero (black) coefficients of the fitted l_1-regularized regression in each subdivision of the partition in Figure 4

Next, we apply SPR to the dataset. Since building energy consumption is a numerical variable, we apply the predictive model in our method. We choose to use the held-out estimator for the predictive model due to its empirically better performance than the penalized estimator. The entire datasets is split into a training set (first 11 days of data), a held-out validation set (next 10 days of data), and a test set (last 10 days of data). Figures 4 and 5 show the partition found by SPR and coefficients of the fitted l_1-regularized regression in each subdivision of the partition. Four out of the six environmental variables are used in the recursive partitioning, including loc, out_temp, out_airflow_frac, and solar_direct. Loc is used as the first variable to start the partitioning, which indicates that it helps the most on lowering the prediction error among all the environmental variables. The grouping of loc to the left and right branches makes sense as the right includes cities with high temperature or/and humidity that are known factors to affect building energy consumption significantly, while the left branch includes cities with different

characteristics. The right branch further splits into $loc = \{MIA, IAH\}$, two high-temperature high-humidity cities, and $loc = \{PHX\}$, a high-temperature low-humidity city. Moreover, within the same location $loc = \{PHX\}$, it is the solar_direct who affects the input-output relationship of buildings. Specifically, a close examination of the last two rows of Figure 4 shows that, compared with the regression model fitted at the high-level solar_direct (>492.5 W/m^2), two additional building operational variables (air_re_hum and window_rad) are selected to predict energy consumption at the lower-level of solar_direct (<492.5 W/m^2). This makes sense because when the outdoor direct solar radiation rate is low, building energy consumption is sensitive to how much of the radiation can be transmitted to indoor by windows (window_rad) and the indoor air humidity (air_re_hum). When the outdoor direct solar radiation rate is high, its effect on building energy consumption tends to be more dominant so as to make window_rad and air_re_hum less important.

Furthermore, we examine the left branch of the root node, which is further split by out_airflow_frac. Outdoor airflow rate affects indoor air circulation. The interaction effect of indoor air circulation and other building variables such as temperature and humidity on energy consumption is well-known. For example, for two buildings to achieve the same indoor temperature and humidity, the one with a lower level of air circulation typically needs to consume more electricity. After splitting by out_airflow_frac, the left branch is further split by solar_direct and loc, which are also variables used in the right branch.

In addition, we compare the regressions fitted for the 11 leaf nodes (subdivisions). The input variables selected by a majority of the regressions include equip_sch (selected by 11/11), air_temp (8/11), air_re_hum (8/11), window_heat_loss (8/11), and

window_sunlit_frac (6/11), which are well-known factors affecting building energy consumption. In particular, equip_sch is found to be the only globally significant input variable that affects building energy consumption. From the practical point of view, the existence of a globally significant input variable like equip_sch is an advantage because it means that equip_sch is a robust input variable against the environment. That is, by adjusting equip_sch, we have a chance to change the electricity consumption regardless of where the building is located. On the other hand, an input variable like heat_coil is not a globally significant input variable. By adjusting it, we can change the electricity consumption of some subdivisions such as 3, 4, 6, 7 but not others.

Moreover, there are no two regressions using the same set of input variables to predict the output. This finding is important for building energy management. Specifically, it suggests that how to adjust building design and operational variables (including what variables to adjust and how much to adjust) in order to reduce energy consumption should consider the environmental condition the building is operated under, especially that characterized by loc, out_airflow_frac, out_temp, and solar_direct. Just like "no treatment fits all" in personalized medicine, there is no energy management strategy that is universally applicable to all the buildings even of the same type (Large Offices in this application). On the other hand, not like personalized medicine in which the precision of treatment needs to be down to the level of individual patients, building energy management can be performed at a much coarser granularity. Not every building needs a different "treatment"; buildings within a certain range of the combinatorial environmental variables can be managed in the same way. By using SPR, ranges of this kind can be automatically

identified. Existences of these ranges are further supported by the superior prediction accuracy of SPR, which will be presented next.

Finally, for the purpose of comparison, we employ two competing methods, GLM-lasso and CART, on the same dataset. Since the output variable is numerical, the GLM-lasso is a lasso model and the CART is a regression tree, both of which use all input and environmental variables as predictors. We compute the MSPEs of the three methods on the test set, which are 5123.99, 11875.65, and 8802.35 for SPR, GLM-lasso, and CART, respectively. SPR has a significantly better prediction accuracy. Furthermore, Figure 6 plots the predicted versus true output variables of the test set by SPR, in which a clear linear trend is observed indicating a good prediction capability.

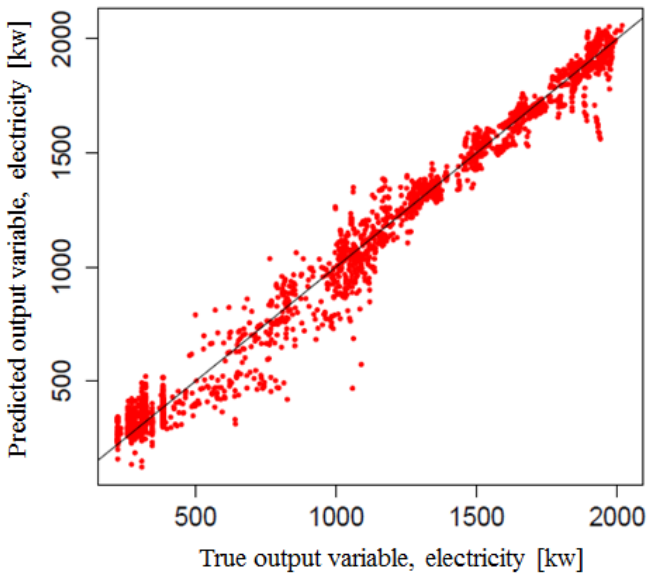

Figure 6: Predicted vs. true output variable (building electricity consumption) on the test set by SPR

2.6 Conclusion

In this paper, we developed a SPR for modeling the nonlinear interaction between a system and the multivariate environment it operates under. We proposed a penalized estimator and a held-out estimator for the SPR, analyzed theoretical properties of the estimators, and developed a recursive partitioning algorithm for model estimation. We conducted extensive simulation experiments to demonstrate the better performance of SPR compared with GLM-lasso and CART. An application of building energy prediction and management was finally presented. Extending from this research, there are abundant future directions. Immediate extension includes use of other sparsity-induced regularizations than the l_1-regularization to account for various structures of the input variables specified by domain knowledge, modeling of multiple correlated output variables, and fitting of nonlinear models between the input and output variables. Extensions that may need more substantial amounts of effort include design of ensemble methods similar to bagging, boosting, and random forest to reduce the variability of the recursive partitioning and development of optimization algorithms to search for the partition with a better optimality property. Both the SPR and its extensions have broad applicability to domains beyond building energy management, including but not limited to, mobile communication networks and wind energy as presented in Introduction, as well as bioinformatics in studying gene-environment interactions in related to diseases or disease traits.

CHAPTER 3

THEORETICAL STUDIES OF SPR

3.1 Oracle Inequalities of the SPR Estimators

We derive the oracle inequalities for the penalized estimator and held-out estimator of the classification and predictive models in Theorems 1-4, respectively. An oracle inequality is a bound on the risk of a statistical estimator, which shows that the performance of the estimator is almost (up to numerical constants) as good as of an ideal estimator that replies on perfect information supplied by an oracle and is not available in practice (Vapnik, 1998). An oracle inequality is an important property of a statistical estimator. For theoretical tractability, we focus on Dyadic Recursive Partitions (DRPs) of the space of the environmental variables. In DRPs, splitting a previously obtained subdivision can only happen at the midpoint of the range of an environmental variable. First, we define some notations. Let R^* be the minimum possible risk, i.e., the Bayes' risk, defined as

$$R^* = R(\boldsymbol{\mathcal{S}}^*, \boldsymbol{\theta}^*) = \inf_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}, \boldsymbol{\theta} \in \boldsymbol{\Omega}_{\boldsymbol{\theta}}} R(\boldsymbol{\mathcal{S}}, \boldsymbol{\theta}),$$

where $R(\boldsymbol{\mathcal{S}}, \boldsymbol{\theta}) = \sum_{r=1}^{n_S} E\left[L\left(Y, \hat{f}\left(\mathbf{x}; \boldsymbol{\theta}^{(r)}\right)\right) \cdot I\left(\mathbf{Z} \in \mathcal{S}^{(r)}\right)\right]$, $\boldsymbol{\mathcal{S}}_{DRP}$ is the set of all DRPs, and $\boldsymbol{\Omega}_{\boldsymbol{\theta}}$ is the domain of the model parameters $\boldsymbol{\theta}$. For a classification model, $\boldsymbol{\theta}$ consists of coefficients of multinomial logistic regressions, i.e., $\boldsymbol{\theta} = \boldsymbol{\beta}$. We assume that $\boldsymbol{\beta}$ is bounded, i.e., $\boldsymbol{\Omega}_{\boldsymbol{\beta}} = \left\{ \boldsymbol{\beta} \left| \max_{\substack{r=1,\dots,n_S \\ l=1,\dots,M}} \left| \mathbf{x}^T \boldsymbol{\beta}_l^{(r)} \right| \leq B \right. \right\}$. B is a positive constant. For a predictive model, $\boldsymbol{\theta}$ consists of coefficients and residual variances of linear regressions, i.e., $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\sigma}_{\varepsilon}^2\}$. We assume that $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}_{\varepsilon}^2$ are bounded, i.e., $\boldsymbol{\Omega}_{\boldsymbol{\alpha}, \boldsymbol{\sigma}_{\varepsilon}^2} = \left\{ \boldsymbol{\alpha}, \boldsymbol{\sigma}_{\varepsilon}^2 \left| \max_{r=1,\dots,n_S} \left| \mathbf{x}^T \boldsymbol{\alpha}^{(r)} \right| \leq \right. \right.$

$A, \max_{r=1,\ldots,n_S} \left|\log \tau^{(r)}\right| \leq L\Big\}$. $\tau^{(r)}$ is the reciprocal of $\sigma_{\varepsilon}^{2(r)}$. A and L be positive constants. Finally, let $[\![\boldsymbol{\mathcal{S}}]\!]$ denote the complexity of $\boldsymbol{\mathcal{S}}$. Specifically, $[\![\boldsymbol{\mathcal{S}}]\!]$ can be the length of a finite-length binary string used to encode $\boldsymbol{\mathcal{S}}$ in computers. Proofs of Theorems 3.1-3.4 can be found in the Appendices.

Theorem 3.1 (oracle inequality of the penalized estimator for the classification model): Let $\widehat{\boldsymbol{\mathcal{S}}}, \widehat{\boldsymbol{\beta}}$ be the penalized estimator. For a sufficiently large n and any $\delta \in (0,1)$, the excess risk of the penalized estimator with respect to the Bayes' risk satisfies the following inequality:

$$R\left(\widehat{\boldsymbol{\mathcal{S}}}, \widehat{\boldsymbol{\beta}}\right) - R^* \leq \inf_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} \left\{ \inf_{\boldsymbol{\beta} \in \boldsymbol{\Omega}_{\boldsymbol{\beta}}} \left(R(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta}) - R^*\right) + 2\, pen_c(\boldsymbol{\mathcal{S}}) \right\},$$

with probability at least $1 - \delta$, where $pen_c(\boldsymbol{\mathcal{S}}) = n_S\left(BM\sqrt{2v_1} + B + \log M\right)\sqrt{\frac{[\![\boldsymbol{\mathcal{S}}]\!] \log 2 + \log(2/\delta)}{n}}$.

Theorem 3.2 (oracle inequality of the held-out estimator for the classification model): Let $\widetilde{\boldsymbol{\mathcal{S}}}, \widetilde{\boldsymbol{\beta}}$ be the held-out estimator. For a sufficiently large n_1, n_2 and any $\delta \in (0,1)$, the excess risk of the held-out estimator with respect to the Bayes' risk satisfies the following inequality:

$$R\left(\widetilde{\boldsymbol{\mathcal{S}}}, \widetilde{\boldsymbol{\beta}}\right) - R^* \leq \inf_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} \left\{ \inf_{\boldsymbol{\beta} \in \boldsymbol{\Omega}_{\boldsymbol{\beta}}} \left(R(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta}) - R^*\right) + 2\phi_c^1(\boldsymbol{\mathcal{S}}) + \phi_c^2(\boldsymbol{\mathcal{S}}) \right\} + \phi_c^2\left(\widetilde{\boldsymbol{\mathcal{S}}}\right),$$

with probability at least $1 - \delta$, where $\phi_c^1(\boldsymbol{\mathcal{S}}) = n_S\left(BM\sqrt{2v_1} + B + \log M\right)\sqrt{\frac{[\![\boldsymbol{\mathcal{S}}]\!] \log 2 + \log(2/\delta)}{n_1}}$ and $\phi_c^2(\boldsymbol{\mathcal{S}}) = n_S\left(BM\sqrt{2v_1} + B + \log M\right)\sqrt{\frac{[\![\boldsymbol{\mathcal{S}}]\!] \log 2 + \log(2/\delta)}{n_2}}$.

Theorem 3.3 (oracle inequality of the penalized estimator for the predictive model): Let $\widehat{\boldsymbol{\mathcal{S}}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\sigma}}_{\varepsilon}^2$ be the penalized estimator. For a sufficiently large n and any $\delta \in (0,1)$, the excess

risk of the penalized estimator with respect to the Bayes' risk satisfies the following inequality:

$$R\left(\widehat{\boldsymbol{S}},\widehat{\boldsymbol{\alpha}},\widehat{\boldsymbol{\sigma}}_{\varepsilon}^{2}\right)-R^{*}\leq\inf_{\boldsymbol{S}\in\boldsymbol{S}_{DRP}}\left\{\inf_{\{\boldsymbol{\alpha},\boldsymbol{\sigma}_{\varepsilon}^{2}\}\in\boldsymbol{\Omega}_{\boldsymbol{\alpha},\boldsymbol{\sigma}_{\varepsilon}^{2}}}\left(R(\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\sigma}_{\varepsilon}^{2})-R^{*}\right)+2\,pen_{p}(\boldsymbol{S})\right\},$$

with probability at least $1-\delta$, where $pen_{p}(\boldsymbol{S})=n_{\boldsymbol{S}}(L+e^{L}C)\sqrt{\frac{[\![\boldsymbol{S}]\!]\log 2+\log(2/\delta)}{n}}$ and $C=\sqrt{2v_{2}}+2A\sqrt{2v_{1}}+A^{2}$.

