– Doctoral Dissertation –

Statistical inference for feature selection algorithms and how to construct robust learning models

(特徴量選択アルゴリズムに関する 統計的推論とロバスト学習の構築に向けた研究)

Shinya Suzumura

鈴村　真矢

January 11, 2021

Contents

List of Figures

List of Tables

Abstract

This paper consists of two parts. First, we propose a new method for quantifying the statistical significance of the features which are selected by a sparse machine learning algorithm. The proposed method can select high-order interactions of features related to responses (e.g., drug resistance of patients) and quantify the association with the responses. In our experiment, several combinations of mutations in gene sequence associated with HIV-1 drug resistance were selected, and the confidence interval of the

fitted coefficients for the selected combinations were provided. In the second part, we propose a new robust learning method that stabilizes the learning results by automatically controlling the degree of influence of outliers. Although it is possible to obtain training data relatively easily by using crowdsourcing in recent years, the learning results become unstable due to anomaly samples in training datasets. The motivation for this research is to overcome such a problem. The proposed method is an annealing method with a continuous temperature parameter, where the parameter can be regarded as the degree of influence of outliers. Although most of conventional methods are also annealing based approaches, the temperature parameter can NOT be continuously changed because of computational efficiency – there is a trade-off between computational time and generalization performance, that is a severe problem in robust learning. Our proposed method can change the parameter continuously by using a new homotopy approach where the optimal solution of the model can be calculated by piecewise linear function with respect to the parameter. Our experiments showed that generalization performance and computational time of the proposed method are better than the conventional method.

Part I

Statistical inference for feature selection algorithms

1 Introduction

In this paper, we consider a stepwise feature selection algorithm for a high-order interaction model that has r-th order interactions of multiple features, and we propose a new statistical inference for selected high-order interaction features. Feature selection and statistical inference for high-order interaction features are important tasks. For example, in a biomedical study, co-occurrence of multiple mutations in multiple genes may have a significant influence on a response to a drug even if occurrence of single mutation in each of these genes has no

influence [1, 2, 3]. In high-order interaction model, a difficulty of these tasks is that there is a huge number of possible combinations of multiple features. If one has a dataset with d original features and takes into account interactions up to order r, the model has $D = \sum_{\rho=1}^{r} \binom{d}{\rho}$ features (e.g., for $d = 10,000$, $r = 5$, $D > 10^{17}$). Feature selection and statistical inference in such an extremely high-dimensional model are challenging both computationally and statistically.

A common approach to high-dimensional modeling is to consider a sparse model, i.e., a model only with a selected subset of features. In the past two decades, considerable amount of studies have been done on sparse modeling and feature selection in high-dimensional models. In these studies, a variety of feature selection algorithms such as *marginal screening* [4], *orthogonal matching pursuit* [5], LASSO [6], and their various extensions have been developed. On the other hand, statistical inference for sparse models (hypothesis testing or confidence interval computation of the fitted coefficients) have not been deeply studied until recently. After the seminal work by [7], significant progress has been made on statistical inference for sparse linear models [7, 8, 9, 10, 11, 12, 13, 14, 15], and these approaches are sometimes called *Selective Inference* or *Post-Selection Inference*. The basic idea of selective inference is to consider the sampling distribution of test statistic after feature selection, where the sampling distribution is a conditional distribution which is characterized by selected and unselected features. By considering such a distribution, we can eliminate the *selection bias* [16] that arises in feature selection. In the following paragraph, we briefly discuss selection bias in testing selected features.

1.1 Selection bias in feature selection

A simple illustration of selection bias is depicted in Figure 1. In this example, we generated several images whose color was randomly generated based on the standard normal distribution $\mathrm{N}(0, 1^2)$. By using a selection algorithm, we selected a region which is composed of 3×3 pixels, and we considered a naive hypothesis testing that the expected average color of pixels on the selected region is zero or not (the detail will be described later). Although each color of pixels was randomly generated, the color level on the selected region tends to be higher, and *type I error* could not be controlled under the desired significance level. The

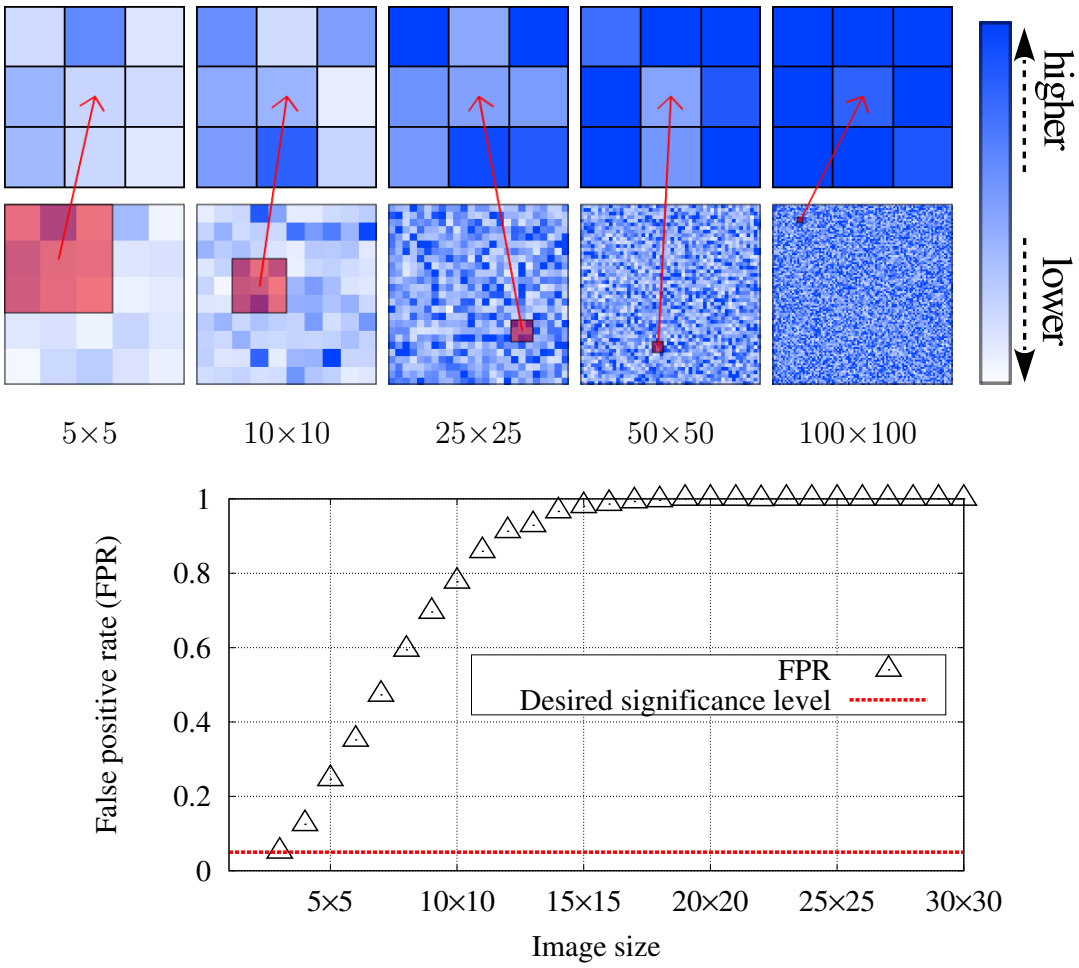


Figure 1: A simple demonstration of selection bias. In this example, we randomly generated 5×5, ..., 100×100 images whose color was generated based on normal distribution $\mathrm{N}(0, 1^2)$. In each image, we selected the region (which is composed of 3×3 pixels) whose average value is the highest compared with the other regions in an image. We observed that the color of selected region looks "significantly higher" when it is selected from larger image. This indicates that the selection bias depends on both selected and unselected regions. The bottom plot shows the frequencies of the false positive findings for various sizes of images, in which we applied a naive hypothesis testing where the null hypothesis is that the average value on the selected region is equals to 0, with desired 5% significant level.

detail of the hypothesis testing is shown as follows. Let τ_j be the average color of pixels on the j-th selected region in an image. Since each color of pixels was randomly generated by using $\mathrm{N}(0, 1^2)$, one expects that the sampling distribution of value τ_j is also $\mathrm{N}(0, 1^2)$. The

classical z-test $H_{j,0} : \mathbb{E}[\tau_j] = 0, s \sim \mathrm{N}(0, 1^2)$ is totally valid when j is completely randomly selected. On the other hand, $\mathbb{E}[\tau_j] \neq 0$ when j is selected based on observations of generated image. It is because the region would NOT be selected when the average value τ_j is relatively smaller than the others. This indicates that the distribution of value τ_j after selection is no longer normal distribution $N(0, 1^2)$, but rather a conditional distribution which is conditioned by the selected and unselected regions. To remove the bias, it is necessary to analyze the sampling space of the test statistic after selection.

1.2 Our contributions

Our main contribution is to develop a selective inference procedure for selected features in high-order interaction model. In high-order interaction model, there is a serious computational problem because the sampling distribution depends not only on selected features but also on an extremely large number of unselected features. For circumventing the computational issue, we consider a tree structure among features and derive a novel pruning condition that enables us to efficiently identify a set of features which have no effect on the sampling distribution.

1.3 Organization of the paper

Here is the outline of this paper. §2 presents problem formulation, illustrative example, formal description of selective inference, and a brief review of recent selective inference literature. §3 describes our main contribution, where we develop a method that enables selective inference in extremely high-dimensional settings. §4 discusses extensions for computational efficiency and statistical power of selective inference. §5 covers numerical experiments for demonstrating the advantage of selective inference framework. §6 concludes the paper.

2 Preliminaries

2.1 Notations

We use the following notations in the remainder. For any natural numbers n and d, we denote the set of indices as $[n] = \{1, \ldots, n\}$, and we denote a vector and a matrix as $\boldsymbol{y} \in \mathbb{R}^n$ and $Z \in \mathbb{R}^{n \times d}$, respectively. Furthermore, the i-th row and j-th column vectors of matrix $Z \in \mathbb{R}^{n \times d}$ are written as $\boldsymbol{z}_{i:} \in \mathbb{R}^{1 \times d}$, $\boldsymbol{z}_{:j} \in \mathbb{R}^{n \times 1}$ respectively, and (i, j)-th element of matrix Z is written as z_{ij}. For a set of indices $S \in [d]$, we denote a sub-matrix as $Z_S \in \mathbb{R}^{n \times |S|}$ whose column vectors are composed of S columns of the matrix Z.

2.2 Problem statement for high-order interaction model

Let $\boldsymbol{z}_{i:} = [z_{i1}, \ldots, z_{id}] \in [0, 1]^{1 \times d}$ be d-dimensional original features with i-th sample index. We consider the following high-order interaction model up to r-th order

$$f(\boldsymbol{z}_{i:}) = \sum_{j_1 \in [d]} \alpha_{j_1} z_{ij_1} + \sum_{\substack{(j_1, j_2) \in [d] \times [d] \\ j_1 \neq j_2}} \alpha_{j_1, j_2} z_{ij_1} z_{ij_2} + \cdots + \sum_{\substack{(j_1, \ldots, j_r) \in [d]^r \\ j_1 \neq \ldots \neq j_r}} \alpha_{j_1, \ldots, j_r} z_{ij_1} \cdots z_{ij_r}, \tag{1}$$

where αs are the coefficients, and the number of coefficients is $D = \sum_{j \in [r]} \binom{d}{j}$. For notation simplicity, we write the high-order interaction model (1) as the following linear model

$$f(\boldsymbol{x}_{i:}; \boldsymbol{\beta}) = \boldsymbol{x}_{i:} \boldsymbol{\beta} = \beta_1 x_{i1} + \cdots + \beta_D x_{iD}, \tag{2}$$

where $\beta_1, \ldots, \beta_D$ are D coefficients corresponding to $\alpha_{j_1}, \ldots, \alpha_{j_1, \ldots, j_r}$ in (1), and $\boldsymbol{x}_{i:}$ is D-dimensional high-order interaction features which is computed by multiplying original features. Although the bias term in (2) can be considered by appending an unit vector $\boldsymbol{x}_{:0} = \mathbf{1}$ and a coefficient β_0, we omit the explicit notation.