Theorem 3.4 (oracle inequality of the held-out estimator for the predictive model): Let $\widetilde{\boldsymbol{S}},\widetilde{\boldsymbol{\alpha}},\widetilde{\boldsymbol{\sigma}}_{\varepsilon}^{2}$ be the held-out estimator. For a sufficiently large n_{1},n_{2} and any $\delta\in(0,1)$, the excess risk of the held-out estimator with respect to the Bayes' risk satisfies the following inequality:

$$R\left(\widetilde{\boldsymbol{S}},\widetilde{\boldsymbol{\alpha}},\widetilde{\boldsymbol{\sigma}}_{\varepsilon}^{2}\right)-R^{*}\leq\inf_{\boldsymbol{S}\in\boldsymbol{S}_{DRP}}\left\{\inf_{\{\boldsymbol{\alpha},\boldsymbol{\sigma}_{\varepsilon}^{2}\}\in\boldsymbol{\Omega}_{\boldsymbol{\alpha},\boldsymbol{\sigma}_{\varepsilon}^{2}}}\left(R(\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\sigma}_{\varepsilon}^{2})-R^{*}\right)+2\phi_{p}^{1}(\boldsymbol{S})+\phi_{p}^{2}(\boldsymbol{S})\right\}+\phi_{p}^{2}(\widetilde{\boldsymbol{S}}),$$

with probability at least $1-\delta$, where $\phi_{p}^{1}(\boldsymbol{S})=n_{\boldsymbol{S}}(L+e^{L}C)\sqrt{\frac{[\![\boldsymbol{S}]\!]\log 2+\log(2/\delta)}{n_{1}}}$ and $\phi_{p}^{2}(\boldsymbol{S})=n_{\boldsymbol{S}}(L+e^{L}C)\sqrt{\frac{[\![\boldsymbol{S}]\!]\log 2+\log(2/\delta)}{n_{2}}}$.

Appendix I Proof of Theorem 3.1

Recall that we use a multinomial logistic regression as the classification model. Then the risk and empirical risk are

$$R(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta}) = \frac{1}{n}\sum_{k=1}^{n}\sum_{r=1}^{n_S} E\left[\left(\log\sum_{l=1}^{M} exp\left\{\mathbf{x}_k^T\boldsymbol{\beta}_l^{(r)}\right\} - \sum_{l=1}^{M} U_l \cdot \mathbf{x}_k^T\boldsymbol{\beta}_l^{(r)}\right) \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right],$$ and

$$\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta}) = \frac{1}{n}\sum_{k=1}^{n}\sum_{r=1}^{n_S} \left(\log\sum_{l=1}^{M} exp\left\{\mathbf{x}_k^T\boldsymbol{\beta}_l^{(r)}\right\} - \sum_{l=1}^{M} u_{kl} \cdot \mathbf{x}_k^T\boldsymbol{\beta}_l^{(r)}\right) \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right),$$

respectively. $U_l = I(Y = l)$ and $u_{kl} = I(y_k = l)$ are indicator variables. By applying triangle inequality $|a + b| \le |a| + |b|$, we can get

$$\begin{aligned}\left|\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta}) - R(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta})\right| \le\ & \sum_{r=1}^{n_S}\left|\frac{1}{n}\sum_{k=1}^{n}\left(\log\sum_{l=1}^{M} exp\left\{\mathbf{x}_k^T\boldsymbol{\beta}_l^{(r)}\right\}\right) \cdot \left(I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right)\right| \\ & + \sum_{r=1}^{n_S}\left|\frac{1}{n}\sum_{k=1}^{n}\sum_{l=1}^{M}\mathbf{x}_k^T\boldsymbol{\beta}_l^{(r)} \cdot \left(u_{kl} \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[U_l \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right)\right|.\end{aligned} \quad \text{(A-1)}$$

Recall that in Section 3.1, we define B to be a positive constant that satisfies $\max_{\substack{r=1,\ldots,n_S \\ l=1,\ldots,M}}\left|\mathbf{x}^T\boldsymbol{\beta}_l^{(r)}\right| \le B$. Applying this inequality to (A-1), we can further derive that

$$\begin{aligned}\left|\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta}) - R(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta})\right| \le\ & \sum_{r=1}^{n_S}(B + \log M) \cdot \left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \\ & + \sum_{r=1}^{n_S} B \cdot \left|\frac{1}{n}\sum_{k=1}^{n} u_{kl} \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[U_l \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right|.\end{aligned} \quad \text{(A-2)}$$

Let A_1 and A_2 denote the two terms at the right hand side of (A-2), respectively. Next, we will derive the upper bounds for A_1 and A_2. The results of the deviation are given in (A-6) and (A-9). The details of the deviation are presented as follow:

First, we derive the upper bound for A_1. Using Hoeffding's inequality, we can get

$$P\left(\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \epsilon\right) \leq 2exp\{-2\epsilon^2\}, \tag{A-3}$$

for any $\epsilon > 0$. Furthermore, letting $\delta = 2exp\{-2\epsilon^2\} \in (0,1)$, we can write (A-3) as

$$P\left(\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{\frac{ln(2/\delta)}{2n}}\right) \leq \delta. \tag{A-4}$$

Recall that in Section 3.1, we define $[\![\boldsymbol{\mathcal{S}}]\!]$ to be the length of a finite-length binary string used to encode $\boldsymbol{\mathcal{S}}$ in computers, which reflects the complexity of $\boldsymbol{\mathcal{S}}$. Then, for $\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}$, $[\![\boldsymbol{\mathcal{S}}]\!]$ satisfies Kraft's inequality $\sum_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} 2^{-[\![\boldsymbol{\mathcal{S}}]\!]} \leq 1$. Letting $\delta_{\boldsymbol{\mathcal{S}}} = \delta \cdot 2^{-[\![\boldsymbol{\mathcal{S}}]\!]} \in (0,1)$ and using $\delta_{\boldsymbol{\mathcal{S}}}$ to replace the δ in (A-4), we can rewrite (A-4) as

$$P\left(\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{\frac{ln(2/\delta_{\boldsymbol{\mathcal{S}}})}{2n}}\right) \leq \delta_{\boldsymbol{\mathcal{S}}}.$$

Furthermore, by the union bound, we have

$$P\left(\exists \boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}: \left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{\frac{ln(2/\delta_{\boldsymbol{\mathcal{S}}})}{2n}}\right)$$

$$= P\left(\sup_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} \left\{\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| - \sqrt{\frac{ln(2/\delta_{\boldsymbol{\mathcal{S}}})}{2n}}\right\} \geq 0\right)$$

$$= P\left(\bigcup_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} \left\{\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| - \sqrt{\frac{ln(2/\delta_{\boldsymbol{\mathcal{S}}})}{2n}} \geq 0\right\}\right)$$

$$\leq \sum_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} P\left(\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{\frac{ln(2/\delta_{\boldsymbol{\mathcal{S}}})}{2n}}\right)$$

$$\leq \sum_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} \delta_{\boldsymbol{\mathcal{S}}}$$

$$= \sum_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} \delta \cdot 2^{-[\![\boldsymbol{\mathcal{S}}]\!]}$$

$$\leq \delta. \tag{A-5}$$

(A-5) indicates that, for any $\delta \in (0,1)$, we have

$$\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| < \sqrt{\frac{ln(2/\delta_{\boldsymbol{\mathcal{S}}})}{2n}} = \sqrt{\frac{[\![\boldsymbol{\mathcal{S}}]\!]ln2 + ln(2/\delta)}{2n}}$$

with probability at least $1-\delta$. That is, the upper bound for A_1 is

$$A_1 \le n_{\boldsymbol{S}}(B+\log M)\cdot\sqrt{\frac{[\![\boldsymbol{S}]\!]ln2+ln(2/\delta)}{2n}}. \tag{A-6}$$

Next, we derive the upper bound for A_2. We first define two constants v_1 and M_1that satisfy $\sup_{\mathcal{S}^{(r)}} E\left|\psi_1\left(\mathcal{S}^{(r)}\right)\right|^m \le \frac{m!M_1^{m-2}v_1}{2}$ for all $m \ge 2$, where $\psi_1\left(\mathcal{S}^{(r)}\right) = U\cdot I\left(\mathbf{Z}\in \mathcal{S}^{(r)}\right) - E\left[U\cdot I\left(\mathbf{Z}\in\mathcal{S}^{(r)}\right)\right]$. Using Bernstein's inequality, we can get

$$P\left(\left|\frac{1}{n}\sum_{k=1}^{n}u_{kl}\cdot I\left(\mathbf{z}_k\in\mathcal{S}^{(r)}\right) - E\left[U_l\cdot I\left(\boldsymbol{Z}\in\mathcal{S}^{(r)}\right)\right]\right|\ge\epsilon\right)\le 2exp\left\{-\frac{n\epsilon^2}{2(v_1+M_1\epsilon)}\right\} \tag{A-7}$$

for any $\epsilon>0$. v_1 and M_1 are defined in Section 3.1. Letting $\delta = 2exp\left\{-\frac{n\epsilon^2}{2(v_1+M_1\epsilon)}\right\}\in(0,1)$ and solving for ϵ, we can get $\epsilon = \frac{2M_1ln(2/\delta)+\sqrt{8nv_1ln(2/\delta)}}{2n}$. ϵ goes to $\sqrt{2v_1}\sqrt{\frac{ln(2/\delta)}{n}}$ as *n* goes to infinity. Therefore, (A-7) can be written into

$$P\left(\left|\frac{1}{n}\sum_{k=1}^{n}u_{kl}\cdot I\left(\mathbf{z}_k\in\mathcal{S}^{(r)}\right) - E\left[U_l\cdot I\left(\boldsymbol{Z}\in\mathcal{S}^{(r)}\right)\right]\right|\ge\sqrt{2v_1}\sqrt{\frac{ln(2/\delta)}{n}}\right)\le\delta.$$

Letting $\delta_{\boldsymbol{S}} = \delta\cdot 2^{-[\![\boldsymbol{S}]\!]}\in(0,1)$ and by the union bound, we have

$$P\left(\exists\boldsymbol{S}\in\boldsymbol{S}_{DRP}:\left|\frac{1}{n}\sum_{k=1}^{n}u_{kl}\cdot I\left(\mathbf{z}_k\in\mathcal{S}^{(r)}\right) - E\left[U_l\cdot I\left(\boldsymbol{Z}\in\mathcal{S}^{(r)}\right)\right]\right|\ge\sqrt{2v_1}\sqrt{\frac{ln(2/\delta_{\boldsymbol{S}})}{n}}\right)$$

$$= P\left(\sup_{\boldsymbol{S}\in\boldsymbol{S}_{DRP}}\left\{\left|\frac{1}{n}\sum_{k=1}^{n}u_{kl}\cdot I\left(\mathbf{z}_k\in\mathcal{S}^{(r)}\right) - E\left[U_l\cdot I\left(\boldsymbol{Z}\in\mathcal{S}^{(r)}\right)\right]\right| - \sqrt{2v_1}\sqrt{\frac{ln(2/\delta_{\boldsymbol{S}})}{n}}\right\}\ge 0\right)$$

$$= P\left(\cup_{\boldsymbol{S}\in \boldsymbol{S}_{DRP}}\left\{\left|\frac{1}{n}\sum_{k=1}^{n} u_{kl}\cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[U_l \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{2v_1}\sqrt{\frac{ln(2/\delta_{\boldsymbol{S}})}{n}}\right\}\right)$$

$$\leq \sum_{\boldsymbol{S}\in \boldsymbol{S}_{DRP}} P\left(\left|\frac{1}{n}\sum_{k=1}^{n} u_{kl}\cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[U_l \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{2v_1}\sqrt{\frac{ln(2/\delta_{\boldsymbol{S}})}{n}}\right)$$

$$\leq \sum_{\boldsymbol{S}\in \boldsymbol{S}_{DRP}} \delta_{\boldsymbol{S}}$$

$$= \sum_{\boldsymbol{S}\in \boldsymbol{S}_{DRP}} \delta \cdot 2^{-[\![\boldsymbol{S}]\!]} \leq \delta. \tag{A-8}$$

(A-8) indicates that, for any $\delta \in (0,1)$, we have

$$\left|\frac{1}{n}\sum_{k=1}^{n} u_{kl}\cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[U_l \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| < \sqrt{2v_1}\sqrt{\frac{[\![\boldsymbol{S}]\!]ln2+ln(2/\delta)}{n}}$$

with probability at least $1-\delta$. That is, the upper bound for A_2 is

$$A_2 \leq n_{\boldsymbol{S}} BM\sqrt{2v_1}\cdot\sqrt{\frac{[\![\boldsymbol{S}]\!]ln2+ln(2/\delta)}{n}}. \tag{A-9}$$

Now, inserting (A-6) and (A-9) into (A-2), we can get $\left|\hat{R}(\boldsymbol{S},\boldsymbol{\beta}) - R(\boldsymbol{S},\boldsymbol{\beta})\right| \leq n_{\boldsymbol{S}}\left(BM\sqrt{2v_1} + B + \log M\right)\sqrt{\frac{[\![\boldsymbol{S}]\!]ln2+ln(2/\delta)}{n}}$.

Since all the above deviation holds universally over the space of $\boldsymbol{S}_{DRP}$, we can get

$$\sup_{\boldsymbol{S},\boldsymbol{\beta}}\left|\hat{R}(\boldsymbol{S},\boldsymbol{\beta}) - R(\boldsymbol{S},\boldsymbol{\beta})\right| \leq n_{\boldsymbol{S}}\left(BM\sqrt{2v_1} + B + \log M\right)\sqrt{\frac{[\![\boldsymbol{S}]\!]ln2+ln(2/\delta)}{n}} \tag{A-10}$$

with probability at least $1-\delta$. Denote the right hand side of (A-10) by $upper(\boldsymbol{S})$. Furthermore, letting $\widehat{\boldsymbol{S}},\widehat{\boldsymbol{\beta}}$ be the penalized estimator and inserting them into (A-10), we can get

$$R\left(\widehat{\boldsymbol{S}},\widehat{\boldsymbol{\beta}}\right) \leq \hat{R}\left(\widehat{\boldsymbol{S}},\widehat{\boldsymbol{\beta}}\right) + upper\left(\widehat{\boldsymbol{S}}\right)$$

$$= \inf_{\boldsymbol{S},\boldsymbol{\beta}}\left\{\hat{R}(\boldsymbol{S},\boldsymbol{\beta}) + upper(\boldsymbol{S})\right\}.$$

Next, letting $pen_c(\boldsymbol{\mathcal{S}})$ take the form of $upper(\boldsymbol{\mathcal{S}})$, we can get

$$R\left(\widehat{\boldsymbol{\mathcal{S}}}, \widehat{\boldsymbol{\beta}}\right) \leq \inf_{\boldsymbol{\mathcal{S}}}\left\{\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta}^*) + pen_c(\boldsymbol{\mathcal{S}})\right\}$$

$$\leq \inf_{\boldsymbol{\mathcal{S}}}\{R(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta}^*) + 2pen_c(\boldsymbol{\mathcal{S}})\}$$

$$\leq \inf_{\boldsymbol{\mathcal{S}}}\left\{\inf_{\boldsymbol{\beta}} R(\boldsymbol{\mathcal{S}}, \boldsymbol{\beta}) + 2pen_c(\boldsymbol{\mathcal{S}})\right\}.$$

Finally, by subtracting R^* from both sides, the desired result in Theorem 3.1 can be obtained. ■

Appendix II Proof of Theorem 3.2

Recall that in defining the held-out estimator in Section 2.1, we divide the entire dataset into a training set $\mathcal{D}_1$ and a validation set $\mathcal{D}_2$ with n_1 and n_2 samples, respectively. Following a similar procedure used to derive (A-10) in the proof of Theorem 3.1, we can derive (A-11) and (A-12), i.e.,

$$\sup_{\boldsymbol{\mathcal{S}},\boldsymbol{\beta}} \left|\hat{R}_{tr}(\boldsymbol{\mathcal{S}},\boldsymbol{\beta}) - R(\boldsymbol{\mathcal{S}},\boldsymbol{\beta})\right| \le \phi_c^1(\boldsymbol{\mathcal{S}}), \text{ and} \tag{A-11}$$

$$\sup_{\boldsymbol{\mathcal{S}}} \left|\hat{R}_{val}(\boldsymbol{\mathcal{S}},\widetilde{\boldsymbol{\beta}}) - R(\boldsymbol{\mathcal{S}},\widetilde{\boldsymbol{\beta}})\right| \le \phi_c^2(\boldsymbol{\mathcal{S}}), \tag{A-12}$$

where $\phi_c^1(\boldsymbol{\mathcal{S}}) = n_S\left(BM\sqrt{2v_1} + B + \log M\right)\sqrt{\frac{[\![\boldsymbol{\mathcal{S}}]\!] ln2 + ln(2/\delta)}{n_1}}$, $\phi_c^2(\boldsymbol{\mathcal{S}}) = n_S\left(BM\sqrt{2v_1} + B + \log M\right)\sqrt{\frac{[\![\boldsymbol{\mathcal{S}}]\!] ln2 + ln(2/\delta)}{n_2}}$, and $\widetilde{\boldsymbol{\beta}}$ is an estimate for $\boldsymbol{\beta}$ from $\mathcal{D}_1$. Furthermore, plugging the held-out estimator, $\widetilde{\boldsymbol{\mathcal{S}}} = \operatorname{argmin}_{\boldsymbol{\mathcal{S}}} \hat{R}_{val}(\boldsymbol{\mathcal{S}},\widetilde{\boldsymbol{\beta}})$ into (A-12), we can get

$$\begin{aligned} R(\widetilde{\boldsymbol{\mathcal{S}}},\widetilde{\boldsymbol{\beta}}) &\le \hat{R}_{val}(\widetilde{\boldsymbol{\mathcal{S}}},\widetilde{\boldsymbol{\beta}}) + \phi_c^2(\widetilde{\boldsymbol{\mathcal{S}}}) \\ &= \inf_{\boldsymbol{\mathcal{S}}} \hat{R}_{val}(\boldsymbol{\mathcal{S}},\widetilde{\boldsymbol{\beta}}) + \phi_c^2(\widetilde{\boldsymbol{\mathcal{S}}}) \\ &\le \inf_{\boldsymbol{\mathcal{S}}} \left\{R(\boldsymbol{\mathcal{S}},\widetilde{\boldsymbol{\beta}}) + \phi_c^2(\boldsymbol{\mathcal{S}})\right\} + \phi_c^2(\widetilde{\boldsymbol{\mathcal{S}}}) \end{aligned}$$

Using (A-11), we then have

$$\begin{aligned} R(\widetilde{\boldsymbol{\mathcal{S}}},\widetilde{\boldsymbol{\beta}}) &\le \inf_{\boldsymbol{\mathcal{S}}} \left\{\hat{R}_{tr}(\boldsymbol{\mathcal{S}},\widetilde{\boldsymbol{\beta}}) + \phi_c^1(\boldsymbol{\mathcal{S}}) + \phi_c^2(\boldsymbol{\mathcal{S}})\right\} + \phi_c^2(\widetilde{\boldsymbol{\mathcal{S}}}) \\ &\le \inf_{\boldsymbol{\mathcal{S}}} \left\{R(\boldsymbol{\mathcal{S}},\widetilde{\boldsymbol{\beta}}) + 2\phi_c^1(\boldsymbol{\mathcal{S}}) + \phi_c^2(\boldsymbol{\mathcal{S}})\right\} + \phi_c^2(\widetilde{\boldsymbol{\mathcal{S}}}) \\ &= \inf_{\boldsymbol{\mathcal{S}}} \left\{\inf_{\boldsymbol{\beta}} \{R(\boldsymbol{\mathcal{S}},\boldsymbol{\beta})\} + 2\phi_c^1(\boldsymbol{\mathcal{S}}) + \phi_c^2(\boldsymbol{\mathcal{S}})\right\} + \phi_c^2(\widetilde{\boldsymbol{\mathcal{S}}}) \end{aligned}$$

By subtracting R^* from both sides, the desired result in Theorem 3.2 can be obtained. ■

Appendix III Proof of Theorem 3.3

Recall that we use a linear regression as the predictive model. Letting $\tau^{(r)}$ be the reciprocal of $\sigma_{\varepsilon}^{2(r)}$, then the risk and empirical risk can be written as

$R(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}, \boldsymbol{\tau}) = \frac{1}{n}\sum_{k=1}^{n}\sum_{r=1}^{n_S} E\left[\left(\tau^{(r)}\left(Y - \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\right)^2 - \log\tau^{(r)}\right)\cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]$, and

$\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}, \boldsymbol{\tau}) = \frac{1}{n}\sum_{k=1}^{n}\sum_{r=1}^{n_S}\left(\tau^{(r)}\left(y_k - \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\right)^2 - \log\tau^{(r)}\right)\cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right)$,

respectively. By applying triangle inequality $|a+b| \le |a| + |b|$, we can get

$$\left|\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}, \boldsymbol{\tau}) - R(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}, \boldsymbol{\tau})\right| \le \sum_{r=1}^{n_S}\left|\log\tau^{(r)}\cdot\left(\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right)\right| +$$

$$\sum_{r=1}^{n_S}\left|\tau^{(r)}\cdot\frac{1}{n}\sum_{k=1}^{n}\left(\left(y_k - \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\right)^2 I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[\left(Y - \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\right)^2\cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right)\right|. \quad \text{(A-13)}$$

Recall that in Section 3.1, we define L to be a positive constant that satisfies $\max_{r=1,\ldots,n_S}\left|\log\tau^{(r)}\right| \le L$. Applying this inequality to (A-13), we can further derive that

$$\left|\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}, \boldsymbol{\tau}) - R(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}, \boldsymbol{\tau})\right| \le \sum_{r=1}^{n_S} L\cdot\left|\left(\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right)\right| +$$

$$\sum_{r=1}^{n_S} e^{L}\cdot\left|\frac{1}{n}\sum_{k=1}^{n}\left(\left(y_k - \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\right)^2 I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[\left(Y - \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\right)^2\cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right)\right|. \quad \text{(A-14)}$$

Let B_1 and B_2 denote the two terms at the right hand side of (A-14), respectively. Next, we will derive the upper bound for B_1 and B_2. The details of the deviation are presented as follow:

First, we derive the upper bound for B_1. Using Hoeffding's inequality, we can get

$$P\left(\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \ge \epsilon\right) \le 2exp\{-2\epsilon^2\}, \quad \text{(A-15)}$$

for any $\epsilon > 0$. Furthermore, letting $\delta = 2exp\{-2\epsilon^2\} \in (0,1)$, we can write (A-15) as

$$P\left(\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{\frac{ln(2/\delta)}{2n}}\right) \leq \delta. \tag{A-16}$$

Recall that in Section 3.1, we define $[\![\boldsymbol{\mathcal{S}}]\!]$ to be the length of a finite-length binary string used to encode $\boldsymbol{\mathcal{S}}$ in computers, which reflects the complexity of $\boldsymbol{\mathcal{S}}$. Then, for $\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}$, $[\![\boldsymbol{\mathcal{S}}]\!]$ satisfies Kraft's inequality $\sum_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} 2^{-[\![\boldsymbol{\mathcal{S}}]\!]} \leq 1$. Letting $\delta_{\boldsymbol{\mathcal{S}}} = \delta \cdot 2^{-[\![\boldsymbol{\mathcal{S}}]\!]} \in (0,1)$ and using $\delta_{\boldsymbol{\mathcal{S}}}$ to replace the δ in (A-16), we can rewrite (A-16) as

$$P\left(\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{\frac{ln(2/\delta_{\boldsymbol{\mathcal{S}}})}{2n}}\right) \leq \delta_{\boldsymbol{\mathcal{S}}}.$$

Furthermore, by the union bound, we have

$$P\left(\exists \boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}: \left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{\frac{ln(2/\delta_{\boldsymbol{\mathcal{S}}})}{2n}}\right)$$

$$= P\left(\sup_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} \left\{\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| - \sqrt{\frac{ln(2/\delta_{\boldsymbol{\mathcal{S}}})}{2n}}\right\} \geq 0\right)$$

$$= P\left(\cup_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} \left\{\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| - \sqrt{\frac{ln(2/\delta_{\boldsymbol{\mathcal{S}}})}{2n}} \geq 0\right\}\right)$$

$$\leq \sum_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} P\left(\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{\frac{ln(2/\delta_{\boldsymbol{\mathcal{S}}})}{2n}}\right)$$

$$\leq \sum_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} \delta_{\boldsymbol{\mathcal{S}}}$$

$$= \sum_{\boldsymbol{\mathcal{S}} \in \boldsymbol{\mathcal{S}}_{DRP}} \delta \cdot 2^{-[\![\boldsymbol{\mathcal{S}}]\!]} \leq \delta. \tag{A-17}$$

(A-17) indicates that, for any $\delta \in (0,1)$, we have

$$\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| < \sqrt{\frac{[\![\boldsymbol{\mathcal{S}}]\!]ln2 + ln(2/\delta)}{2n}}$$

with probability at least $1 - \delta$. That is, the upper bound for B_1 is

$$B_1 \le n_{\mathcal{S}} L \cdot \sqrt{\frac{[\![\mathcal{S}]\!] ln2 + ln(2/\delta)}{2n}}. \tag{A-18}$$

Next, we derive the upper bound for B_2. We will decompose the second part of B_2 as follows:

$$\left|\frac{1}{n}\sum_{k=1}^{n}\left(\left(y_k - \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\right)^2 I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[\left(Y - \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\right)^2 \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right)\right|$$

$$\le \left|\frac{1}{n}\sum_{k=1}^{n} y_k^2 \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y^2 \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right|$$

$$+2\left|\frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)} \cdot \left(y_k \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right)\right|$$

$$+\left|\frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\left(\mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\right)^{\boldsymbol{T}} \cdot \left(I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right)\right|. \tag{A-19}$$

We now analyze each term in the summation on the right hand side of (A-19), respectively. We denote each of the items as below

$$B_{21} = \left|\frac{1}{n}\sum_{k=1}^{n} y_k^2 \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y^2 \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right|$$

$$B_{22} = 2\left|\frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)} \cdot \left(y_k \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right)\right|$$

$$B_{23} = \left|\frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\left(\mathbf{x}_k^T\boldsymbol{\alpha}^{(r)}\right)^{\boldsymbol{T}} \cdot \left(I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right)\right|$$

First, we derive an upper bound for B_{21}. We define two constants v_2 and M_2that satisfy $\sup_{\mathcal{S}^{(r)}} E\left|\psi_2\left(\mathcal{S}^{(r)}\right)\right|^m \le \frac{m! M_2^{m-2} v_2}{2}$ for all $m \ge 2$, where U is random variable and $\psi_2\left(\mathcal{S}^{(r)}\right) = U^2 \cdot I\left(\mathbf{Z} \in \mathcal{S}^{(r)}\right) - E\left[U^2 \cdot I\left(\mathbf{Z} \in \mathcal{S}^{(r)}\right)\right]$. Applying Bernstein's inequality on B_{21}, we can get

$$P\left(\left|\frac{1}{n}\sum_{k=1}^{n} y_k^2 \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y^2 \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \epsilon\right) \leq$$
$$2exp\left\{-\frac{n\epsilon^2}{2(\nu_2 + M_2\epsilon)}\right\} \tag{A-20}$$

for any $\epsilon > 0$. Letting $\delta = 2exp\left\{-\frac{n\epsilon^2}{2(\nu_2+M_2\epsilon)}\right\} \in (0,1)$ and solving for ϵ, we can get $\epsilon = \frac{2M_2 ln(2/\delta)+\sqrt{8n\nu_2 ln(2/\delta)}}{2n}$. ϵ goes to $\sqrt{2\nu_2}\sqrt{\frac{ln(2/\delta)}{n}}$ as n goes to infinity. Therefore, (A-20) can be written into

$$P\left(\left|\frac{1}{n}\sum_{k=1}^{n} y_k^2 \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y^2 \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{2\nu_2}\sqrt{\frac{ln(2/\delta)}{n}}\right) \leq \delta.$$