Our goal is to select high-order interaction features which are highly associated with the n-sample continuous responses $\boldsymbol{y} = (y_1, \cdots, y_n)^\top \in \mathbb{R}^n$, and provide the statistical inference of the association for each of the selected features. The original training set is denoted as $(Z, \boldsymbol{y}) \in [0, 1]^{n \times d} \times \mathbb{R}^n$, while the expanded training set is written as $(X, \boldsymbol{y}) \in [0, 1]^{n \times D} \times \mathbb{R}^n$. The short description of symbols is summarized in Table 1.

Table 1: The short description of symbols

Symbol	Description
n	The number of samples
d	The number of original features
D	The number of high-order interaction features
r	The maximum order of interactions
$\boldsymbol{Z} \in [0,1]^{n \times d}$	Original design matrix
$\boldsymbol{X} \in [0,1]^{n \times D}$	Design matrix for high-order interaction features
$\boldsymbol{y} \in \mathbb{R}^n$	Response vector

2.3 Sparse high-order interaction model

Since the number of all features D in the high-order interaction model is enormous, we consider a sparse model by using a forward greedy feature selection algorithm called Orthogonal Matching Pursuit (OMP) [5]. Let $S^{(t)} \subseteq [D]$ be the set of indices for selected features at feature selection step $t \geq 1$. In OMP, each feature is selected by

$$S^{(t)} = \{j^{(t)} \cup S^{(t-1)}\}, \quad j^{(t)} = \arg \max_{j \in [D] \setminus S^{(t-1)}} |\boldsymbol{x}_{:j}^\top \boldsymbol{r}^{(t)}|, \tag{3}$$

where $S^{(0)} = \emptyset$ and $\boldsymbol{r}^{(1)} = \boldsymbol{y}$. The (residual) vector $\boldsymbol{r}^{(t)}$ is

$$\boldsymbol{r}^{(t)} = \boldsymbol{y} - X_{S^{(t-1)}} \hat{\boldsymbol{\beta}}_{S^{(t-1)}}, \tag{4}$$

and $\hat{\boldsymbol{\beta}}_{S^{(t-1)}}$ is the least square estimator $\hat{\boldsymbol{\beta}}_{S^{(t-1)}} = X^+_{S^{(t-1)}} \boldsymbol{y}$ where the super script $+$ means pseudo inverse. The feature selection step t is sequentially incremented.

In this paper, we do not care about the normalization of $\boldsymbol{x}_{:j}$ for each $j \in [D]$, because high-order interaction features tend to be very sparse when original features are defined in $[0,1]$. Moreover, we do not care about the centralization of features but it can be considered when the residual vector is centralized as $\mathbf{1}^\top \boldsymbol{r}^{(t)} = 0$ because the inner product will be shift invariant $\boldsymbol{x}_{:j}^\top \boldsymbol{r}^{(t)} = (\boldsymbol{x}_{:j} + c\mathbf{1})^\top \boldsymbol{r}^{(t)}$ for any scalar parameter c.

2.4 Selective inference for sparse linear model

In this section, we review statistical inference after feature selection, which is based on selective inference recently developed by [7]. To formalize the problem, we consider the following response which follows the multivariate normal distribution

$$\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{\mu}, \sigma^2 I_n), \tag{5}$$

where $\boldsymbol{\mu}$ is unknown mean, σ^2 is a known variance, and I_n is n sample identity matrix. We model the expected value of response as $\boldsymbol{\mu} = X_{S^{(t)}}\boldsymbol{\beta}_{S^{(t)}}$, where $X_{S^{(t)}}$ is the fixed design matrix at feature selection step t, and $\boldsymbol{\beta}_{S^{(t)}}$ is the best linear predictor, that is

$$\boldsymbol{\beta}_{S^{(t)}} = \arg\min_{\boldsymbol{\beta}_{S^{(t)}} \in \mathbb{R}^D} \mathbb{E}\|\boldsymbol{y} - X_{S^{(t)}}\boldsymbol{\beta}_{S^{(t)}}\|_2^2 = X_{S^{(t)}}^{+}\boldsymbol{\mu}.$$

Lee et al.,[7] suggested to consider a family of intervals $\{C_{S^{(t)},j}\}_{j \in S^{(t)}}$ that have conditional $(1-\alpha)$ coverage:

$$\Pr(\beta_{S^{(t)},j} \in C_{S^{(t)},j} | S^{(t)}) \geq 1 - \alpha. \tag{6}$$

If (6) holds, then, FCR (false coverage-statement rate) [17] and pFCR (positive false coverage-statement rate) [18] can be controlled (the proof is shown at Lemma 2.1 of [7]) as

$$\mathrm{FCR} = \mathbb{E}\left[\frac{|\{j \in S^{(t)} : \beta_{S^{(t)},j} \notin C_{S^{(t)},j}\}|}{|S^{(t)}|}; |S^{(t)}| > 0\right] \leq \alpha,$$
$$\mathrm{pFCR} = \mathbb{E}\left[\frac{|\{j \in S^{(t)} : \beta_{S^{(t)},j} \notin C_{S^{(t)},j}\}|}{|S^{(t)}|}\middle| |S^{(t)}| > 0\right] \leq \alpha.$$

In this paper, our interest is to construct the conditional intervals shown in (6) for each of the selected high-order interaction features. Our interest is not in evaluating the correctness of the selected model and we do not discuss how to terminate the feature selection step t in OMP. Lee et al.[7] showed that valid confidence intervals can be computed even if the selected model is wrong, but the requirement is the normality of error as shown in (5). On the other hand, the selective inference for determining the feature selection step t is studied in [9], where the interest is to check whether the selected model is correct or not.

2.4.1 Basic idea of selective inference

Selective inference is developed for two stage methods, where a subset of features is *selected* in the first stage, and *inferences* for the selected features are made in the second stage. A key finding by [7] is that, if the first selection stage is described as a *linear selection event*, then exact (non-asymptotic) statistical inference for the fitted coefficients conditional on the selection event can be done.

2.4.2 Feature selection stage

Suppose that, in the first feature selection stage, a subset of features $S^{(t)} = \{j^{(1)}, \cdots, j^{(t)}\} \subseteq [D]$ are selected, where $j^{(t)}$ is defined by (3). The selective inference framework in [7] can be applied to feature selection algorithms whose selection process can be characterized by a set of linear inequalities in the form of $A\boldsymbol{y} \leq \boldsymbol{b}$ with a certain matrix A and a certain vector $\boldsymbol{b}$ that do not depend on $\boldsymbol{y}$. This type of selection event is called a *linear selection event*. In the selective inference framework, inferences are made conditional on the selection event, it means that, in the case of a linear selection event, we only care about the cases where $\boldsymbol{y}$ is observed in a polytope $\mathrm{Pol}(S^{(t)}) = \{\boldsymbol{y} \in \mathbb{R}^n \mid A\boldsymbol{y} \leq \boldsymbol{b}\}$. In [7], [19] and [20], some feature selection procedures are shown to be linear selection events.

As pointed out in [19], [20] the selection process of OMP is a linear selection event, i.e., characterized by a set of linear constraints. Let $S^{(t)}$ and $\bar{S}^{(t)}$ be the set of indices for selected and unselected features, respectively at feature selection step $t \in [k]$ for a fixed natural number k. It indicates that $S^{(1)} \subseteq S^{(2)} \subseteq \cdots \subseteq S^{(k)}$ and $\bar{S}^{(1)} \supseteq \bar{S}^{(2)} \supseteq \cdots \supseteq \bar{S}^{(k)}$. Moreover, let $j^{(t)}$ be the index of t-th selected feature, which is defined by (3). The selection event of OMP is rewritten by the following constraints

$$\forall j' \in \bar{S}^{(t)},\ t \in [k],\ |\boldsymbol{x}_{:j^{(t)}}^\top \boldsymbol{r}^{(t)}| \geq |\boldsymbol{x}_{:j'}^\top \boldsymbol{r}^{(t)}|. \tag{7}$$

These constraints are rephrased by

$$
\begin{aligned}
&\forall j' \in \bar{S}^{(t)},\ t \in [k], \\
&\text{if } \boldsymbol{x}_{:j^{(t)}}^{\top} \boldsymbol{r}^{(t)} > 0, \text{ then} \\
&\qquad\qquad (-\boldsymbol{x}_{:j^{(t)}} - \boldsymbol{x}_{:j'})^{\top} \boldsymbol{r}^{(t)} \leq 0, \\
&\qquad\qquad (-\boldsymbol{x}_{:j^{(t)}} + \boldsymbol{x}_{:j'})^{\top} \boldsymbol{r}^{(t)} \leq 0, \\
&\text{if } \boldsymbol{x}_{:j^{(t)}}^{\top} \boldsymbol{r}^{(t)} \leq 0, \text{ then} \\
&\qquad\qquad (\ \ \boldsymbol{x}_{:j^{(t)}} - \boldsymbol{x}_{:j'})^{\top} \boldsymbol{r}^{(t)} \leq 0, \\
&\qquad\qquad (\ \ \boldsymbol{x}_{:j^{(t)}} + \boldsymbol{x}_{:j'})^{\top} \boldsymbol{r}^{(t)} \leq 0.
\end{aligned}
$$

By using $s_{j^{(t)}} = \mathrm{sign}(\boldsymbol{x}_{:j^{(t)}}^{\top} \boldsymbol{r}^{(t)})$, these constraints are summarized as

$$
\begin{aligned}
&\forall j' \in \bar{S}^{(t)},\ t \in [k], \\
&(-s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}} - \boldsymbol{x}_{:j'})^{\top} \boldsymbol{r}^{(t)} \leq 0, \\
&(-s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}} + \boldsymbol{x}_{:j'})^{\top} \boldsymbol{r}^{(t)} \leq 0, \\
&-s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}}^{\top} \boldsymbol{r}^{(t)} \leq 0,
\end{aligned}
$$

or equivalently

$$
\begin{aligned}
&\forall (j', \xi) \in \bar{S}^{(t)} \times \{-1, 1, 0\},\ t \in [k], \\
&(-s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}} + \xi \boldsymbol{x}_{:j'})^{\top} \boldsymbol{r}^{(t)} \leq 0.
\end{aligned}
$$

Since the residual vector (4) is $\boldsymbol{r}^{(t)} = (I_n - X_{S^{(t-1)}} X_{S^{(t-1)}}^{+}) \boldsymbol{y}$, the constraints are rephrased by the set of linear inequalities in the form of $A\boldsymbol{y} \leq \boldsymbol{b}$ as

$$
\begin{aligned}
&\forall (j', \xi) \in \bar{S}^{(t)} \times \{-1, 1, 0\},\ t \in [k], \\
&(-s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}} + \xi \boldsymbol{x}_{:j'})^{\top} \Gamma^{(t)} \boldsymbol{y} \leq 0,
\end{aligned} \tag{8}
$$

where $\Gamma^{(t)} = (I_n - X_{S^{(t-1)}} X_{S^{(t-1)}}^{+})$ and $\Gamma^{(1)} = I_n$.