Letting $\delta_{\mathcal{S}} = \delta \cdot 2^{-[\![\mathcal{S}]\!]} \in (0,1)$ and by the union bound, we have

$$P\left(\exists \mathcal{S} \in \mathcal{S}_{DRP}: \left|\frac{1}{n}\sum_{k=1}^{n} y_k^2 \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y^2 \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{2\nu_2}\sqrt{\frac{ln(2/\delta_{\mathcal{S}})}{n}}\right)$$

$$= P\left(\sup_{\mathcal{S}\in\mathcal{S}_{DRP}}\left\{\left|\frac{1}{n}\sum_{k=1}^{n} y_k^2 \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y^2 \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| - \sqrt{2\nu_2}\sqrt{\frac{ln(2/\delta_{\mathcal{S}})}{n}}\right\} \geq 0\right)$$

$$= P\left(\cup_{\mathcal{S}\in\mathcal{S}_{DRP}}\left\{\left|\frac{1}{n}\sum_{k=1}^{n} y_k^2 \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y^2 \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{2\nu_2}\sqrt{\frac{ln(2/\delta_{\mathcal{S}})}{n}}\right\}\right)$$

$$\leq \sum_{\mathcal{S}\in\mathcal{S}_{DRP}} P\left(\left|\frac{1}{n}\sum_{k=1}^{n} y_k^2 \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y^2 \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{2\nu_2}\sqrt{\frac{ln(2/\delta_{\mathcal{S}})}{n}}\right)$$

$$\leq \sum_{\mathcal{S}\in\mathcal{S}_{DRP}} \delta_{\mathcal{S}}$$

$$= \sum_{\mathcal{S}\in\mathcal{S}_{DRP}} \delta \cdot 2^{-[\![\mathcal{S}]\!]} \leq \delta.$$

Therefore, for any $\delta \in (0,1)$, we have an upper bound for B_{21} in the form of

$$B_{21} < \sqrt{2\nu_2}\sqrt{\frac{[\![\mathcal{S}]\!]ln2+ln(2/\delta)}{n}} \tag{A-21}$$

with probability at least $1-\delta$.

Then, we derive an upper bound for B_{22}. Recall that we define A to be a positive constant that satisfies $\max_{r=1,\dots,n_S} \left|\mathbf{x}^T \boldsymbol{\alpha}^{(r)}\right| \leq A$. Applying this to B_{22}, then we can write

$$B_{22} \leq 2A \left| \frac{1}{n} \sum_{k=1}^{n} y_k \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right] \right|.$$

We define two constants ν_1 and M_1that satisfy $\sup_{\mathcal{S}^{(r)}} E\left|\psi_1\left(\mathcal{S}^{(r)}\right)\right|^m \leq \frac{m! M_1^{m-2} \nu_1}{2}$ for all $m \geq 2$, where $\psi_1\left(\mathcal{S}^{(r)}\right) = U \cdot I\left(\mathbf{Z} \in \mathcal{S}^{(r)}\right) - E\left[U \cdot I\left(\mathbf{Z} \in \mathcal{S}^{(r)}\right)\right]$. Using Bernstein's inequality, we can get

$$P\left(\left|\frac{1}{n} \sum_{k=1}^{n} y_k \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \epsilon\right) \leq 2exp\left\{-\frac{n\epsilon^2}{2(\nu_1 + M_1\epsilon)}\right\} \tag{A-22}$$

for any $\epsilon > 0$. ν_1 and M_1 are defined in Section 4. Letting $\delta = 2exp\left\{-\frac{n\epsilon^2}{2(\nu_1 + M_1\epsilon)}\right\} \in (0,1)$ and solving for ϵ, we can get $\epsilon = \frac{2M_1 ln(2/\delta) + \sqrt{8n\nu_1 ln(2/\delta)}}{2n}$. ϵ goes to $\sqrt{2\nu_1}\sqrt{\frac{ln(2/\delta)}{n}}$ as n goes to infinity. Therefore, (A-22) can be written into

$$P\left(\left|\frac{1}{n} \sum_{k=1}^{n} y_k \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{2\nu_1}\sqrt{\frac{ln(2/\delta)}{n}}\right) \leq \delta.$$

Letting $\delta_{\boldsymbol{S}} = \delta \cdot 2^{-[\![\boldsymbol{S}]\!]} \in (0,1)$ and by the union bound, we have

$$P\left(\exists \boldsymbol{S} \in \boldsymbol{S}_{DRP}: \left|\frac{1}{n} \sum_{k=1}^{n} y_k \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{2\nu_1}\sqrt{\frac{ln(2/\delta_{\boldsymbol{S}})}{n}}\right)$$

$$= P\left(\bigcup_{\boldsymbol{S} \in \boldsymbol{S}_{DRP}} \left\{\left|\frac{1}{n} \sum_{k=1}^{n} y_k \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{2\nu_1}\sqrt{\frac{ln(2/\delta_{\boldsymbol{S}})}{n}}\right\}\right)$$

$$\leq \sum_{\boldsymbol{S} \in \boldsymbol{S}_{DRP}} P\left(\left|\frac{1}{n} \sum_{k=1}^{n} y_k \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| \geq \sqrt{2\nu_1}\sqrt{\frac{ln(2/\delta_{\boldsymbol{S}})}{n}}\right)$$

$$\leq \sum_{\boldsymbol{S} \in \boldsymbol{S}_{DRP}} \delta_{\boldsymbol{S}}$$

$$= \sum_{\boldsymbol{S}\in\boldsymbol{S}_{DRP}} \delta \cdot 2^{-[\![\boldsymbol{S}]\!]} \leq \delta. \tag{A-23}$$

(A-23) indicates that, for any $\delta \in (0,1)$, we have

$$\left|\frac{1}{n}\sum_{k=1}^{n} y_k \cdot I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[Y \cdot I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right| < \sqrt{2v_1}\sqrt{\frac{[\![\boldsymbol{S}]\!]ln2+ln(2/\delta)}{n}}$$

with probability at least $1-\delta$. That is, the upper bound for B_{22} is

$$B_{22} \leq 2A\sqrt{2v_1} \cdot \sqrt{\frac{[\![\boldsymbol{S}]\!]ln2+ln(2/\delta)}{n}} \tag{A-24}$$

with probability at least $1-\delta$.

Finally, applying the assumption of $\max_{r=1,\ldots,n_S}\left|\mathbf{x}^T\boldsymbol{\alpha}^{(r)}\right| \leq A$ to B_{23}, then we can write

$$B_{23} \leq A^2\left|\frac{1}{n}\sum_{k=1}^{n} I\left(\mathbf{z}_k \in \mathcal{S}^{(r)}\right) - E\left[I\left(\boldsymbol{Z} \in \mathcal{S}^{(r)}\right)\right]\right|.$$

An upper bound for B_{23} is obtained following (A-17). For any $\delta \in (0,1)$,

$$B_{23} \leq A^2 \cdot \sqrt{\frac{[\![\boldsymbol{S}]\!]ln2+ln(2/\delta)}{2n}} \tag{A-25}$$

Now, inserting (A-21), (A-24), and (A-25) into (A-19), we can derive an upper bound for B_2 as follow

$$B_2 \leq n_{\boldsymbol{S}}e^L C \cdot \sqrt{\frac{[\![\boldsymbol{S}]\!]ln2+ln(2/\delta)}{n}}, \tag{A-26}$$

where $C = \sqrt{2v_2} + 2A\sqrt{2v_1} + A^2$. Combining (A-18) and (A-26) into (A-14), we can get

$$\left|\hat{R}(\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\tau}) - R(\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\tau})\right| \leq n_{\boldsymbol{S}}(L+e^L C) \cdot \sqrt{\frac{[\![\boldsymbol{S}]\!]ln2+ln(2/\delta)}{n}}.$$

Since all the above deviation holds universally over the space of $\boldsymbol{S}_{DRP}$, we can get

$$\sup_{\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\tau}}\left|\hat{R}(\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\tau}) - R(\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\tau})\right| \leq n_{\boldsymbol{S}}(L+e^L C) \cdot \sqrt{\frac{[\![\boldsymbol{S}]\!]ln2+ln(2/\delta)}{n}}, \tag{A-27}$$

with probability at least $1-\delta$. Denote the right hand side of (A-27) by $upper(\boldsymbol{\mathcal{S}})$. Furthermore, letting $\widehat{\boldsymbol{\mathcal{S}}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\tau}}$ be the penalized estimator and inserting them into (A-27), we can get

$$\begin{aligned} R\left(\widehat{\boldsymbol{\mathcal{S}}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\tau}}\right) &\leq \hat{R}\left(\widehat{\boldsymbol{\mathcal{S}}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\tau}}\right) + upper\left(\widehat{\boldsymbol{\mathcal{S}}}\right) \\ &= \inf_{\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}, \boldsymbol{\tau}} \{R(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}, \boldsymbol{\tau}) + upper(\boldsymbol{\mathcal{S}})\}. \end{aligned}$$

Next, letting $pen_P(\boldsymbol{\mathcal{S}})$ take the form of $upper(\boldsymbol{\mathcal{S}})$, we can get

$$\begin{aligned} R\left(\widehat{\boldsymbol{\mathcal{S}}}, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\tau}}\right) &\leq \inf_{\boldsymbol{\mathcal{S}}}\left\{\hat{R}(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}^*, \boldsymbol{\tau}^*) + pen_P(\boldsymbol{\mathcal{S}})\right\} \\ &\leq \inf_{\boldsymbol{\mathcal{S}}}\{R(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}^*, \boldsymbol{\tau}^*) + 2pen_P(\boldsymbol{\mathcal{S}})\} \\ &\leq \inf_{\boldsymbol{\mathcal{S}}}\left\{\inf_{\boldsymbol{\alpha}, \boldsymbol{\tau}} R(\boldsymbol{\mathcal{S}}, \boldsymbol{\alpha}, \boldsymbol{\tau}) + 2pen_P(\boldsymbol{\mathcal{S}})\right\}. \end{aligned}$$

Finally, by subtracting R^* from both sides, the desired result in Theorem 3.3 can be obtained. ■

Appendix IV Proof of Theorem 3.4

Recall that in defining the held-out estimator in Section 2.1, we divide the entire dataset into a training set $\mathcal{D}_1$ and a validation set $\mathcal{D}_2$ with $n_1$and n_2 samples, respectively. Following a similar procedure used to derive (A-27) in the proof of Theorem 3.3, we can derive (A-28) and (A-29), i.e.,

$$\sup_{\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\tau}} \left|\hat{R}_{tr}(\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\tau}) - R(\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\tau})\right| \le \phi_p^1(\boldsymbol{S}), \text{ and} \tag{A-28}$$

$$\sup_{\boldsymbol{S}} \left|\hat{R}_{val}(\boldsymbol{S},\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\tau}}) - R(\boldsymbol{S},\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\tau}})\right| \le \phi_p^2(\boldsymbol{S}), \text{ and} \tag{A-29}$$

where $\phi_p^1(\boldsymbol{S}) = n_{\boldsymbol{S}}(L + e^L C) \cdot \sqrt{\frac{[\![\boldsymbol{S}]\!] ln2 + ln(2/\delta)}{n_1}}$, $\phi_p^2(\boldsymbol{S}) = n_{\boldsymbol{S}}(L + e^L C) \cdot \sqrt{\frac{[\![\boldsymbol{S}]\!] ln2 + ln(2/\delta)}{n_2}}$, and $\tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\tau}}$ are estimates for $\boldsymbol{\alpha}, \boldsymbol{\tau}$ from $\mathcal{D}_1$. Furthermore, plugging the held-out estimator, $\widetilde{\boldsymbol{S}} = \underset{\boldsymbol{S}}{\operatorname{argmin}}\, \hat{R}_{val}(\boldsymbol{S},\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\tau}})$ into (A-29), we can get

$$\begin{aligned} R(\widetilde{\boldsymbol{S}},\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\tau}}) &\le \hat{R}_{val}(\widetilde{\boldsymbol{S}},\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\tau}}) + \phi_p^2(\widetilde{\boldsymbol{S}}) \\ &= \inf_{\boldsymbol{S}} \hat{R}_{val}(\boldsymbol{S},\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\tau}}) + \phi_p^2(\widetilde{\boldsymbol{S}}) \\ &\le \inf_{\boldsymbol{S}} \left\{R(\boldsymbol{S},\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\tau}}) + \phi_p^2(\boldsymbol{S})\right\} + \phi_p^2(\widetilde{\boldsymbol{S}}) \end{aligned}$$

Using (A-28), we then have

$$\begin{aligned} R(\widetilde{\boldsymbol{S}},\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\tau}}) &\le \inf_{\boldsymbol{S}} \left\{\hat{R}_{tr}(\boldsymbol{S},\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\tau}}) + \phi_p^1(\boldsymbol{S}) + \phi_p^2(\boldsymbol{S})\right\} + \phi_p^2(\widetilde{\boldsymbol{S}}) \\ &\le \inf_{\boldsymbol{S}} \left\{R(\boldsymbol{S},\tilde{\boldsymbol{\alpha}},\tilde{\boldsymbol{\tau}}) + 2\phi_p^1(\boldsymbol{S}) + \phi_p^2(\boldsymbol{S})\right\} + \phi_p^2(\widetilde{\boldsymbol{S}}) \\ &\le \inf_{\boldsymbol{S}} \left\{\inf_{\boldsymbol{\alpha},\boldsymbol{\tau}} \{R(\boldsymbol{S},\boldsymbol{\alpha},\boldsymbol{\tau})\} + 2\phi_p^1(\boldsymbol{S}) + \phi_p^2(\boldsymbol{S})\right\} + \phi_p^2(\widetilde{\boldsymbol{S}}) \end{aligned}$$

By subtracting R^* from both sides, the desired result in Theorem 3.4 can be obtained. ■

CHAPTER 4

TREE-BASED STRUCTURE-REGULARIZED REGRESSION MODEL

4.1 Introduction

TBSR models the relationship between indoor variables and response variable in each leaf node of a tree grown by partitioning on outdoor variables. A novel joint feature selection method is proposed by applying a hierarchical structured regularization schema on the shared features of the models within the leaf nodes. An estimation procedure and a tree growing algorithm are also developed. Simulation studies are conducted to demonstrate the better performance of TBSR compared with other competing methods. Finally, TBSR is applied to a real dataset that is collected from a solar-powered house to provide a forecasting module that could be used for a more efficient temperature control to reduce HVAC system energy consumption.

The rest of this chapter is organized as follow: Section 4.2 introduces the mathematic formulation, estimation, and tree growing algorithm of TBSR model. Section 4.3 presents the simulation studies of TBSR model and the comparisons with other competing methods. Section 4.4 involves an application of TBSR model in indoor temperature forecasting. Section 4.5 concludes this chapter.

4.2 Formulation

The proposed method includes two parts: tree growing and regression fitting at each step of the tree growing process. In this section, we first assume a given tree and discuss the regression modeling at each leaf node of the tree (Section 4.2.1). Specifically, we propose a hierarchical multitask learning (HierML) model that jointly build regressions at each leaf node with consideration of the hierarchical structure of the leaf nodes. Next, we present the algorithm for estimating the model parameters of HierML (Section 4.2.2). Noting that Sections 4.2.1-4.2.2 actually provide the formulation and algorithm of regression fitting at each step of the tree growing process, we finally present the algorithm for growing the tree in Section 4.2.3.

4.2.1 Hierarchical Multitask Learning (HierML) Model: Mathematical Formulation

Denote the tree obtained at the s-th step of the tree growing process by T_s. Denote all the nodes of T_s by $\mathbf{V}_s$. $\mathbf{V}_s$ includes both internal and leaf nodes. Leaf nodes are those without children in the tree. Denote the leaf nodes by $\mathbf{V}_{l,s} \subset \mathbf{V}_s$. Let $\mathbf{X}$, $\mathbf{Z}$, y be the set of indoor variables, the set of outdoor variables, and the response variable, respectively. The splitting variables in T_s are outdoor variables from $\mathbf{Z}$. Each leaf node v_l corresponds to a subdivision of the space defined by $\mathbf{Z}$, on which a regression needs to be built to link the response with indoor variables, i.e., $\mathbf{y}_{v_l} = \mathbf{X}_{v_l}\boldsymbol{\beta}_{v_l} + \boldsymbol{\varepsilon}_{v_l}$.

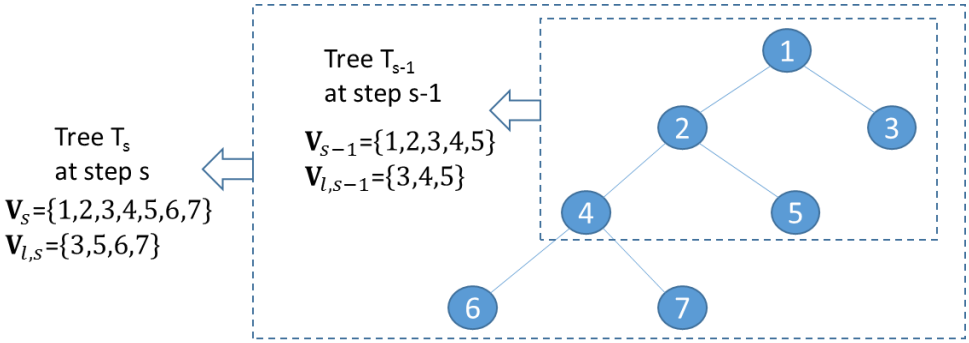

Figure 7:An example of the tree growing process and notations (only two successive steps of the process are showing for simplicity of presentation)

In the existing methods including our previously developed SPR, the regression fitting at the s-th step will fit a regression model for each of the two child nodes separately (e.g., nodes 6 and 7 in Figure 7). In this paper, we propose to fit regression models for all the leaf nodes of T_s jointly. This means, for example, fitting regression models for nodes 6, 7, 3, and 5 jointly. The basic idea is to leverage the hierarchical grouping structure encoded by the tree to regularize the extent to which the regression models should be similar to each other. Two regression models are more similar if their corresponding leaf nodes are grouped at a lower level of the hierarchy. For example, the regression models for nodes 6 and 7 in Figure 7 should be more similar to each other than those for nodes 6 and 5; the models for 6 and 5 should be more similar than those for 6 and 3. A clear advantage of this joint fitting is that it exploits all the data, while separate fitting of each model is likely to suffer from small sample sizes especially when the tree grows deeper.

Next, we present the details of the proposed model. For notation simplicity, we will drop the subscript "s" in the following discussion. Our objective is to fit a regression model for each leaf node, $v_l \in \mathbf{V}_l$. This means, for example, to fit a regression for nodes 6, 7, 3, and 5, respectively, for tree T_s in Figure 7. Let $\mathbf{y}_{v_l}$ and $\mathbf{X}_{v_l}$ be the data of the response and

input variables for leaf node v_l, respectively. The regression model is $\mathbf{y}_{v_l} = \mathbf{X}_{v_l}\boldsymbol{\beta}_{v_l} + \boldsymbol{\varepsilon}_{v_l}$. To estimate the regression coefficients $\boldsymbol{\beta}_{v_l}$, we propose the following penalized formulation:

$$\left\{\widehat{\boldsymbol{\beta}}_{v_l}\right\}_{v_l \in \mathbf{V}_l} = \ argmin \ \ \sum_{v_l}\left(\mathbf{y}_{v_l} - \mathbf{X}_{v_l}\boldsymbol{\beta}_{v_l}\right)^T\left(\mathbf{y}_{v_l} - \mathbf{X}_{v_l}\boldsymbol{\beta}_{v_l}\right) + \lambda \sum_j \sum_v w_v \left\|\boldsymbol{\beta}^j_{\mathbf{G}_v}\right\|_2. \tag{4.1}$$

The first term in (4.1) is the least-square error loss. The second term is a specially-designed penalty that deserves more discussion. Specifically, for each node of the tree, i.e., $v \in \mathbf{V}$, $\mathbf{G}_v$ denotes the set of leaf nodes growing from v. Taking tree T_s Figure 7 as an example, $v = 1,2,\ldots,7$. $\mathbf{G}_1 = \{3,5,6,7\}$, $\mathbf{G}_2 = \{5,6,7\}$, $\mathbf{G}_3 = \{3\}$, $\mathbf{G}_4 = \{6,7\}$, $\mathbf{G}_5 = \{5\}$, $\mathbf{G}_6 = \{6\}$, and $\mathbf{G}_7 = \{7\}$. Let $\boldsymbol{\beta}^j_{\mathbf{G}_v}$ contain the set of regression coefficients corresponding to the j-th predictor in $\mathbf{G}_v$, e.g., $\boldsymbol{\beta}^j_{\mathbf{G}_1} = \{\beta^j_3, \beta^j_5, \beta^j_6, \beta^j_7\}$, $\boldsymbol{\beta}^j_{\mathbf{G}_2} = \{\beta^j_5, \beta^j_6, \beta^j_7\}$, $\boldsymbol{\beta}^j_{\mathbf{G}_3} = \{\beta^j_3\}$, and so on. Following a similar idea to multitask learning using l_{21} regularization (Obozinski et al., 2009), we treat the $\mathbf{G}_v$'s as different tasks and put a weighted l_2-norm on $\boldsymbol{\beta}^j_{\mathbf{G}_v}$, i.e., $w_v\left\|\boldsymbol{\beta}^j_{\mathbf{G}_v}\right\|_2$. w_v is a weight that will discussed later. Then, we put an l_1-norm outside the weighted l_2-norm, i.e., $\sum_v w_v\left\|\boldsymbol{\beta}^j_{\mathbf{G}_v}\right\|_2$, to enable selection of regression coefficients contained in each $\boldsymbol{\beta}^j_{\mathbf{G}_v}$ as a group, and another l_1-norm further outside to enable selection of regression coefficients corresponding to the j-th predictor as a group, i.e., $\sum_j \sum_v w_v\left\|\boldsymbol{\beta}^j_{\mathbf{G}_v}\right\|_2$. λ in (4.1) is a tuning parameter to balance the least-square loss and the proposed penalty.

Furthermore, we discuss how to design the weight w_v associated with each node v. There are two types of nodes in tree T_s: internal nodes and leaf nodes. An internal node has two children, each of which can be a leaf node or a subtree by itself. Start from the lowest internal node in T_s, i.e., $v = 4$, whose children are leaf nodes 6 and 7. We consider that the regression coefficients for the j-th predictor in nodes 6 and 7, β_6^j and β_7^j, should be similar because they share the same internal node and meanwhile should not be exactly the same (otherwise they would not have been split into two different nodes). To account for these two aspects simultaneously, we propose the following penalty on β_6^j and β_7^j:

$$W_j(4) \triangleq g_4\sqrt{\left(\beta_6^j\right)^2 + \left(\beta_7^j\right)^2} + s_4\left(\left|\beta_6^j\right| + \left|\beta_7^j\right|\right), \tag{4.2}$$

where $\sqrt{\left(\beta_6^j\right)^2 + \left(\beta_7^j\right)^2}$ encourages the two coefficients to be selected jointly while $\left|\beta_6^j\right|$ and $|\beta_7^j|$ encourage selection separately, and g_4 and s_4 are the corresponding weights. Using the definition of $\boldsymbol{\beta}_{\mathbf{G}_v}^j$, (4.2) can be written as

$$W_j(4) = g_4\left\|\boldsymbol{\beta}_{\mathbf{G}_4}^j\right\|_2 + s_4\left(\left|\beta_6^j\right| + \left|\beta_7^j\right|\right). \tag{4.3}$$

Furthermore, we can move up to the next internal node, i.e., $v = 2$, whose children are a subtree starting from node 4 on the left and leaf node 5 on the right. Following a similar idea to (4.3), we can write down the penalty associated with $v = 2$ as

$$W_j(2) = g_2\left\|\boldsymbol{\beta}_{\mathbf{G}_2}^j\right\|_2 + s_2\left(W_j(4) + \left|\beta_5^j\right|\right),$$

where $\left\|\boldsymbol{\beta}_{\mathbf{G}_2}^j\right\|_2$ encourages the regression coefficients in the leaf nodes growing from node 2 to be selected jointly, while $W_j(4)$ and $|\beta_5^j|$ encourage the coefficients in the left and

right children of node 2 to be selected separately. In a similar way, the penalty associated with the internal node $v = 1$ is

$$W_j(1) = g_1\left\|\boldsymbol{\beta}_{\mathbf{G}_1}^j\right\|_2 + s_1\left(W_j(2) + |\beta_3^j|\right).$$

To generalize the above scheme, we can write the definition of $W_j(v)$ as

$$W_j(v) = \begin{Bmatrix} g_v\left\|\boldsymbol{\beta}_{\mathbf{G}_v}^j\right\|_2 + s_v \sum_{c\in children(v)} W_j(c) & if\ v\ is\ an\ internal\ node \\ \left|\beta_v^j\right| & if\ v\ is\ leaf\ node \end{Bmatrix},$$

with $g_v + s_v = 1$ for identifiability consideration. Furthermore, we can write the penalty term in (4.1) as

$$\sum_j \sum_v w_v\left\|\boldsymbol{\beta}_{\mathbf{G}_v}^j\right\|_2 = \sum_j W_j(v_{root}), \tag{4.4}$$

where v_{root} is the root node of the tree. Through some algebra, w_v can be shown to related to g_v and s_v in the following way:

$$w_v = \begin{Bmatrix} g_v \prod_{m\in Ancestors(v)} s_m & if\ v\ is\ an\ internal\ node \\ \prod_{m\in Ancestors(v)} s_m & if\ v\ is\ leaf\ node \end{Bmatrix}. \tag{4.5}$$

Using T_s in Figure 7 as an example, it can be shown that the right-hand side of (4.4) becomes:

$$\begin{aligned}
\sum_j W_j(1) &= g_1\left\|\boldsymbol{\beta}_{\mathbf{G}_1}^j\right\|_2 + s_1\left(W_j(2) + \left|\beta_3^j\right|\right) \\
&= g_1\left\|\boldsymbol{\beta}_{\mathbf{G}_1}^j\right\|_2 + s_1\left[g_2\left\|\boldsymbol{\beta}_{\mathbf{G}_2}^j\right\|_2 + s_2\left(W_j(4) + |\beta_5^j|\right) + |\beta_3^j|\right] \\
&= g_1\left\|\boldsymbol{\beta}_{\mathbf{G}_1}^j\right\|_2 \\
&\quad + s_1\left\{g_2\left\|\boldsymbol{\beta}_{\mathbf{G}_2}^j\right\|_2 + s_2\left[g_4\left\|\boldsymbol{\beta}_{\mathbf{G}_4}^j\right\|_2 + s_4\left(\left|\beta_6^j\right| + |\beta_7^j|\right) + |\beta_5^j|\right] + |\beta_3^j|\right\} \\
&= g_1\left\|\boldsymbol{\beta}_{\mathbf{G}_1}^j\right\|_2 + g_2 s_1\left\|\boldsymbol{\beta}_{\mathbf{G}_2}^j\right\|_2 + g_4 s_1 s_2\left\|\boldsymbol{\beta}_{\mathbf{G}_4}^j\right\|_2 + s_1 s_2 s_4\left|\beta_6^j\right| + s_1 s_2 s_4\left|\beta_7^j\right| \\
&\quad + s_1 s_2\left|\beta_5^j\right| + s_1\left|\beta_3^j\right|.
\end{aligned}$$

Comparing above equation with the left-hand side of (4.4), we can get $w_1 = g_1, w_2 = g_2 s_1$, $w_4 = s_1 s_2 g_4, w_6 = s_1 s_2 s_4, w_7 = s_1 s_2 s_4, w_5 = s_1 s_2, w_3 = s_1$. It is easy to verify that these results comply with the formula in (4.5).