2.4.3 Statistical inference stage

In the remainder, we use the notation $S = S^{(k)}$ with a certain fixed natural number k. If we consider the case where S is NOT selected from the data, i.e., independent of $\boldsymbol{y}$, then the

sampling distribution of each fitted coefficient $\hat{\beta}_{S,j} = (X_S^+ \boldsymbol{y})_j$ is

$$\hat{\beta}_{S,j} \sim \mathrm{N}(\beta_{S,j}, \sigma_{S,j}^2), \text{ where } \sigma_{S,j}^2 = \sigma^2 (X_S^\top X_S)_{jj}^{-1}. \tag{9}$$

If we define $\ell_{\alpha/2}^{(j)}$ and $u_{\alpha/2}^{(j)}$ to be the lower and upper $\alpha/2$ points of the sampling distribution in (9), then the *type I error* at level α is controlled as

$$\Pr(\beta_{S,j} \notin [\ell_{\alpha/2}^{(j)}, u_{\alpha/2}^{(j)}]) \leq \alpha. \tag{10}$$

On the other hand, after feature selection, we would like to control the following *selective type I error*

$$\begin{aligned} &\Pr(\beta_{S,j} \notin [\ell_{\alpha/2}^{(S,j)}, u_{\alpha/2}^{(S,j)}] \mid S) \\ =&\Pr(\beta_{S,j} \notin [\ell_{\alpha/2}^{(S,j)}, u_{\alpha/2}^{(S,j)}] \mid \boldsymbol{y} \in \mathrm{Pol}(S)) \leq \alpha, \end{aligned} \tag{11}$$

where the selection event is written as $\boldsymbol{y} \in \mathrm{Pol}(S)$ in the case of a linear selection event. [7] derived how to compute these confidence intervals as formally stated in the following lemma.

Lemma 1. *Let $F_{\mu,\sigma^2}^{[L,U]}$ be the CDF of a truncated normal distribution with the mean μ, the variance σ^2, and the truncation interval $[L, U]$, i.e.,*

$$F_{\mu,\sigma^2}^{[L,U]}(x) = \frac{\Phi((x-\mu)/\sigma) - \Phi((L-\mu)/\sigma)}{\Phi((U-\mu)/\sigma) - \Phi((L-\mu)/\sigma)},$$

where Φ is the CDF of the standard normal distribution. Furthermore, let the truncation points be

$$L(S,j) = \hat{\beta}_{S,j} + \theta_L (X_S^\top X_S)_{jj}^{-1}, \tag{12a}$$

$$\text{where } \theta_L = \min_{\theta \in \mathbb{R}} \theta \text{ s.t. } \boldsymbol{y} + \theta (X_S^+)^\top \boldsymbol{e}_j \in \mathrm{Pol}(S),$$

$$U(S,j) = \hat{\beta}_{S,j} + \theta_U (X_S^\top X_S)_{jj}^{-1}, \tag{12b}$$

$$\text{where } \theta_U = \max_{\theta \in \mathbb{R}} \theta \text{ s.t. } \boldsymbol{y} + \theta (X_S^+)^\top \boldsymbol{e}_j \in \mathrm{Pol}(S),$$

where $\boldsymbol{e}_j$ is an unit vector, j-th element of which is one. If $\ell_{\alpha/2}^{(S,j)}$ and $u_{\alpha/2}^{(S,j)}$ are defined such that

$$F_{\ell_{\alpha/2}^{(S,j)}, \sigma_{S,j}^2}^{[L(S,j),U(S,j)]}(\hat{\beta}_{S,j}) = 1 - \alpha/2, \tag{13a}$$

$$F_{u_{\alpha/2}^{(S,j)}, \sigma_{S,j}^2}^{[L(S,j),U(S,j)]}(\hat{\beta}_{S,j}) = \alpha/2, \tag{13b}$$

then, the interval $[\ell_{\alpha/2}^{(S,j)}, u_{\alpha/2}^{(S,j)}]$ *satisfies the conditional coverage property, i.e.,*

$$\Pr(\beta_{S,j} \notin [\ell_{\alpha/2}^{(S,j)}, u_{\alpha/2}^{(S,j)}] \mid S) = \alpha.$$

The proof of Lemma 1 is presented in § 7.1 although it is easily proved by using the results in [7]. Lemma 1 indicates that the sampling distribution of each fitted coefficient $\hat{\beta}_{S,j} = (X_S^+ \boldsymbol{y})_j$ is a truncated normal distribution when the selection event is a linear selection event $\{A\boldsymbol{y} \leq \boldsymbol{b}\}$. Figure 2 schematically illustrates that truncation points can be computed by optimizing θ along with the direction $\boldsymbol{\eta} = (X_S^+)^\top \boldsymbol{e}_j$ since each fitted coefficient $\hat{\beta}_{S,j}$ is $\boldsymbol{\eta}^\top \boldsymbol{y}$.

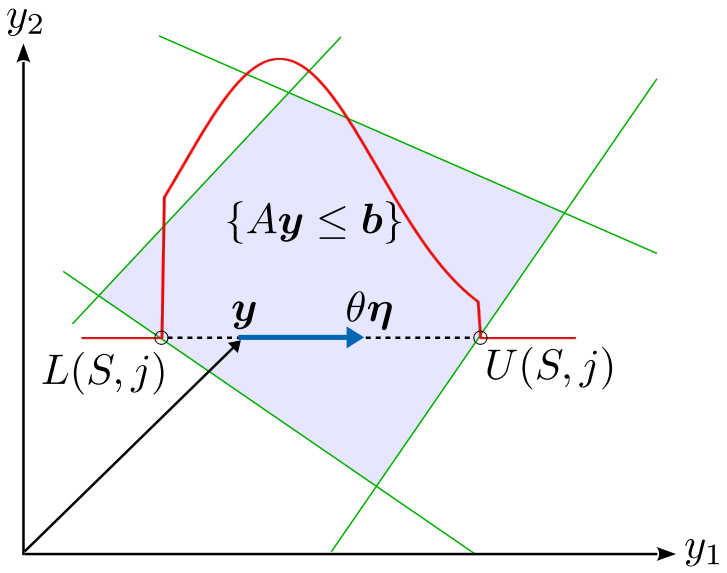
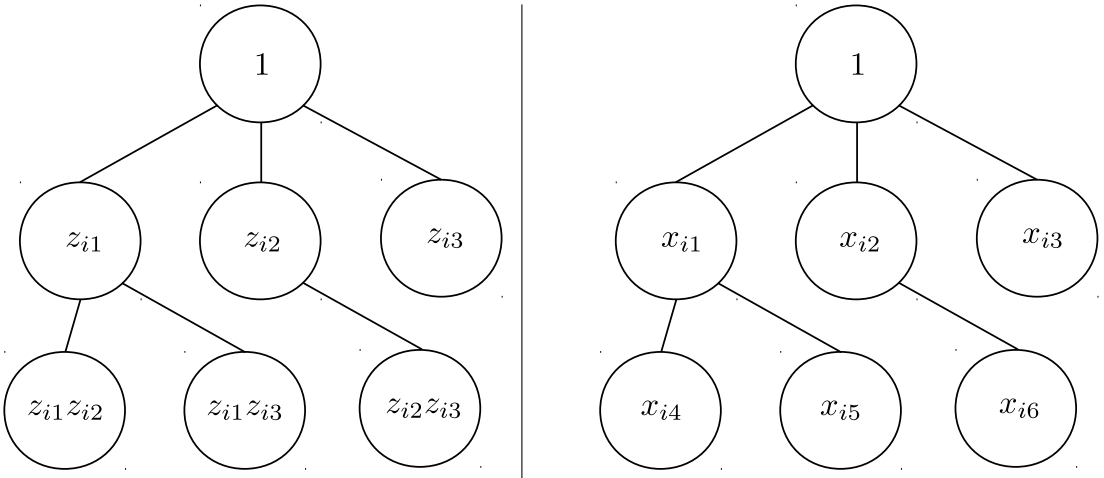


Figure 2: An illustration of polyhedral lemma. The polyhedron represents the selection event and truncation points can be computed by optimizing θ along with the direction $\boldsymbol{\eta} = (X_S^+)^\top \boldsymbol{e}_j$.

Unfortunately, we cannot directly apply this selective inference framework to the high-order interaction model because the polytope $\mathrm{Pol}(S)$ is characterized by extremely large number of linear inequalities, and the optimization problems in (12) are hard to solve.

3 Computational tricks by using tree-based feature representation

The OMP feature selection (3) and the computation of truncation points (12) are computationally infeasible because the number of high-order interaction features D is extremely huge. For circumventing this issue, we introduce a tree-based representation of high-order interaction features shown in Figure 3 and consider the following computational pruning techniques in both feature selection and statistical inference stages.



(a) Tree-based feature representation | (b) Another tree-based feature representation

Figure 3: (a) An example of high-order interaction features represented by a tree structure with a fixed sample index i, where the number of original features $d = 3$ and the maximum interaction order $r = 2$. In the tree, each node can be efficiently computed by using a tree (or graph) search technique such as DFS (depth first search). This tree originally has a lattice structure, e.g., the node $z_{i1}z_{i2}$ has two edges from z_{i1} and z_{i2}, but we remove the edges for simplicity. (b) Another tree-based feature representation by using high-order interaction features.

3.1 Pruning technique in feature selection stage

Since a high-order interaction feature is the product of multiple original features whose value are defined in $[0,1] \in \mathbb{R}$, the following property holds:

$$0 \leq x_{i\tilde{j}} \leq x_{ij}, \forall (i,\tilde{j}) \in [n] \times Des(j), \tag{14}$$

where $Des(j)$ is the set of indicates for descendant features of j-th parent feature. By using this, for any residual vector $\boldsymbol{r}^{(t)}$, we obtain

$$|\boldsymbol{x}_{:\tilde{j}}^{\top}\boldsymbol{r}^{(t)}| \;\leq\; \max \left\{ \sum_{i:r_i^{(t)}>0} x_{ij} r_i^{(t)}, - \sum_{i:r_i^{(t)}<0} x_{ij} r_i^{(t)} \right\}. \tag{15}$$

When we search over the tree, if the right-hand side of (15) is smaller than the current largest correlation in (3), then, we can quit searching over its descendant nodes $\tilde{j} \in Des(j)$. This pruning technique has been widely used in item-set or sub-graph mining, e.g.,[21, 22, 23, 24].

3.2 Proposed pruning technique in statistical inference stage

In the following lemma, we show the solutions of the optimization problems (12).

Lemma 2. *For $\boldsymbol{\eta} = (X_S^{+})^{\top}\boldsymbol{e}_j$, the solutions of the optimization problems in (12) are respectively written as*

$$\theta_L = \max_{t\in[k]} \max_{\substack{(j',\xi)\in \bar{S}^{(t)}\times\{-1,1,0\},\\ (\xi \boldsymbol{x}_{:j'} - s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}})^{\top}\Gamma^{(t)}\boldsymbol{\eta}<0}} \frac{(s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}} - \xi \boldsymbol{x}_{:j'})^{\top}\Gamma^{(t)}\boldsymbol{y}}{(\xi \boldsymbol{x}_{:j'} - s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}})^{\top}\Gamma^{(t)}\boldsymbol{\eta}} \leq 0, \tag{16a}$$

$$\theta_U = \min_{t\in[k]} \min_{\substack{(j',\xi)\in \bar{S}^{(t)}\times\{-1,1,0\},\\ (\xi \boldsymbol{x}_{:j'} - s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}})^{\top}\Gamma^{(t)}\boldsymbol{\eta}>0}} \frac{(s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}} - \xi \boldsymbol{x}_{:j'})^{\top}\Gamma^{(t)}\boldsymbol{y}}{(\xi \boldsymbol{x}_{:j'} - s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}})^{\top}\Gamma^{(t)}\boldsymbol{\eta}} \geq 0. \tag{16b}$$

where $\bar{S}^{(t)}, s_{j^{(t)}}$ and $\Gamma^{(t)}$ are appeared in (8).

The proof of Lemma 2 is presented in § 7.2. Since the inner maximization and inner minimization of (16) are difficult to compute, we present an efficient computational trick for those inner optimization problems, which is our main contribution. Our basic idea for addressing the computational difficulty is to note that most of the inequalities actually do

not affect the results of the selective inference, and a large portion of them can be identified by exploiting the anti-monotonicity properties defined in the tree structure.

We consider a tree which consists of a set of nodes corresponding to each of the unselected features $j' \in \bar{S}^{(t)}$. We define $Des(j')$ be the set of descendant nodes of j'-th node in the tree. When we search over the tree, we introduce a novel pruning strategy by deriving a condition such that, if the j'-th node in the tree satisfies certain conditions, then all the $\tilde{j}' \in Des(j')$ are guaranteed to be irrelevant to the selective inference results because they do not affect the optimal solutions in (16).

Theorem 3. *Consider solving the inner maximization problem in (16a), and let $\hat{\theta}_L$ be the current optimal solution, i.e., we know that the optimal θ_L is at least no greater than $\hat{\theta}_L$. For any $j' \in \bar{S}^{(t)}$ and all $\xi \in \{-1, 1, 0\}$, if either of the following conditions*

$$\sum_{i:\xi\left(\Gamma^{(t)}\boldsymbol{\eta}\right)_i<0} x_{ij'}\xi\left(\Gamma^{(t)}\boldsymbol{\eta}\right)_i - s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}}^\top\Gamma^{(t)}\boldsymbol{\eta} \geq 0, \ \textit{or}$$
$$\frac{\max\{0, s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}}^\top\Gamma^{(t)}\boldsymbol{y} - \sum_{i:\xi\left(\Gamma^{(t)}\boldsymbol{y}\right)_i>0} x_{ij'}\xi\left(\Gamma^{(t)}\boldsymbol{y}\right)_i\}}{\min\{0, \sum_{i:\xi\left(\Gamma^{(t)}\boldsymbol{\eta}\right)_i<0} x_{ij'}\xi\left(\Gamma^{(t)}\boldsymbol{\eta}\right)_i - s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}}^\top\Gamma^{(t)}\boldsymbol{\eta}\}} \leq \hat{\theta}_L \tag{17}$$

are satisfied, then its descendant nodes $\tilde{j}' \in Des(j')$ do not affect the solution of (16a).

Similarly, for any $j' \in \bar{S}^{(t)}$ and all $\xi \in \{-1, 1, 0\}$, if either of the following conditions

$$\sum_{i:\xi\left(\Gamma^{(t)}\boldsymbol{\eta}\right)_i>0} x_{ij'}\xi\left(\Gamma^{(t)}\boldsymbol{\eta}\right)_i - s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}}^\top\Gamma^{(t)}\boldsymbol{\eta} \leq 0, \ \textit{or}$$
$$\frac{\max\{0, s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}}^\top\Gamma^{(t)}\boldsymbol{y} - \sum_{i:\xi\left(\Gamma^{(t)}\boldsymbol{y}\right)_i>0} x_{ij'}\xi\left(\Gamma^{(t)}\boldsymbol{y}\right)_i\}}{\max\{0, \sum_{i:\xi\left(\Gamma^{(t)}\boldsymbol{\eta}\right)_i>0} x_{ij'}\xi\left(\Gamma^{(t)}\boldsymbol{\eta}\right)_i - s_{j^{(t)}}\boldsymbol{x}_{:j^{(t)}}^\top\Gamma^{(t)}\boldsymbol{\eta}\}} \geq \hat{\theta}_U \tag{18}$$

are satisfied, then its descendant nodes $\tilde{j}' \in Des(j')$ do not affect the solution of (16b).

The proof of Theorem 3 is presented in § 7.3. Figure 4 shows that all the conditions in Theorem 3 can be checked at the j'-th node in each tree, and if the conditions are satisfied as the j'-th node, then one can skip searching over its subtree. It allows us to perform selective inference even if the number of constraints that defines the selection event is extremely large. A brief statement of selective inference with our proposed pruning technique is shown in Algorithm 1.

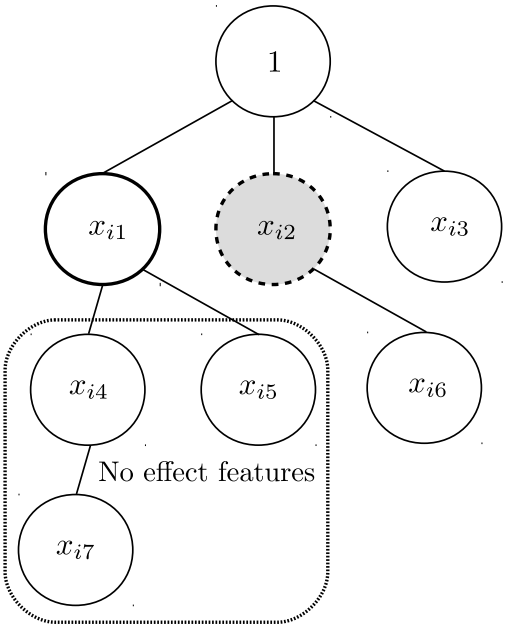


Figure 4: An example of our pruning technique behavior. In the tree, we consider the situation that the feature x_{i2} is selected at step $t = 1$ and the sets of indices are $S^{(1)} = \{2\}$, $\bar{S}^{(1)} = \{1, 3, 4, 5, 6, 7\}$. Then, we consider the tree corresponding to the unselected features $j' \in \bar{S}^{(1)}$. In this example, we update the current optimal solutions $\hat{\theta}_L$ and $\hat{\theta}_U$ by using the feature $x_{ij'}$ with the $j' = 1$. Moreover, the pruning conditions are satisfied at the $j' = 1$, then one can skip searching over its subtree which has no effect on the selective inference.

4 Extensions

In this section, we improve computational efficiency and statistical power of selective inference. The drawback of selective inference is that one has to consider the number of considerable constraints as shown in (8). This is not desirable in terms of both computational efficiency and statistical power – according to [20], "a greater degree of conditioning will generally lead to less powerful tests and wider intervals," and this fact is also studied in [25]. Moreover, the statistical power of selective inference is generally less powerful when the number of selected features is increased, because the degree of conditioning is linearly increased with the number of selected features.

In order to circumvent such a over conditioning, we consider the following confidence

Algorithm 1 Selective inference with our proposed pruning technique.

Input: Dataset $(\boldsymbol{Z}, \boldsymbol{y})$, the number of selected features k, the maximum interaction order r, desired significance level α.

Output: The set of selected features $S^{(k)}$, and confidence intervals for the selected features.

1: Consider the rth-order interaction features which are represented by the tree in Figure 3.

2: **for** $t = 1, \cdots, k$ **do**

3: Select a feature and update the current set $S^{(t)}$ by using the pruning condition (15) with OMP in (3).

4: **end for**

5: **for** $j = 1, \cdots, k$ **do**

6: For $\boldsymbol{\eta} = (X^{+}_{S^{(k)}})^{\top} \boldsymbol{e}_j$, compute θ_L, θ_U in (16) by using our proposed pruning conditions (17) and (18).

7: Compute the confidence interval $[\ell^{(S^{(k)},j)}_{\alpha/2}, u^{(S^{(k)}S,j)}_{\alpha/2}]$ which satisfies equation (13).

8: **end for**

interval instead of (6): for any $t \in [k]$ with a fixed natural number k,

$$\Pr(\beta_{S^{(t)},t} \in C_{S^{(t)},t} | S^{(t)}) \geq 1 - \alpha. \tag{19}$$

In this inequality, we consider incrementally selected k model, and we only interested in the coefficient of t-th selected feature in the t-th model. The advantage of this approach is that the coefficient $\beta_{S^{(t)},t}$ is independent of $t+1, \cdots, k$ selected features. In other words, we do NOT need to consider the selection events that $t+1, \cdots, k$ features are selected, and then the degree of conditioning is strictly smaller than fully conditioned (8). Although a hypothesis testing for incrementally selected model is introduced in [9, 26] and the null hypothesis is called *incremental null*, our contribution is to utilize (19) for reducing the degree of conditioning.

Simulation study for FCR control Here, we checked whether selective inference can control FCR or not when we consider the confidence interval (19). We generated the following

synthetic response as $y_i = \mu_i + \epsilon_i, \mu_i = -z_{i1} + 2z_{i2} - 1.5z_{i3},\ \epsilon_i \sim N(0, 0.5^2)$, and the element of design matrix $z_{ij} \in \{0, 1\}$ was randomly generated by

$$z_{ij} = \begin{cases} 0 & \text{if } u \leq \zeta, \\ 1 & \text{otherwise,} \end{cases} \tag{20}$$

where u was a random variable generated by the uniform distribution Unif$(0, 1)$, and ζ is the sparsity of design matrix. We fixed the number of original features d and the sparsity ζ as 100, 95%, respectively. FCR was estimated by using v/k where v is the number of false rejections and k is the number of selected features, e.g., if two features z_{i1}, z_{i5} are selected with $k = 2$ and each of the intervals does not cover zero, then the number of false rejections v is one (because z_{i5} is not in the true model) and FCR $= 1/2$. We randomly generated the synthetic data over 2000 times and averaged FCR over those independent simulations. Figure 5 showed FCRs for various parameters with significance level was 0.05. FCR could be controlled for any the number of selected features $k \in \{1, \cdots, 10\}$ even if the number of features in the true model was three. In particular, FCR in the case of greater n and smaller k was nearly zero since the selected features were in the true model in most cases and false rejections were almost none.

5 Experiments

5.1 Experiments on synthetic data

We compared our selective inference approach with non-adjusted z-testing using (9) and data-splitting approach [8] on synthetic data. In data-splitting, we splitted the data into two subsets, and used one for feature selection and another for z-testing. The performance of data-splitting is basically weak both in selection and inference stages because only a part of the available data is used in each stage. We wrote the short description of each approach in Table 2.

First, we generated the synthetic data whose response was $y_i = \epsilon_i,\ \epsilon_i \sim N(0, \sigma^2)$, this means that all coefficients are 0. The element of original design matrix Z was randomly generated by using (20). Since there were several hyper-parameters (σ, ζ, etc.), we used

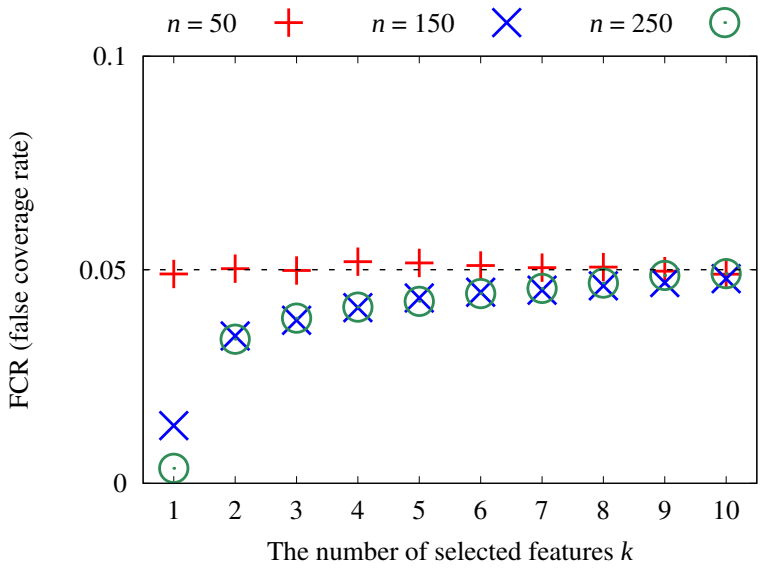


Figure 5: FCRs for various parameters: the number of samples n and selected features k. We fixed the number of original features d and sparsity of design matrix ζ as 100, 95%, respectively.

Table 2: The short description of approaches

Symbol	Description
`SI`	Our selective inference approach.
$\texttt{SI}_{\texttt{inc}}$	`SI` with the incrementally constructed interval (19).
`OLS`	z-testing using ordinary least squares in (9).
`split`	Data-splitting approach.

the default settings of parameters as shown in Table 3, but those parameters were changed depending on the experiment.