To recap how we design the penalty term $\sum_j \sum_v w_v \left\| \boldsymbol{\beta}_{\mathbf{G}_v}^j \right\|_2$ to integrate group effect and individual effect on the features across different tasks, we visualize the hierarchical structured regularization in Figure 8. For j-th feature, l_2-norms are put to each group of coefficients $\boldsymbol{\beta}_{\mathbf{G}_v}^j$. By assigning weights to each of these l_2-norms, an l_1-norm is added on the top and is designed for group selection across all the $\boldsymbol{\beta}_{\mathbf{G}_v}^j$'s for feature j. Similarly, for other features, the same weighted l_{21} norms are added. Finally, another l_1-norm is added across all the features. This penalty term not only adopts the idea of group lasso that selects features jointly, but also involves the idea of elastic net that controls the strength of penalty on group effect and individual effect by adjusting the weights. To this end, the features can be flexibly selected by their similarities across different tasks rather than selecting in all.

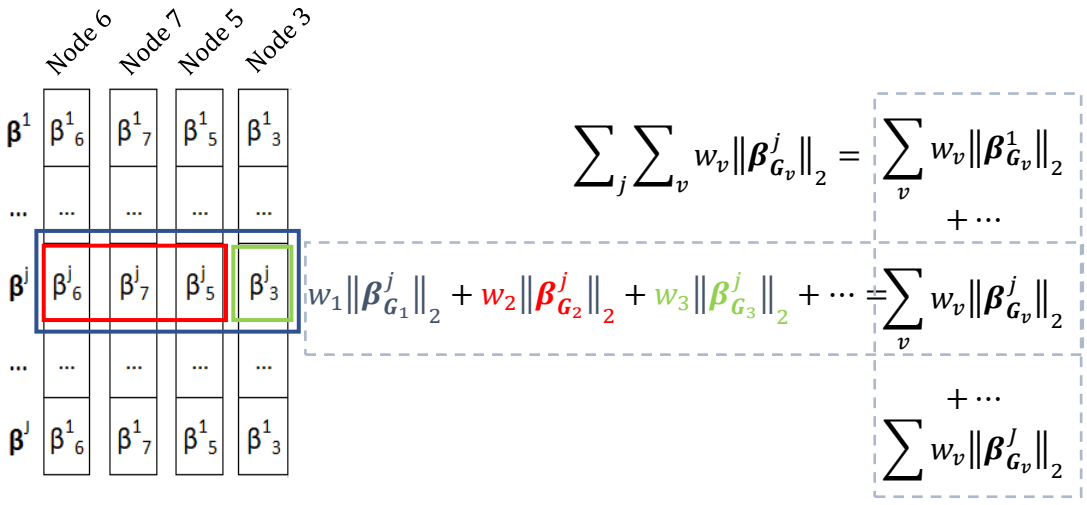

Figure 8: Visualization for the hierarchical structured regularization representing group effect and individual effect on the features across different tasks

Although the proposed weighting schema seems to be complicated, it has property that ensures all the regression coefficients to be overall penalized in a balance manner across all nested overlapping groups. That says, even if each leaf node could belong to multiple sets $\mathbf{G}_v$ (e.g. leaf node 6 belongs to $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_4$, and $\mathbf{G}_6$), the sum of weights over all the sets that contain this leaf node is always one (i.e., $w_1 + w_2 + w_4 + w_6 = 1$). The property can be summarized in Proposition 1.

Proposition 1: For any leaf node v_k, the sum of the weight w_v for all nodes v along the path from v_k to v_{root} equals to 1.

Proof:

Assume $v_1, \cdots, v_M$ are the ordered nodes along the path from the leaf node v_k to the root v_{root}. Since the relationship of $g_v + s_v = 1$ holds for all $v \in \mathbf{V}$, from (4.5), we have the following relationships:

$\sum_{v\in\{1,\cdots,M\}} w_v = \prod_{m=1}^{M} s_m + \sum_{l=1}^{M} g_l \prod_{m=l+1}^{M} s_m$

$= s_1 \prod_{m=2}^{M} s_m + g_1 \prod_{m=2}^{M} s_m + \sum_{l=2}^{M} g_l \prod_{m=l+1}^{M} s_m$

$= (s_1 + g_1) \cdot \prod_{m=2}^{M} s_m + \sum_{l=2}^{M} g_l \prod_{m=l+1}^{M} s_m$

$= \prod_{m=2}^{M} s_m + \sum_{l=2}^{M} g_l \prod_{m=l+1}^{M} s_m = \cdots = 1$

■

To summarize, our proposed method is the optimization in (4.1) with weight w_v given in (4.5). According to (4.5), w_v is a function of $\{(g_{v_i}, s_{v_i}): v_i \in \mathbf{V}\backslash\mathbf{V}_l, i.e., the\ set\ of\ internal\ nodes\}$. Therefore, the problem becomes how to choose (g_{v_i}, s_{v_i}) and indeed s_{v_i} due to the constraint of $g_{v_i} + s_{v_i} = 1$. Next, we discuss how to

choose s_{v_i}. We propose the following strategy: Because s_{v_i} reflects the extent to which the regression coefficients in $\boldsymbol{\beta}^j_{\mathbf{G}_{s_{v_i}}}$ should be selected separately and the larger the s_{v_i}, and more the regression fitting favors separate selection, we propose to make s_{v_i} in proportion to the distance between node v_i and the bottom level of the tree. The rationale is that the farther away of v_i from the bottom of the tree, the more likely the regression models in the leaf nodes contained in $\mathbf{G}_{s_{v_i}}$ should be different from each other. Using Figure 7 as an example, since node 1 ($v_i = 1$) is three levels up from the bottom level, $s_1 = 3$. Likewise, $s_2 = 2$, $s_4 = 1$, and $s_6 = 0$. Normalize the weights to make the one corresponding to the root node equal to one. Then, $s_1 = 1$, $s_2 = 0.67$, and $s_4 = 0.33$, and $s_6 = 0$. Using $g_v = 1 - s_v$, $g_1 = 0$, $g_2 = 0.33$, and $g_4 = 0.67$, and $g_6 = 1$. Therefore, the regularization corresponding to the tree in Figure 7 becomes:

$$\sum_j \sum_v w_v \left\| \boldsymbol{\beta}^j_{\mathbf{G}_v} \right\|_2$$

$$= 0.33 \left\| \boldsymbol{\beta}^j_{\mathbf{G}_2} \right\|_2 + 0.45 \left\| \boldsymbol{\beta}^j_{\mathbf{G}_4} \right\|_2 + 0.22 \left| \beta^j_6 \right| + 0.22 \left| \beta^j_7 \right| + 0.67 \left| \beta^j_5 \right| + \left| \beta^j_3 \right|.$$

From above deviation, we can see that leaf node 3 is without any group effect and has the strongest individual regularization since it is at the highest level compared to other leaves and doesn't share any similarities with others. Leaf node 6 and 7 have the strongest group regularization since they are at the bottom level of the tree and hence share more similarities. This example also verifies the property derived in Proposition 1. For instance, leaf node 6 belongs to $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_4$, and $\mathbf{G}_6$ and the corresponding weights are 0, 0.33, 0.45, and 0.22 which sums to 1.

4.2.2 HierML Model: Model Parameter Estimation

To solve the optimization problem in (4.1), we first need to convert our formulation into convex so that we apply an alternative formulation in Bach (2008) as below, let $B = \{\boldsymbol{\beta}_{v_l}\}_{v_l \in \mathbf{V}_l}$,

$$\hat{B} = \underset{B}{\operatorname{argmin}} \sum_{v_l} \left(\mathbf{y}_{v_l} - \mathbf{X}_{v_l}\boldsymbol{\beta}_{v_l}\right)^T \left(\mathbf{y}_{v_l} - \mathbf{X}_{v_l}\boldsymbol{\beta}_{v_l}\right) + \lambda \left(\sum_j \sum_v w_v \left\|\boldsymbol{\beta}_{\mathbf{G}_v}^j\right\|_2\right)^2. \quad (4.6)$$

The above objective function is convex, but the penalty function is not smooth. We now introduce an equivalent formulation with additional variables d_j's which leads to a closed-form solution of B. We re-write the objective function of (4.6) below:

$$L(B, D) = \sum_{v_l} \left(\mathbf{y}_{v_l} - \mathbf{X}_{v_l}\boldsymbol{\beta}_{v_l}\right)^T \left(\mathbf{y}_{v_l} - \mathbf{X}_{v_l}\boldsymbol{\beta}_{v_l}\right) + \lambda \sum_j \sum_v d_j^{-1} \left(w_v \left\|\boldsymbol{\beta}_{\mathbf{G}_v}^j\right\|_2\right)^2.$$

We need to show that if $\hat{B}$ is the optimal solution for $min\{L(B, D)\}$, then $L(\hat{B}) \leq L(\hat{B}, \widehat{D})$ and the equal sign is obtained when $d_j = \frac{w_v \left\|\hat{\boldsymbol{\beta}}_{\mathbf{G}_v}^j\right\|_2}{\sum_j \sum_v w_v \left\|\hat{\boldsymbol{\beta}}_{\mathbf{G}_v}^j\right\|_2}$ and $\widehat{D} = Diag(\hat{d}_1, \cdots, \hat{d}_J)$. Specifically, we need to show

$$\left(\sum_j \sum_v w_v \left\|\boldsymbol{\beta}_{\mathbf{G}_v}^j\right\|_2\right)^2 \leq \sum_j \sum_v d_j^{-1} \left(w_v \left\|\boldsymbol{\beta}_{\mathbf{G}_v}^j\right\|_2\right)^2. \quad (4.7)$$

Since $\sum_v w_v \left\|\boldsymbol{\beta}_{\mathbf{G}_v}^j\right\|_2$ is a scalar, let $u_j = \sum_v w_v \left\|\boldsymbol{\beta}_{\mathbf{G}_v}^j\right\|_2$ and $\boldsymbol{u} = (u_1, \cdots, u_J)$. Thus, we are going to show that $\|\boldsymbol{u}\|_1^2 \leq \sum_j d_j^{-1} (u_j)^2$. By Cauchy-Schwarz inequality, we have

$$\|\boldsymbol{u}\|_1 = \sum_j d_j^{\frac{1}{2}} d_j^{-\frac{1}{2}} |u_j| \leq \left(\sum_j d_j\right)^{\frac{1}{2}} \left(\sum_j d_j^{-1} u_j^2\right)^{\frac{1}{2}} \leq \left(\sum_j d_j^{-1} u_j^2\right)^{\frac{1}{2}}.$$

Equal sign is achieved if and only if $\sum_j d_j = 1$ and $d_j = \frac{u_j}{\|\boldsymbol{u}\|_1}$. Therefore, the above inequality (4.7) is valid and the equal sign holds if and only if $\sum_j d_j = 1$ and $d_j = \frac{w_v \left\|\widehat{\boldsymbol{\beta}}^j_{\mathbf{G}_v}\right\|_2}{\sum_j \sum_v w_v \left\|\widehat{\boldsymbol{\beta}}^j_{\mathbf{G}_v}\right\|_2}$. Our optimization problem reduces to minimize

$$\sum_{v_l} \left(\mathbf{y}_{v_l} - \mathbf{X}_{v_l}\boldsymbol{\beta}_{v_l}\right)^T \left(\mathbf{y}_{v_l} - \mathbf{X}_{v_l}\boldsymbol{\beta}_{v_l}\right) + \lambda \sum_j \sum_v d_j^{-1} \left(w_v \left\|\boldsymbol{\beta}^j_{\mathbf{G}_v}\right\|_2\right)^2$$

$$\text{subject to } \sum_j \sum_v d_{j,v} = 1 \text{ and } d_{j,v} \geq 0.$$

We solve the problem by optimizing $\boldsymbol{\beta}_{v_l}$'s and $d_{j,v}$'s alternatively in a fixed number of iterations. In each iteration, we first fix $\boldsymbol{\beta}_{v_l}$, and update $d_{j,v}$ by above deviation. Then, hold $d_{j,v}$ as constant and solve for $\boldsymbol{\beta}_{v_l}$ by differentiating the objective function respect to $\boldsymbol{\beta}_{v_l}$ and solve system equations for $\boldsymbol{\beta}_{v_l}$ as follow

$$\boldsymbol{\beta}_{v_l} = (\mathbf{X}_{vl}^T \mathbf{X}_{vl} + \lambda \mathbf{D})^{-1} \mathbf{X}_{vl}^T \boldsymbol{y}_{vl}$$

where **D** is p × p diagonal matrix with *j*th diagonal element $d_j = \frac{w_v \left\|\widehat{\boldsymbol{\beta}}^j_{\mathbf{G}_v}\right\|_2}{\sum_j \sum_v w_v \left\|\widehat{\boldsymbol{\beta}}^j_{\mathbf{G}_v}\right\|_2}$.

4.2.3 HierML Model: Tree Growing Algorithm

Finally, we discuss how to grow the tree. The inputs to the algorithm are a training set and a validation set of which each includes indoor variable **X**, outdoor variable **Z**, and response variable Y. We assume that the tree is at the s-th step of the tree growing process and node v_s is going to be split next. The algorithm is summarized by the steps below:

<u>*Step 1*</u>: If the sample size of the node v_s in the training set is smaller than n_{min}, stop splitting this node; otherwise, proceed to Step 2.