FCR controlling Here, we checked whether FCR can be properly controlled or not. FCR was estimated by using v/k where v is the number of false rejections and k is the number of selected features. We randomly generated the synthetic data over 2000 times and averaged FCR over those independent simulations. Figure 6 showed that our approaches and `split` could control FCRs for any parameters under the desired significance level, while `OLS` could not control them. (a) The selection bias was more intense if the number of samples n

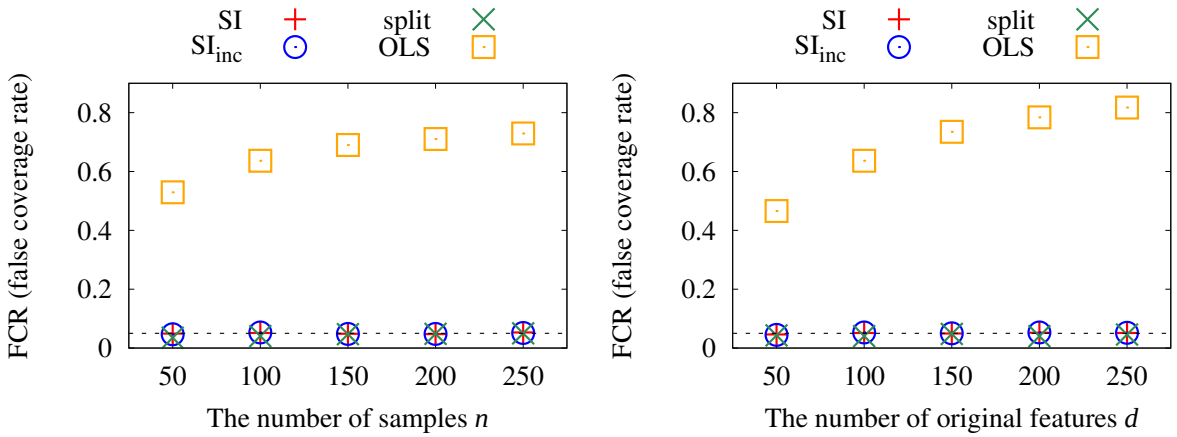
Table 3: Default setting of parameters

Parameter	Value	Description
n	100	The number of samples.
d	100	The number of original features.
k	5	The number of selected features.
r	3	The maximum order of interactions.
σ	0.5	Noise level for response.
ζ	95%	Sparsity of original design matrix.
α	5%	Desired significance level.

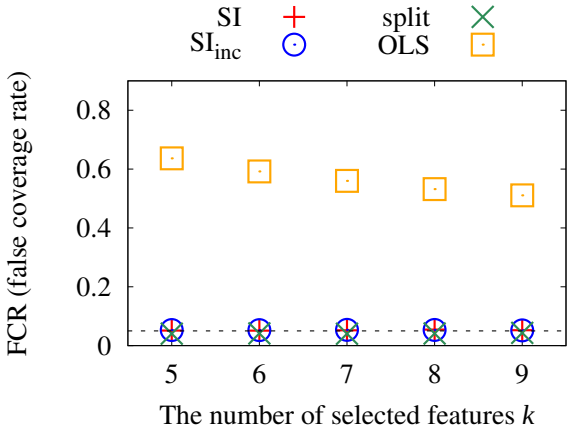
is increased. We conjecture that the non-adjusted confidence intervals might not cover 0 because a greater n will leads tighter confidence interval. (b) The selection bias critically depends on the number of original features. (c) The selection bias was smaller if the number of selected features k is increased. We conjecture that redundant features make the absolute value of fitted coefficients be smaller, and then the confidence intervals tend to cover 0 if k is excessively increased.

Statistical power comparison Second, we evaluated the statistical power of the inference. We replaced the response as $y_i = -z_{i1} + 2z_{i2}z_{i3} - 3z_{i4}z_{i5}z_{i6} + \epsilon_i,\ \epsilon \sim N(0, \sigma^2)$, and we set the sparsity ζ of the original vectors $\boldsymbol{z}_{:1}, \cdots, \boldsymbol{z}_{:6}$ to 80%. We computed TPR (true positive rate) of the approaches. We defined TPR = TP/3 where TP is the number of true positives, e.g., TP is 3 if the confidence intervals of coefficients corresponding to $z_{i1}, z_{i2}z_{i3}$ and $z_{i4}z_{i5}z_{i6}$ do not cover 0. We randomly generated the synthetic data over 2000 times and averaged TPRs over those independent simulations. Figure 7 showed that $\texttt{SI}_{\texttt{inc}}$ quite outperformed data-splitting, while the power of `SI` decreases as the number of selected features increases. This indicates that the degree of conditioning critically depends on k in `SI`, which leads to wider intervals.

Computational efficiency and sensitivity Lastly, we evaluated the computational efficiency and sensitivity of our approaches. We used the same synthetic data in the previous



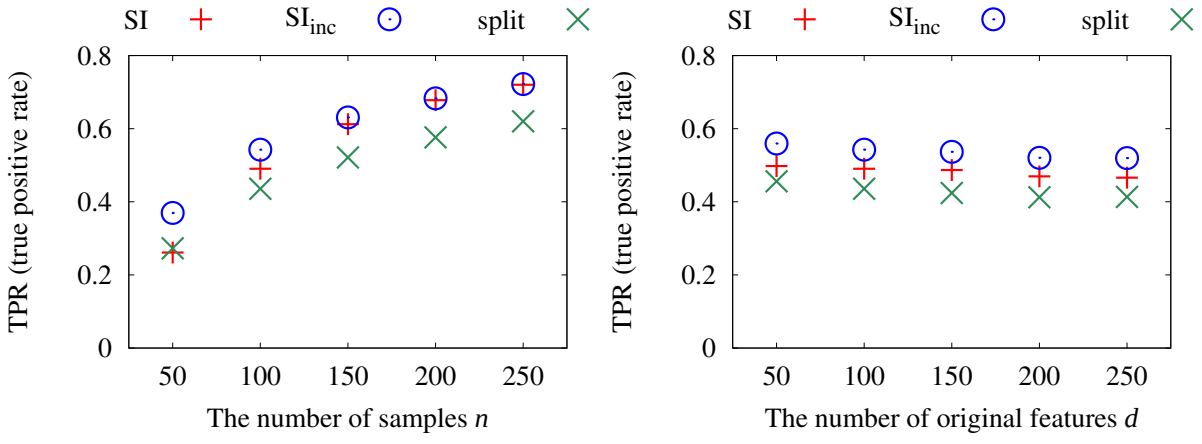
(a) $n \in \{50, 100, 150, 200, 250\}$ (b) $d \in \{50, 100, 150, 200, 250\}$



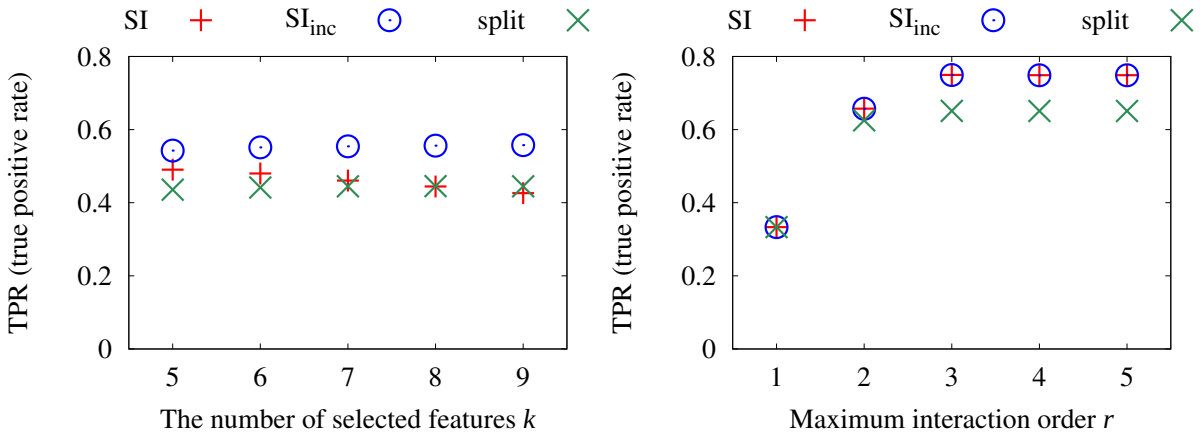
(c) $k \in \{5, 6, 7, 8, 9\}$

Figure 6: FCRs for various parameters. Our approaches and `split` could control FCRs under the desired significance level, while OLS could not control them.

paragraph (statistical power comparison), but we changed the number of original features d and the sparsity of original design matrix ζ. We randomly generated the synthetic data over 100 times and averaged computational time over those independent simulations. The machine spec was Intel(R) Core(TM) i7-3517U CPU 1.90GHz with 4GByte memory, and we implemented our approach as a single thread application. Figure 8 showed that the computational cost strongly depends on three parameters: (i)the number of original features d, (ii)the sparsity ζ, and (iii)the maximum interaction order r when d is large and ζ is lower. `SI` and $\texttt{SI}_{\texttt{inc}}$ were almost the same if the number of selected features k is small.



(a) $n \in \{50, 100, 150, 200, 250\}$ (b) $d \in \{50, 100, 150, 200, 250\}$



(c) $k \in \{5, 6, 7, 8, 9\}$ (d) $r \in \{1, 2, 3, 4, 5\}$ and $n = 300$

Figure 7: TPRs for various parameters. (a) and (b) $\texttt{SI}_{\texttt{inc}}$ quite outperformed data-splitting. (c) the power of `SI` decreases as the number of selected features increases. (d) FPRs did not depend on the maximum interaction order r if $r \geq 4$; however, we observed that FPRs were slightly decreased depending on r when the number of samples n was very small.

5.2 Application to HIV drug resistance data

We applied our selective inference approaches to HIV-1 sequence data obtained from Stanford HIV Drug Resistance Database [27]. The goal here was to find statistically significant high-order interaction features that are highly associated with the drug resistances. The detail of the features is that, each feature means a mutation that is defined as amino acid differences from HIV-1 subtype B consensus sequence. If a position of the sequence is mutated, then the feature corresponding to the position is 1 otherwise 0. We used three datasets for an-

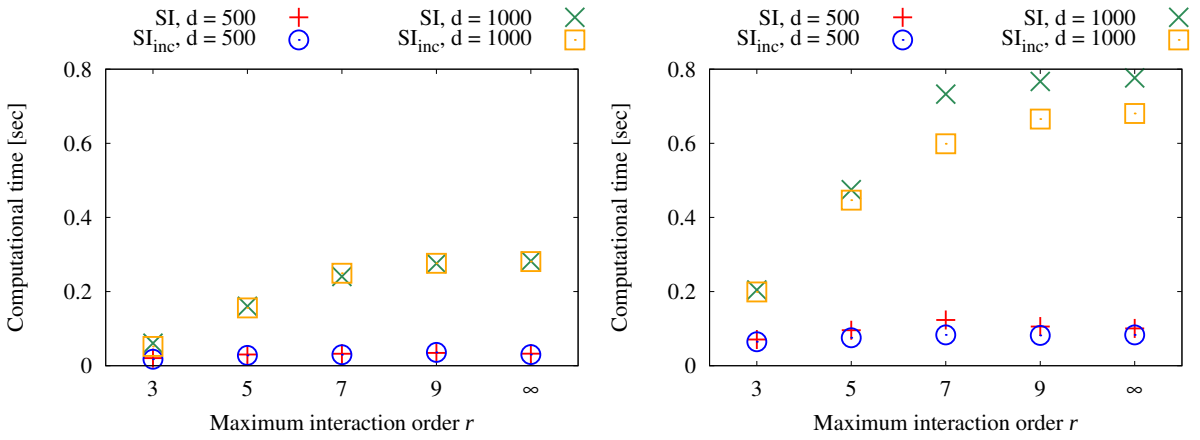
tiretroviral drugs: Delavirdine(DLV), Tenofovir(TDF), and Atazanavir(ATV). We converted the response (drug resistance) y_i as $y_i \leftarrow \log(y_i)$ since the responses were heavy tailed. We estimated the variance parameter σ^2 by using $\hat{\sigma}^2 = \frac{1}{n-1}\sum_i^n (y_i - \bar{y})^2$ where $\bar{y}$ is the sample mean of responses. We assumed that the estimated variance $\hat{\sigma}^2$ is greater than the true variance σ^2, and this leads to wider intervals but type I error is still controlled. We selected $k = 30$ features with the maximum interaction order $r = 5$, and evaluated the confidence intervals for those features with Bonferroni-adjusted significance level $\alpha = 0.05/k$.

Figure 9 showed the confidence intervals of fitted coefficients for selected features. These results indicate that our approaches could successfully identify statistically significant high-order interaction features. Our proposed selective inference approaches are applicable to genome-wide association studies (GWASs). In GWASs, evaluating combinatorial effects of mutations (or SNPs: single nucleotide polymorphisms) is important because a large proportion of heritability remains unexplained by single effect [1, 2, 3]. Although several existing pattern mining algorithms [21, 22, 23] can detect interactions of features which are associated with the response, these can NOT control false positive findings, which means that adaptation of the mining results to real world application would be risky. Our approaches can detects them under FCR controlling even if the number of possible combinations of features is extremely large. Table 4 showed that the computational times of our approaches are feasible and $\texttt{SI}_{\texttt{inc}}$ is strictly faster than `SI`. Lastly, we applied data-splitting approach

Table 4: Computational times [s] in HIV datasets

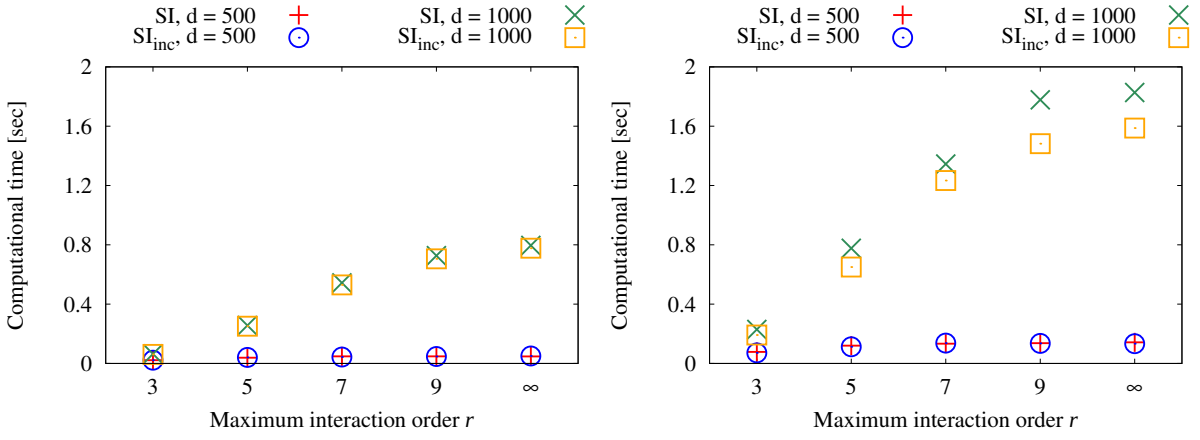
Dataset	`SI`	$\texttt{SI}_{\texttt{inc}}$
`DLV` $(n = 732, d = 371)$	54.7	28.5
`TDF` $(n = 353, d = 348)$	36.4	15.6
`ATV` $(n = 329, d = 225)$	16.8	8.1

to `ATV` dataset in different splitting. Figure 10 showed that both confidence interval and set of selected features are affected by the randomness of splitting, that is a drawback of data-splitting approach.



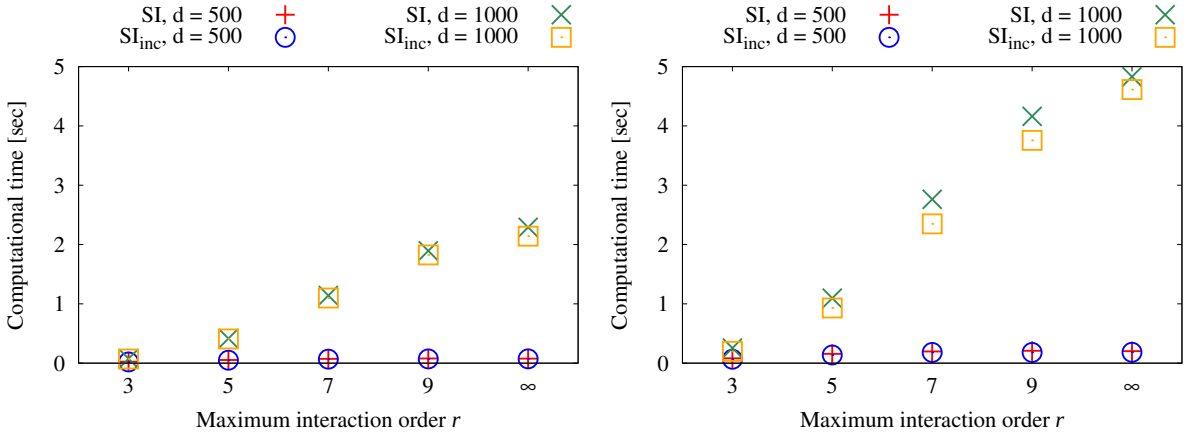
(a) $k = 5, \zeta = 99.6\%$

(b) $k = 10, \zeta = 99.6\%$



(c) $k = 5, \zeta = 99.5\%$

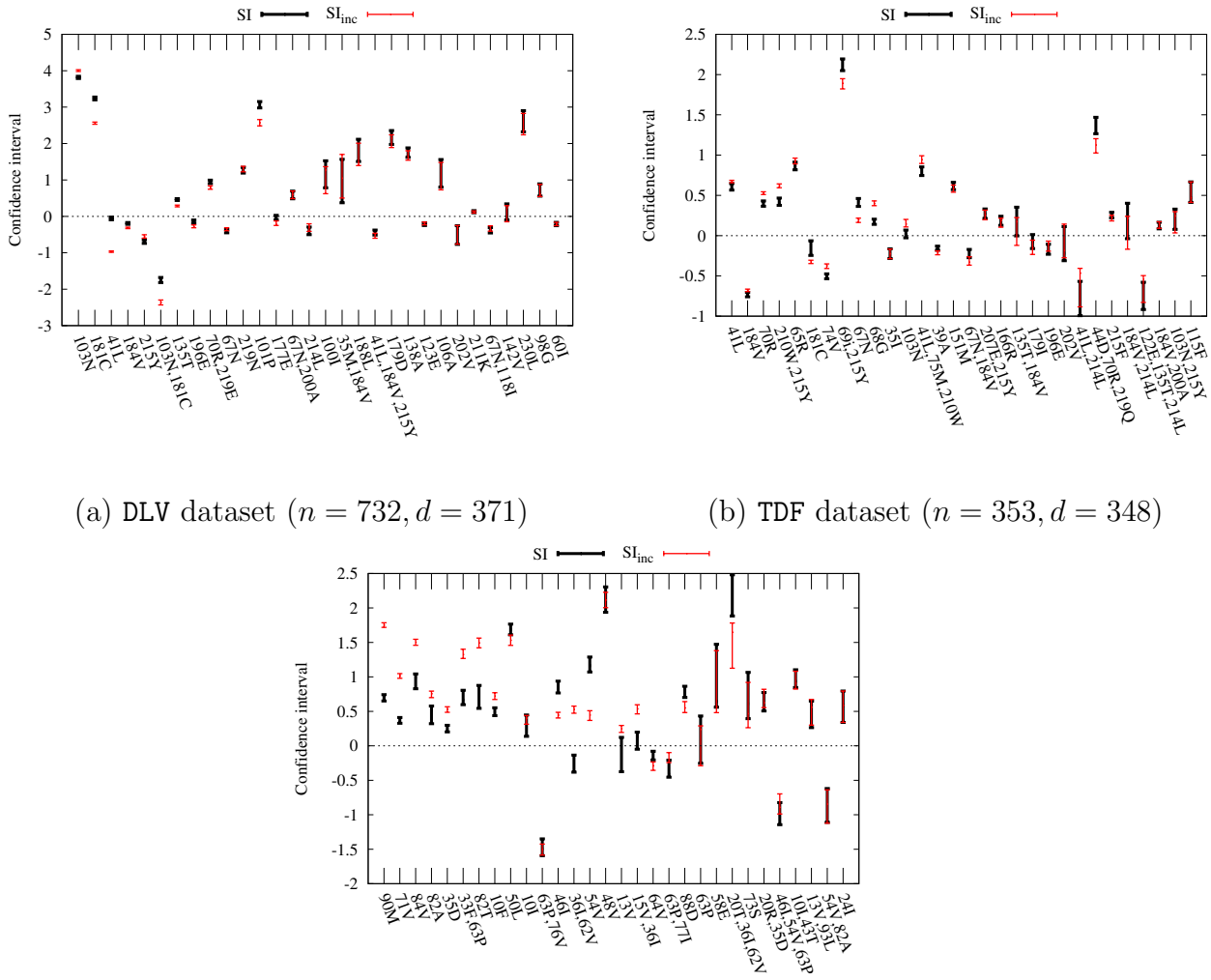
(d) $k = 10, \zeta = 99.5\%$



(e) $k = 5, \zeta = 99.4\%$

(f) $k = 10, \zeta = 99.4\%$

Figure 8: Computational times for various parameters: the number of original features $d \in \{500, 1000\}$, the maximum interaction order $r \in \{3, 5, 7, 9, \infty\}$, the sparsity of original design matrix $\zeta \in \{99.6\%, 99.5\%, 99.4\%\}$, the number of selected features $k \in \{5, 10\}$.

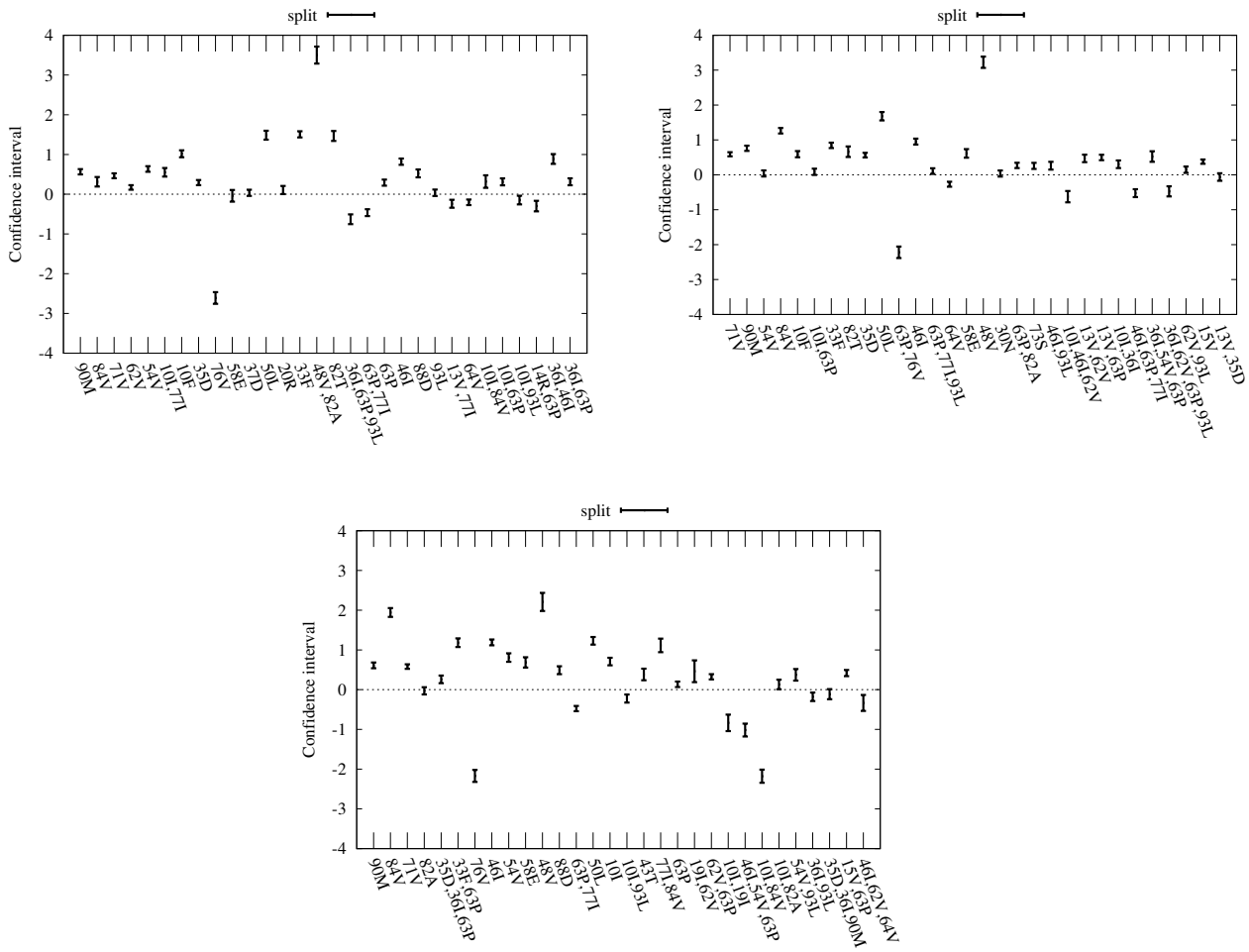


(a) `DLV` dataset ($n = 732, d = 371$)

(b) `TDF` dataset ($n = 353, d = 348$)

(c) `ATV` dataset ($n = 329, d = 225$)

Figure 9: The confidence intervals of fitted coefficients for $k = 30$ selected features with Bonferroni-adjusted significance level $\alpha = 0.05/k$.