<u>*Step 2*</u>: For each environmental variable $Z_j \in \mathbf{Z}$ and each candidate splitting point z_j, split v_s into a left child node defined by $Z_j \leq z_j$ and a right child node defined by $Z_j > z_j$. Fit a

lasso for each child node on the training set and obtain the estimates $\widehat{\boldsymbol{\beta}}_{tr}^{(z_j \le z_j)}$ and $\widehat{\boldsymbol{\beta}}_{tr}^{(z_j > z_j)}$. Then compute the empirical risk reduction $\Delta\hat{R}_{val}^{j}$ on validation set by using

$$\Delta\hat{R}_{val}^{j} = \hat{R}_{val}\left(\widehat{\boldsymbol{\beta}}_{tr}^{(v_s)}\right) - \hat{R}_{val}\left(\widehat{\boldsymbol{\beta}}_{tr}^{(z_j \le z_j)}\right) - \hat{R}_{val}\left(\widehat{\boldsymbol{\beta}}_{tr}^{(z_j > z_j)}\right)$$

Step 3: If no positive $\Delta\hat{R}_{val}^{j}$ is found, stop splitting node v_s; otherwise, split node v_s using the environmental variable and splitting point with the largest $\Delta\hat{R}_{val}^{j}$.

Step 4: Apply HierML model to update the regression models of all leaf nodes v_l's of the current tree and obtain the estimates $\{\widehat{\boldsymbol{\beta}}_{v_l}\}_{v_l \in \boldsymbol{V}_l}$. Then compute the empirical risk $\{\hat{R}_{val}(\widehat{\boldsymbol{\beta}}_{v_l})\}_{v_l \in \mathbf{V}_l}$ on validation set for each leaf node.

Step 5: Choose the leaf node that has the largest empirical risk (i.e. $argmax_{v_l \in \mathbf{V}_l}\{\hat{R}_{val}(\widehat{\boldsymbol{\beta}}_{v_l})\}$) for further split. Then repeat steps 1 to 3.

In this section, we first develop a HierML model that performs joint selection of features across all the leaf nodes with consideration of the similarities shared by the adjacent leaf nodes. We propose a hierarchical structured penalty term that integrates group effect and individual effect on feature selection. A novel weighting schema is designed for adjusting the strength of regularization on each of the groups of the regression parameters. We then convert the original objective function to a convex funcction and introduce an equivalent formulation to overcome the issue of non-smoothness in the penalty term. A closed form estimation of model parameters is derived. Finally, a tree growing algorithm is developed to select the outdoor variable for split and update the regression models of all the leaf nodes at each step.

4.3 Simulation Studies

In this section, we demonstrate the performance of our method on simulated datasets and compare the results with those from three competing methods: SPR, MOB, and GUIDE. We evaluate these methods by Mean Squared Prediction Error (MSPE) on testing sets. In what follows, we start with introducing the data generation process of the environmental, input, and output variables.

We consider five environmental variables that are uniformly distributed in the unit hyper-cube $[0,1]^5$. We assume that the first two environmental variables, Z_1 and Z_2, are the ones that truly partition the space of environmental variables into different subdivisions, while the remaining three variables are noise. Specifically, Figure 7(a) shows the ground-truth tree structure in which the leaf nodes (i.e., the subdivisions) are generated by median splits on Z_1 and Z_2. In each leaf node/subdivision, we consider 75 input variables and generate samples from a multivariate normal distribution $N(\mathbf{0}, \mathbf{\Sigma}_{75\times75})$. Each element of $\mathbf{\Sigma}_{75\times75}$ is set to be $\sigma_{ij} = 0.5^{|i-j|}, i, j = 1, \cdots, 75$, to account for the potential correlation between input variables. Furthermore, we generate samples for the output variable in each subdivision using a linear regression. The key is properly choosing regression coefficients to account for the multilevel correlation structure of the leaf nodes as shown in Figure 9. With this consideration, we set the regression coefficients in each leaf node to follow the pattern in Figure 10. Each column corresponds to a leaf node and each row corresponds to one out of 75 coefficients (for 75 input variables). Non-zero/zero coefficients are shown as black/white boxes. In each leaf node, we assume that 10 out of the 75 input variables have non-zero coefficients. Positions of the 10 non-zero coefficients (i.e., black boxes) are chosen according to the following considerations: Leaf nodes at the lowest level should the

most similar schema for non-zero coefficients. In our case, node 8 and node 9 share eight positions of non-zero coefficients. Furthermore, the higher level of a leaf node in the tree, the fewer shared positions of non-zero coefficients it should have with node 8 or 9. Finally, to decide the magnitudes and signs of non-zero coefficients, we sample each coefficient from $N(0,1) + 3$.

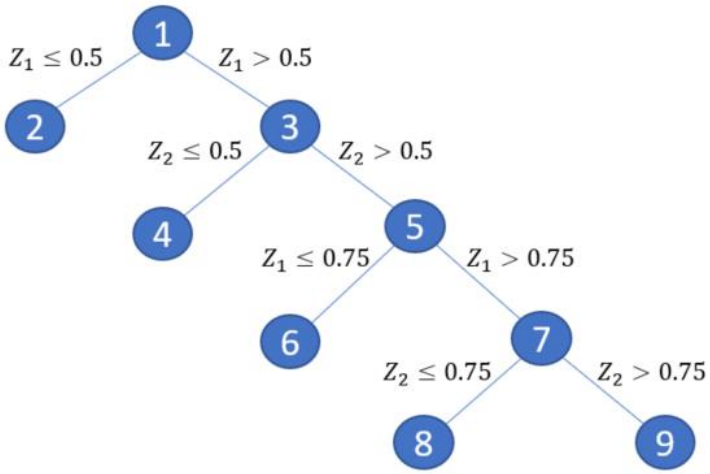

Figure 9: True tree structure partitioned by Z_1 and Z_2

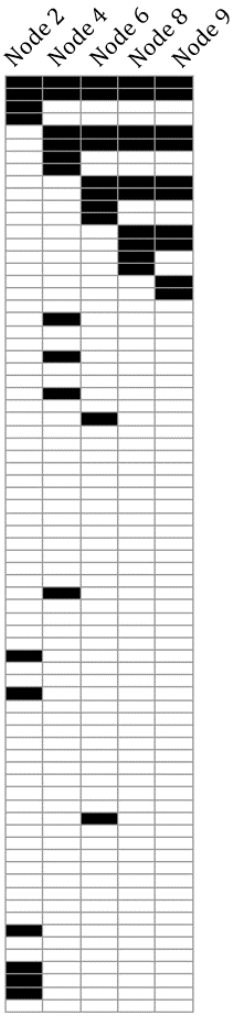

Figure 10: Pattern of regression coefficients within each leaf node of the tree

Following the afore-described data generation process, we generate a training set and a validation set with 2000 samples, respectively. Under this setting, the lowest leaf nodes contain around 125 samples, which is a small sample size scenario compared with 75 input variables. Then, we apply the proposed method to the simulated data. For comparison purposes, we also apply three competing methods to the same simulation datasets: SPR, GUIDE, and MOB. In applying each method, tuning parameters are chosen to minimize the MSPE on the validation set. Moreover, we run another more challenging experiment with only 1000 samples in the training and validation set, respectively. This

means around 60 samples at each lowest leaf node. We evaluate the performance of all four methods in the two experiments (i.e., sample sizes equal to 2000 and 1000) on another independently simulated test set. This process is repeated for 100 times. We report the MSPE of each method on the test set in Table 6. It can be clearly seen that our proposed model has significantly higher prediction accuracy than other competing methods. SPR performs worse, especially in the case of 1000 samples; this is because SPR fits models independently at each leaf node rather than considering the similarities among neighborhood nodes and jointly estimating with each other. MOB and GUIDE perform worse because both of methods fit ordinary linear regression models within each node and their prediction performances are significantly influenced by the "small n, large p" setting in our simulation. Moreover, their splitting criteria are based on statistical hypothesis tests, which are also sensitive to the sample size at every split.

Table 6: MSPE on testing data for four methods under two different sample sizes

	Sample size=2000	**Sample size=1000**
TBSR	0.49	1.81
SPR	0.73	3.11
MOB	1.80	10.61
GUIDE	39.32	45.20

Furthermore, noting that among the three competing methods, SPR follows the same splitting criteria with our proposed approach (i.e. reduction on risk) so it makes sense to compare the splitting outcomes between our approach and SPR. We count the number of times that the ground-truth tree structure as shown in Figure 9 is recovered by SPR and our method, respectively, within 100 simulation runs. The results are shown in Table 7,

together with the MSPE on test data computed only on the fully-recovered runs. We can observe that our proposed model has a better chance to fully recover the ground-truth structure. This is because SPR is more likely to stop splitting early since the models are poorly fitted in the nodes that are at the lower level and has limited samples. However, our proposed approach is benefit from fitting the models in a collaborative manner with other leaf nodes, which results in having greater chance to split on the correct environmental variables and reduce the overall risk. For the fully recovered runs, the MSPE of our model slightly outperforms SPR when the overall sample size is 2000; while our model shows a better prediction accuracy when the sample size is reduced to 1000.

Table 7: Recovery of true tree structure and MSPE in fully-recovered runs in comparison with TBSR and SPR

	# of Full Recovery		**MSPE**	
	Sample size=2000	**Sample size=1000**	**Sample size=2000**	**Sample size=1000**
TBSR	94	81	0.45	0.9
SPR	69	57	0.54	1.99

Finally, to understand performance of the proposed method in each leaf node (i.e., each subdivision defined on the environmental variables), we compute the MSPE and Pearson correlation between the true and predicted responses on test data within each leaf node. The results are summarized in Table 8. It can be seen that the leaf nodes that are closed to the root node have better prediction performances (i.e. smaller MPSE and larger Pearson correlation) than the leaf nodes at the bottom level (i.e. node 8 and node 9). This makes sense because node 8 and node 9 have the smallest sample sizes. Moreover, when we reduce the overall sample size to 1000, the prediction performance at each leaf mode

becomes worse, especially for the bottom leaf nodes as even fewer samples exist in these nodes.

Table 8: MSPE and Pearson Correlation on testing data for each leaf node

	MSPE		**Pearson Correlation**	
leaf nodes	**Sample size=2000**	**Sample size=1000**	**Sample size=2000**	**Sample size=1000**
node2	0.25(0.01)	0.26(0.02)	1(0.00)	1(0.00)
node4	0.26(0.02)	0.26(0.02)	1(0.00)	1(0.00)
node6	0.26(0.03)	0.30(0.24)	1(0.00)	1(0.00)
node8	2.08(2.86)	5.00(3.94)	0.99(0.02)	0.96(0.03)
node9	2.18(2.76)	5.60(3.92)	0.99(0.02)	0.96(0.03)

4.4 Application

Nowadays, recent studies indicate that energy spent in buildings represents about 40% of the total energy consumed, where more than a half is used by Heating, Ventilation, and Air Conditioning (HVAC) systems (Álvarez et al., 2013). A lot of energy are wasted during the process of HVAC maintaining the comfort level of a home. To reach a balance between energy consumption and streamlined comfort, a new type of solar-powered house was built which has a forecasting module that could allow the house to adapt itself to future temperature conditions in an energy-efficient manner (Zamora et al., 2014). The module, by using predicted values of temperature, could help to decide whether to activate or not the HVAC system to maintain current temperature regardless its prevent value. The aim is to derive a prediction model to be used for a more efficient temperature control to reduce HVAC system energy consumption.

In this section, we present an application of our proposed model to predict indoor temperature using other indoor controllable variables and outdoor uncontrollable

environmental variables. The real data set was recorded at a modular house built in Madrid, Spain, in March and June 2011 (Bache and Lichman, 2013). A system with indoor and outdoor sensors captured each data sample for every 15-minute interval. Six indoor and six outdoor variables are used to predict the overall indoor temperature in this application. A full list of all the variables and their physical meanings are given in Table 9. The time span of this dataset is 42 days, which results in a total of 4137 samples. We split the entire dataset into a training set (50% of data), a validation set (25% of data), and a test set (25% of data).

Table 9:Abbreviation and physical meanings of indoor, outdoor, and output variables in indoor temperature prediction modeling

	Variable abbreviation [unit]	Physical meaning
Indoor variables	CO_2_dining [ppm]	CO_2 in kitchen area
	CO_2_living [ppm]	CO_2 in living room area
	RH_dining [%]	Relative humidity in kitchen area
	RH_living [%]	Relative humidity in living room area
	Light_dining [Lux]	Lighting in kitchen area
	Light_living [Lux]	Lighting in living room area
Outdoor variables	Outdoor_temp[℃]	Outdoor temperature [9.4, 29.9]
	Outdoor_humidity[%]	Outdoor relative humidity [22.7, 83.6]
	Rain	The proportion of the last 15 minutes where rain was detected (a value in range [0,1])
	Windspeed[m/s]	Windspeed [0.0, 4.9]
	Sun_light [Lux]	Sun light [0.6, 625.0]
	Sun_irradiance [W/m^2]	Sun irradiance [0.0, 975.6]
Output variable	Indoor temperature [℃]	Indoor temperature

We apply TBSR to the dataset. Figure 11 shows the tree structure found by the proposed method. Half of the environmental variables are used in the tree generation, including *Sun_light*, *Sun_irradiance*, and *Outdoor_temp*. Environmental features related

to the sun contribute the most in the partitioning process. This is not hard to understand because the outdoor conditions are changing as the sun moves and the measurements on the sun shift more and faster than other measurements. When sun light is below a threshold (e.g., 312.8 lux), it could be an indication of turning dark and a prediction model for the night needs to start functioning. Sun irradiance is selected to further split the right branch of the tree. During the daytime, the indoor temperature prediction system is sensitive to the sun energy received on the exterior surface of the house. If sun irradiance is low, cloudy weather is likely present and the temperature is relatively stable. When sun irradiance is high, outdoor temperature becomes a significant factor that affects the indoor temperature prediction. High outdoor temperature will cause more HVAC operations in order to cool down the house. Therefore, fitting prediction models in different bins of outdoor temperature would help HVAC to better adjust activation/deactivation strategy and reduce energy consumption.