ATV dataset ($n = 329, d = 225$)

Figure 10: The confidence intervals using data-splitting approach where data is splitted differently. Each label of horizontal axis is sorted in the order of OMP selection steps: the left side corresponds to step = 1, and the right side corresponds to step = 30. It is quite annoying that both confidence interval and set of selected features are affected by the randomness of splitting.

6 Conclusion

In this paper, we extended selective inference framework to a high-order interaction model by introducing a novel computational trick for computing the sampling distribution of test statistic. We demonstrated that our approaches are computationally useful, the statistical power of inference is better than data-splitting approach, and it allows us to address selection bias issue.

7 Proofs

7.1 Proof of Lamma 1

Proof. To prove the lemma, we first state the polyhedral lemma in [7] that is slightly general case of $\mathrm{V}[\boldsymbol{y}] = \Sigma$ as follows:

Lemma 4 (Polyhedral Lemma; [7])**.** *Suppose* $\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{\mu}, \Sigma)$*. Let* $\boldsymbol{c} = \Sigma\boldsymbol{\eta}(\boldsymbol{\eta}^\top\Sigma\boldsymbol{\eta})^{-1}$ *for any* $\boldsymbol{\eta} \in \mathbb{R}^n$*, and let* $\boldsymbol{z} = (I_n - \boldsymbol{c}\boldsymbol{\eta}^\top)\boldsymbol{y}$*. Then we have*

$$\begin{aligned}\mathrm{Pol}(S) &= \{\boldsymbol{y} \in \mathbb{R}^n \mid A\boldsymbol{y} \leq \boldsymbol{b}\} \\ &= \left\{ \boldsymbol{y} \in \mathbb{R}^n \;\middle|\; \begin{array}{l} L(S,\boldsymbol{z}) \leq \boldsymbol{\eta}^\top\boldsymbol{y} \leq U(S,\boldsymbol{z}), \\ N(S,\boldsymbol{z}) \geq 0 \end{array} \right\},\end{aligned}$$

where

$$L(S,\boldsymbol{z}) = \max_{j:(A\boldsymbol{c})_j<0} \frac{b_j - (A\boldsymbol{z})_j}{(A\boldsymbol{c})_j}, \tag{21a}$$

$$U(S,\boldsymbol{z}) = \min_{j:(A\boldsymbol{c})_j>0} \frac{b_j - (A\boldsymbol{z})_j}{(A\boldsymbol{c})_j}, \tag{21b}$$

$$N(S,\boldsymbol{z}) = \max_{j:(A\boldsymbol{c})_j=0} b_j - (A\boldsymbol{z})_j. \tag{21c}$$

In addition, $(L(S,\boldsymbol{z}), U(S,\boldsymbol{z}), N(S,\boldsymbol{z}))$ *is independent of* $\boldsymbol{\eta}^\top\boldsymbol{y}$*.*

The polyhedral lemma allows us to construct a pivotal quantity as a truncated normal distribution, that is

$$[F^{[L(S,\boldsymbol{z}),U(S,\boldsymbol{z})]}_{\boldsymbol{\eta}^\top\boldsymbol{\mu},\boldsymbol{\eta}^\top\Sigma\boldsymbol{\eta}}(\boldsymbol{\eta}^\top\boldsymbol{y})|\boldsymbol{y} \in \mathrm{Pol}(S)] \sim \mathrm{Unif}(0,1),$$

where $\mathrm{Unif}(0,1)$ denotes the standard (continuous) uniform distribution. For concreteness, $\boldsymbol{\eta}^\top \boldsymbol{y}$ is the least square estimator $\hat{\beta}_{S,j}$ when $\boldsymbol{\eta} = (X_S^+)^\top \boldsymbol{e}_j$. By using this, the conditional $(1-\alpha)$ interval $[\ell_{\alpha/2}^{(S,\boldsymbol{z})}, u_{\alpha/2}^{(S,\boldsymbol{z})}]$ are defined by

$$F^{[L(S,\boldsymbol{z}),U(S,\boldsymbol{z})]}_{\ell_{\alpha/2}^{(S,\boldsymbol{z})},\boldsymbol{\eta}^\top\Sigma\boldsymbol{\eta}}(\boldsymbol{\eta}^\top\boldsymbol{y}) = 1-\alpha/2,$$
$$F^{[L(S,\boldsymbol{z}),U(S,\boldsymbol{z})]}_{u_{\alpha/2}^{(S,\boldsymbol{z})},\boldsymbol{\eta}^\top\Sigma\boldsymbol{\eta}}(\boldsymbol{\eta}^\top\boldsymbol{y}) = \alpha/2.$$

Since $\boldsymbol{z}$ is independent of $\boldsymbol{\eta}^\top\boldsymbol{y}$ (in fact, the direction of two vectors is orthogonal as $\boldsymbol{\eta}^\top\boldsymbol{z}=0$), we abbreviate $\boldsymbol{z}$ that can be integrated out.

The remaining is to show that truncation points in (21) are equivalent to

$$L(S) = \boldsymbol{\eta}^\top\boldsymbol{y} + \theta_L \boldsymbol{\eta}^\top\Sigma\boldsymbol{\eta} \tag{22a}$$
$$\text{where} \quad \theta_L = \min_{\theta\in\mathbb{R}} \theta \quad \text{s.t.} \quad \boldsymbol{y} + \theta\Sigma\boldsymbol{\eta} \in \mathrm{Pol}(S)$$

and

$$U(S) = \boldsymbol{\eta}^\top\boldsymbol{y} + \theta_U \boldsymbol{\eta}^\top\Sigma\boldsymbol{\eta} \tag{22b}$$
$$\text{where} \quad \theta_U = \max_{\theta\in\mathbb{R}} \theta \quad \text{s.t.} \quad \boldsymbol{y} + \theta\Sigma\boldsymbol{\eta} \in \mathrm{Pol}(S),$$

respectively. Simple calculation shows that, for any $\theta \in \mathbb{R}$, we have

$$\begin{aligned}
&\boldsymbol{y} + \theta\Sigma\boldsymbol{\eta} \in \mathrm{Pol}(S) \\
&\Leftrightarrow A(\boldsymbol{y} + \theta\Sigma\eta) \le \boldsymbol{b} \\
&\Leftrightarrow \theta \cdot A\Sigma\eta \le \boldsymbol{b} - A\boldsymbol{y}. \\
&\Leftrightarrow \begin{cases} \theta \le (\boldsymbol{b}-A\boldsymbol{y})_j/(A\Sigma\boldsymbol{\eta})_j, & (A\Sigma\boldsymbol{\eta})_j > 0 \\ \theta \ge (\boldsymbol{b}-A\boldsymbol{y})_j/(A\Sigma\boldsymbol{\eta})_j, & (A\Sigma\boldsymbol{\eta})_j < 0 \\ 0 \le (\boldsymbol{b}-A\boldsymbol{y})_j, & (A\Sigma\boldsymbol{\eta})_j = 0 \end{cases}.
\end{aligned}$$

On the other hand, by the definition of $\boldsymbol{c}$ and $\boldsymbol{z}$ in Lemma 4, it is easy to see that

$$L(S) = \boldsymbol{\eta}^\top\boldsymbol{y} + \boldsymbol{\eta}^\top\Sigma\boldsymbol{\eta} \max_{j:(A\Sigma\boldsymbol{\eta})_j<0} \frac{(\boldsymbol{b}-A\boldsymbol{y})_j}{(A\Sigma\boldsymbol{\eta})_j}.$$

Therefore, for each j such that $(A\Sigma\boldsymbol{\eta})_j < 0$, we have

$$\max_{j:(A\Sigma\boldsymbol{\eta})_j<0} \frac{(\boldsymbol{b}-A\boldsymbol{y})_j}{(A\Sigma\boldsymbol{\eta})_j} \le \theta$$

and thus the minimum possible feasible θ would be

$$\begin{aligned}\theta_L &= \min\{\theta \in \mathbb{R} \mid \boldsymbol{y} + \theta\Sigma\eta \in \mathrm{Pol}(S)\} \\ &= \max_{j:(A\Sigma\boldsymbol{\eta})_j<0} \frac{(\boldsymbol{b} - A\boldsymbol{y})_j}{(A\Sigma\boldsymbol{\eta})_j}.\end{aligned}$$

Similarly, we see that the equivalency of $U(S)$.

To complete the proof, let us consider the covariance matrix $\sigma^2 I_n$ and choose $\boldsymbol{\eta} = (X_S^+)^\top \boldsymbol{e}_j$. In this case, (22) can be written as

$$\begin{aligned}&L(S,j) = \hat{\beta}_{S,j} + \theta_L \sigma^2 (X_S^\top X_S)^{-1}_{jj} \\ &\text{where} \quad \theta_L = \min_{\theta\in\mathbb{R}} \theta \quad \text{s.t.} \quad \boldsymbol{y} + \theta\sigma^2 (X_S^+)^\top \boldsymbol{e}_j \in \mathrm{Pol}(S)\end{aligned}$$

and

$$\begin{aligned}&U(S,j) = \hat{\beta}_{S,j} + \theta_U \sigma^2 (X_S^\top X_S)^{-1}_{jj} \\ &\text{where} \quad \theta_U = \max_{\theta\in\mathbb{R}} \theta \quad \text{s.t.} \quad \boldsymbol{y} + \theta\sigma^2 (X_S^+)^\top \boldsymbol{e}_j \in \mathrm{Pol}(S),\end{aligned}$$

respectively, but we can ignore the scaling factor σ^2 because

$$\begin{aligned}&\min\{\theta \in \mathbb{R}^n \mid \boldsymbol{y} + \theta (X_S^+)^\top \boldsymbol{e}_j \in \mathrm{Pol}(S)\} \\ &= \min\{\theta\sigma^2 \in \mathbb{R}^n \mid \boldsymbol{y} + \theta\sigma^2 (X_S^+)^\top \boldsymbol{e}_j \in \mathrm{Pol}(S)\}\end{aligned}$$

and

$$\begin{aligned}&\max\{\theta \in \mathbb{R}^n \mid \boldsymbol{y} + \theta (X_S^+)^\top \boldsymbol{e}_j \in \mathrm{Pol}(S)\} \\ &= \max\{\theta\sigma^2 \in \mathbb{R}^n \mid \boldsymbol{y} + \theta\sigma^2 (X_S^+)^\top \boldsymbol{e}_j \in \mathrm{Pol}(S)\}.\end{aligned}$$