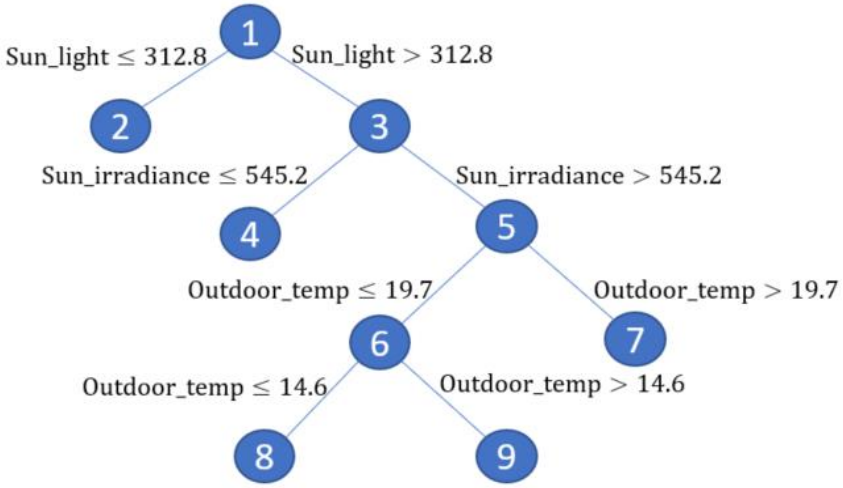

Figure 11: Tree structure found by our proposed approach

Furthermore, we examine the fitted model coefficients in each of the leaf node. The indoor variable selected by all the regression models is relative humidity in living room. This makes sense, as temperature and relative humidity are highly correlated, and people usually spend a lot of time on activities at living room area as well, all of which contribute to the changes on the indoor temperature. We also notice that all indoor variables are selected by the regression models in leaf node 2 and leaf node 4. These two nodes are partitioned in the range of low sun light and low sun irradiance, which indicates that the prediction models in these nodes are probably used for evening or night when people usually stay at home and have influence on these indoor measurements. Based on the idea of our proposed approach, the fitted models in the leaf nodes (i.e., node 8 and node 9) that are at the bottom of the tree should be similar. In Figure 12, regression model in node 8 only has one variable that is selected differently from the fitted model in node 9; all other coefficients are either selected or are shrunk to zero in both models.

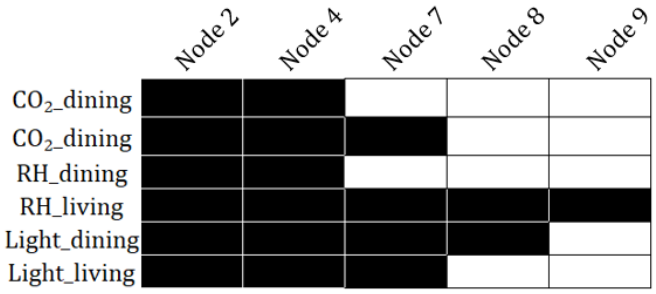

Figure 12: Zero (white) and non-zero (black) coefficients of the fitted regression model in each leaf node of the tree in Figure 9

In addition, for comparison, we fit the SPR model on the same data sets. We compute the MSPEs of the two approaches on the same test data, which are 4.57 and 7.27 for our approach and SPR, respectively. Our model improves the prediction accuracy by

37% over SPR. We also make a scatter plot with predicted indoor temperature against true indoor temperature in Figure 13, in which a linear trend can be observed and our model indicates a good prediction capability.

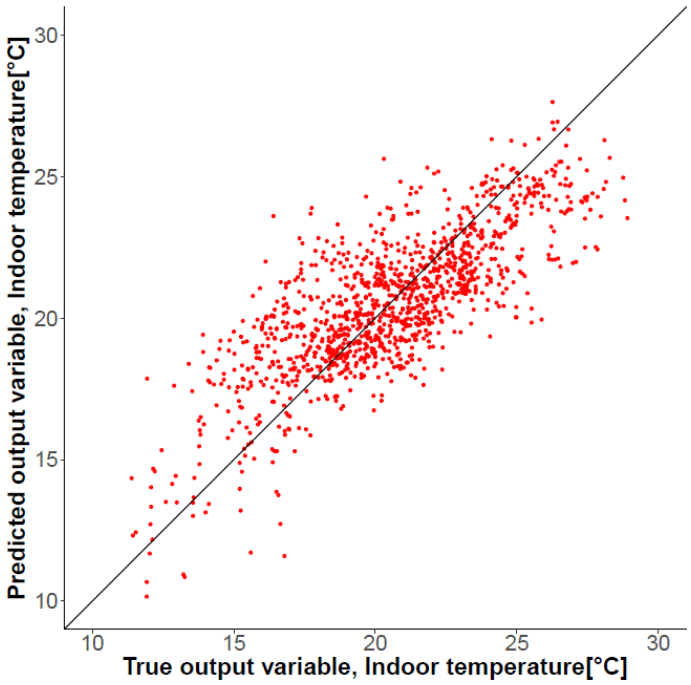

Figure 13: Predicted vs. true output variable (indoor temperature) on the test set by our proposed approach

4.5 Conclusion

In this research, we developed an TBSR for tackling the nonlinear interactions between indoor and outdoor variables in predicting building energy consumption. TBSR allows more controls and adjustments on outdoor variables so that the relationship between response and indoor variables could be modeled more appropriately. We proposed a HireML approach to jointly estimate the regression models within each leaf nodes and developed an efficient algorithm for tree growing. We conducted simulation studies to demonstrate the better prediction accuracy of TBSR than other competing methods. Finally, an application of reducing HVAC system energy consumption in solar-powered house is presented.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In my dissertation, I developed two approaches, SPR and TBSR, for modeling the complicated interaction between a system and the multivariate environment under which it operates. To our best knowledge, SPR was the first of its kind that integrated recursive partitioning and high-dimensional regression model fitting within a single framework. As an extension of SPR, TBSR outcomes the limitation of SPR that the regression model fitting at each subdivision is independent by jointly estimating these models under a structured weighted schema, which is especially advantageous when the sample size of each subdivision is small relative to the dimensionality of regression models. I extensively analyzed the theoretical properties of SPR estimators and quantified their risks from Bayes' risk. I applied both approaches to real building energy datasets and achieved interpretable models with good prediction accuracies.

My research can be extended to incorporate ensemble methods similar to bagging, boosting, and random forest to reduce the variability of the recursive partitioning. Optimization algorithms can be developed to search for the partition with a better optimality property. Both the SPR and TBSR have broad applicability to domains beyond building energy management, including but not limited to, mobile communication networks and wind energy as presented in Introduction, as well as bioinformatics in studying gene-environment interactions in related to diseases or disease traits.

REFERENCES

Álvarez, J. D., Redondo, J. L., Camponogara, E., Normey-Rico, J., Berenguel, M., and Ortigosa, P. M. (2013). Optimizing building comfort temperature regulation via model predictive control. *Energy and Buildings*, *57*, 361-372.

Argyriou, A., Evgeniou, T., and Pontil, M. (2008). Convex multi-task feature learning. *Machine Learning*, 73(3), 243-272.

Bach, F. R. (2008). Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, *9*(6), 1179-1225.

Bache, K. and Lichman M. (2013). UCI Machine Learning Repository. http://archive.ics.uci.edu/ml.

Bien, J., Taylor, J. and Tibshirani, R. (2013). A lasso for hierarchical interactions. *The Annals of Statistics*, 41(3), 1111-1141.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, 37(4), 373-384.

Breiman, L., Friedman, J., Stone, C. J. and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Byon, E., Choe, Y., and Yampikulsakul, N. (2015). Adaptive Learning in Time-Variant Processes With Application to Wind Power Systems. *IEEE Transactions on Automation Science and Engineering*, 99, 1-11.

Cai, Z., Fan, J. and Li, R. (1999). Generalized Varying-coefficient models. Department of Statistics, UCLA.

Chan, K. Y., and Loh, W. Y. (2004). LOTUS: An algorithm for building accurate and comprehensible logistic regression trees. *Journal of Computational and Graphical Statistics*, 13(4), 826-852.

Chaudhuri, P., Huang, M. C., Loh, W. Y., and Yao, R. (1994). Piecewise-polynomial regression trees. *Statistica Sinica*, 4, 143-167.

Chiang, C. T., Rice, J. A. and Wu, C. O. (2001). Smoothing spline estimation for varying coefficient models with repeatedly measured dependent variables. *Journal of the American Statistical Association*, 96(454), 605-619.

Chipman, H. (1996). Bayesian variable selection with related predictors. *Canadian Journal of Statistics*, 24(1), 17-36.

Choi, N. H., Li, W. and Zhu, J. (2010). Variable selection with the strong heredity constraint and its oracle property. *Journal of the American Statistical Association*, 105(489), 354-364.

Eisenhower, B., O'Neill, Z., Narayanan, S., Fonoberov, V. A. and Mezić, I. (2012). A methodology for meta-model based optimization in building energy models. *Energy and Buildings*, 47, 292-301.

Fan, J., Yao, Q. and Cai, Z. (2003). *Adaptive varying coefficient linear models. Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1), 57-80.

Fan, J., and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456), 1348-1360.

Fan, J. and Zhang, W. (1999). Statistical estimation in varying coefficient models. *Annals of Statistics*, 1491-1518.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*.

Friedman, J. H. and Popescu. B. E. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3), 916–954.

Hardy, W. C., and Preface By-Cardoso, L. (2001). *QoS: measurement and evaluation of telecommunications quality of service*. John Wiley & Sons, Inc..

Hamada, M. and Wu, C. J. (1992). Analysis of designed experiments with complex aliasing. *Journal of Quality Technology*, 24(3), 130-137.

Hastie, T., and Tibshirani, R. (1993). Varying-coefficient models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 757-796.

Hoover, D. R., Rice, J. A., Wu, C. O., and Yang, L. P. (1998). Nonparametric smoothing estimates of time-varying coefficient models with longitudinal data. *Biometrika*, 85(4), 809-822.

Hu, T. and Xia, Y. (2012). Adaptive semi-varying coefficient model selection. *Statistica Sinica*, 575-599.

Jacob, L., Obozinski, G., and Vert, J. P. (2009). Group lasso with overlap and graph lasso. *In Proceedings of the 26th International Conference on Machine Learning*, 433-440.

Jenatton, R., Audibert, J. Y., and Bach, F. (2011). Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, *12*(10), 2777-2824.

Kim, H. and Loh, W. Y. (2001). Classification trees with unbiased multiway splits. *Journal of the American Statistical Association*, 96(454).

Kim, S., and Xing, E. P. (2010). Tree-guided group lasso for multi-task regression with structured sparsity. *In Proceedings of the 27th International Conference on Machine Learning*, 543–550.

Kim, Y., Kim, J., and Kim, Y. (2006). Blockwise sparse regression. *Statistica Sinica*, 16, 375-390.

Landwehr, N., Hall, M., and Frank, E. (2005). Logistic model trees. *Machine Learning*, 59(1-2), 161-205.

Lee, G., Ding, Y., Genton, M. G., and Xie, L. (2014). Power curve estimation with multivariate environmental factors for inland and offshore wind farms. *Journal of the American Statistical Association*, (just-accepted).

Loh, W. Y., and Shih, Y. S. (1997). Split selection methods for classification trees. *Statistica sinica*, 7(4), 815-840.

Loh, W. Y. (2002). Regression tress with unbiased variable selection and interaction detection. *Statistica Sinica*, 12(2), 361-386.

Loh, W. Y. (2006). Logistic regression tree analysis. *In Springer Handbook of Engineering Statistics* (pp. 537-549). Springer London.

Loh, W. Y. (2008). Regression by parts: fitting visually interpretable models with GUIDE. *In Handbook of Data Visualization*, 447-469.

Ma, S., Yang, L., Romero, R., and Cui, Y. (2011). Varying coefficient model for gene–environment interaction: a non-linear look. *Bioinformatics*, 27(15), 2119-2126.

Montgomery, D. C., Peck, E. A. and Vining, G. G. (2015). *Introduction to linear regression analysis*. John Wiley & Sons.

Obozinski, G., Taskar, B., and Jordan, M. I. (2010). Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, *20*(2), 231-252.

Quinlan, J. R. (1992). Learning with continuous classes. *In 5th Australian Joint Conference on Artificial Intelligence*, 92, 343-348.

Radchenko, P. and James, G. M. (2010). Variable selection using adaptive nonlinear interaction structures in high dimensions. Journal of the American Statistical Association, 105(492), 1541-1553.

Rusch, T., and Zeileis, A. (2013). Gaining insight with recursive partitioning of generalized linear models. *Journal of Statistical Computation and Simulation*, 83(7), 1301-1315.

Su, X., Wang, M., and Fan, J. (2004). Maximum likelihood regression trees. *Journal of Computational and Graphical Statistics*, 13(3), 586-598.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*. Series B (Methodological), 267-288.

Torgo, L. (1997). Functional models for regression tree leaves. *In Proceedings of International Conference on Machine Learning*, 97, 385-393.

Van de Geer, Sara A. (2008). High-dimensional generalized linear models and the lasso. The Annals of Statistics , 614-645.

Vapnik, V. (1998). Statistical learning theory. 1998.

Xia, Y. and Li, W. K. (1999). On the estimation and testing of functional-coefficient linear models. *Statistica Sinica*, 735-757.

Xiong, T., Bi, J., Rao, R. B., and Cherkassky, V. (2007). Probabilistic Joint Feature Selection for Multi-task Learning. *SDM*, 332-342.

Yuan, M., Joseph, V. R. and Zou, H. (2009). Structured variable selection and estimation. The Annals of Applied Statistics, 1738-1757.

Yuan, M., and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1), 49-67.

Zamora-Martínez, F., Romeu, P., Botella-Rocamora, P., and Pardo, J. (2014). On-line learning of indoor temperature forecasting models towards energy efficiency. *Energy and Buildings*, 83, 162-172.

Zeileis, A., Hothorn, T., and Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2), 492-514.

Zhao, P., Rocha, G. and Yu, B. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, 3468-3497.

Zhang, W., Lee, S. Y. and Song, X. (2002). Local polynomial fitting in semivarying coefficient model. *Journal of Multivariate Analysis*, 82(1), 166-188.

Zhang, J., Ghahramani, Z., and Yang, Y. (2008). Flexible latent variable models for multi-task learning. *MachineFurther Learning*, 73(3), 221-242.

Zhou, X., and You, J. (2004). Wavelet estimation in varying-coefficient partially linear regression models. *Statistics & probability letters*, 68(1), 91-104.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2), 301-320.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476), 1418-1429.