□

7.2 Proof of Lemma 2

Proof. From (8), for all possible pairs of $(j', \xi) \in \bar{S}^{(t)} \times \{-1, 1, 0\},\ t \in [k]$, the constraint $\boldsymbol{y} + \theta\boldsymbol{\eta} \in \mathrm{Pol}(S^{(k)})$ is written as

$$(-s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}} + \xi \boldsymbol{x}_{:j'})^\top \Gamma^{(t)} (\boldsymbol{y} + \theta\boldsymbol{\eta}) \leq 0,$$

that is equivalent to

$$\text{if } (\xi \boldsymbol{x}_{:j'} - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}})^\top \Gamma^{(t)} \boldsymbol{\eta} < 0, \text{ then}$$
$$\frac{(s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}} - \xi \boldsymbol{x}_{:j'})^\top \Gamma^{(t)} \boldsymbol{y}}{(\xi \boldsymbol{x}_{:j'} - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}})^\top \Gamma^{(t)} \boldsymbol{\eta}} \leq \theta, \tag{23a}$$
$$\text{if } (\xi \boldsymbol{x}_{:j'} - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}})^\top \Gamma^{(t)} \boldsymbol{\eta} > 0, \text{ then}$$
$$\frac{(s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}} - \xi \boldsymbol{x}_{:j'})^\top \Gamma^{(t)} \boldsymbol{y}}{(\xi \boldsymbol{x}_{:j'} - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}})^\top \Gamma^{(t)} \boldsymbol{\eta}} \geq \theta. \tag{23b}$$

(i) First, the numerators of (23) are non-negative because of (8). (ii) Second, if the denominators of (23) are negative, then the minimum possible feasible θ would be

$$\max_{t \in [k]} \max_{(j', \xi) \in \bar{S}^{(t)} \times \{-1,1,0\}} \frac{(s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}} - \xi \boldsymbol{x}_{:j'})^\top \Gamma^{(t)} \boldsymbol{y}}{(\xi \boldsymbol{x}_{:j'} - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}})^\top \Gamma^{(t)} \boldsymbol{\eta}},$$

and the maximum possible feasible θ would be 0. (iii) Similarly, if the denominators of (23) are positive, then the maximum possible feasible θ would be

$$\min_{t \in [k]} \min_{(j', \xi) \in \bar{S}^{(t)} \times \{-1,1,0\}} \frac{(s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}} - \xi \boldsymbol{x}_{:j'})^\top \Gamma^{(t)} \boldsymbol{y}}{(\xi \boldsymbol{x}_{:j'} - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}})^\top \Gamma^{(t)} \boldsymbol{\eta}},$$

and minimum possible feasible θ would be 0. Since the requirements in (i), (ii) and (iii) must be satisfied for all possible $(j', \xi) \in \bar{S}^{(t)} \times \{-1, 1, 0\},\ t \in [k]$, θ_L and θ_U are given by (16a) and (16b), respectively. □

7.3 Proof of Theorem 3

Proof. Nothing that $0 \leq x_{i\tilde{j}'} \leq x_{ij'} \leq 1$ from (14), for any descendant node $\tilde{j}' \in Des(j')$. For any vector $\boldsymbol{v} \in \mathbb{R}^n$, the following inequalities are satisfied

$$\begin{aligned}
(\xi \boldsymbol{x}_{:\tilde{j}'} - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}})^\top \boldsymbol{v} &= \sum_{\xi v_i > 0} x_{i\tilde{j}'} \xi v_i + \sum_{\xi v_i < 0} x_{i\tilde{j}'} \xi v_i - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}}^\top \boldsymbol{v} \\
&\geq \sum_{\xi v_i < 0} x_{i\tilde{j}'} \xi v_i - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}}^\top \boldsymbol{v} \\
&\geq \sum_{\xi v_i < 0} x_{ij'} \xi v_i - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}}^\top \boldsymbol{v}, \qquad (24a) \\
(s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}} - \xi \boldsymbol{x}_{:\tilde{j}'})^\top \boldsymbol{v} &= s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}}^\top \boldsymbol{v} - \sum_{\xi v_i > 0} x_{i\tilde{j}'} \xi v_i - \sum_{\xi v_i < 0} x_{i\tilde{j}'} \xi v_i \\
&\geq s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}}^\top \boldsymbol{v} - \sum_{\xi v_i > 0} x_{i\tilde{j}'} \xi v_i \\
&\geq s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}}^\top \boldsymbol{v} - \sum_{\xi v_i > 0} x_{ij'} \xi v_i. \qquad (24b)
\end{aligned}$$

We prove the pruning condition (17) in the theorem. (i) First, from Lemma 2, any pairs $(j^{(t)}, \tilde{j}')$ such that $(\xi \boldsymbol{x}_{:\tilde{j}'} - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}})^\top \Gamma^{(t)} \boldsymbol{\eta} \geq 0$ are irrelevant to the solution θ_L. Also from (24a),

$$\begin{aligned}
&\sum_{i:\xi\left(\Gamma^{(t)}\boldsymbol{\eta}\right)_i < 0} x_{ij'} \xi \left(\Gamma^{(t)} \boldsymbol{\eta}\right)_i - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}}^\top \Gamma^{(t)} \boldsymbol{\eta} \geq 0 \\
&\qquad \Rightarrow \ (\xi \boldsymbol{x}_{:\tilde{j}'} - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}})^\top \Gamma^{(t)} \boldsymbol{\eta} \geq 0.
\end{aligned}$$

(ii) Moreover, when $(\xi \boldsymbol{x}_{:\tilde{j}'} - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}})^\top \Gamma^{(t)} \boldsymbol{\eta} < 0$, from (24a) and (24b),

$$\begin{aligned}
&\frac{(s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}} - \xi \boldsymbol{x}_{:j'})^\top \Gamma^{(t)} \boldsymbol{y}}{(\xi \boldsymbol{x}_{:j'} - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}})^\top \Gamma^{(t)} \boldsymbol{\eta}} \\
&\leq \frac{\max\{0, s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}}^\top \Gamma^{(t)} \boldsymbol{y} - \sum_{\xi\left(\Gamma^{(t)}\boldsymbol{y}\right)_i > 0} x_{ij'} \xi \left(\Gamma^{(t)} \boldsymbol{y}\right)_i\}}{\min\{0, \sum_{i:\xi\left(\Gamma^{(t)}\boldsymbol{\eta}\right)_i < 0} x_{ij'} \xi \left(\Gamma^{(t)} \boldsymbol{\eta}\right)_i - s_{j^{(t)}} \boldsymbol{x}_{:j^{(t)}}^\top \Gamma^{(t)} \boldsymbol{\eta}\}},
\end{aligned}$$

note that the numerators of above inequalities are non-negative because of (8). By combining (i) and (ii), if (17) holds, then (j, ℓ') for $\ell' \in Des(j')$ do not affect the solution of (16a). The first half of the theorem is proved, and the other half can be shown similarly. □

Part II

Robust machine learning by simulated annealing with continuous temperature parameter

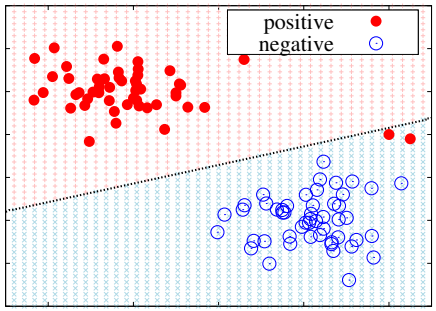
8 Introduction

The *support vector machine* (SVM) has been one of the most successful machine learning algorithms [29, 30, 31]. However, in recent practical machine learning applications with less reliable data that contains outliers, non-robustness of the SVM often causes considerable performance deterioration. In robust learning context, outliers indicate observations that have been contaminated, inaccurately recorded, or drawn from different distributions from other normal observations.

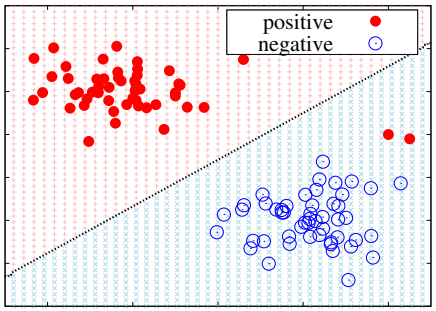
Figure 11 shows examples of robust SV classification and regression. In classification problems, we regard the instances whose labels are flipped from the ground truth as outliers. As illustrated in Figure 11(a), standard SV classifier may be highly affected by outliers. On the other hand, robust SV classifier can alleviate the effects of outliers as illustrated in Figure 11(b). In regression problems, we regard instances whose output response are highly contaminated than other normal observations as outliers. Standard SV regression tends to produce an unstable result as illustrated in Figure 11(c), while robust SV regression can alleviate the effects of outliers as illustrated in Figure 11(d).

8.1 Existing robust classification and regression methods

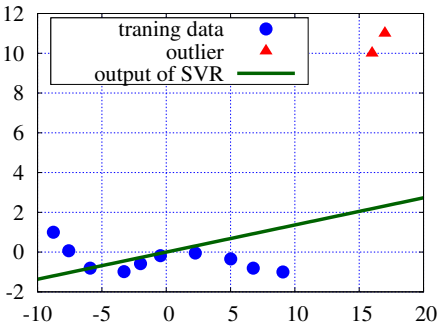
A great deal of efforts have been devoted to improve the robustness of the SVM and other similar learning algorithms [36, 28, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46]. In the context of SV classification, so-called *Ramp* loss function (see Figure 12(a)) is introduced for alleviating the effects of outliers. Similarly, robust loss function for SV regression (see Figure 12(b)) is also introduced. Robustness properties obtained by replacing the loss functions to these



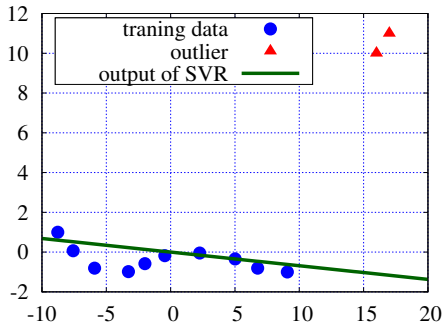
(a) Standard SV Classification



(b) Robust SV Classification



(c) Standard SV Regression



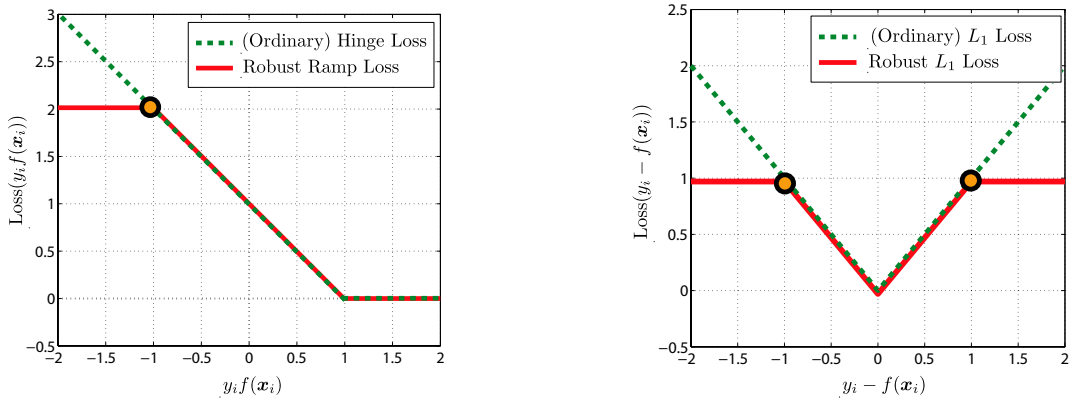
(d) Robust SV Regression

Figure 11: Illustrative examples of (a) standard SVC (support vector classification) and (b) robust SVC, (c) standard SVR (support vector regression) and (d) robust SVR on toy datasets. In robust SVM, the classification and regression results are not sensitive to the two red outliers in the right-hand side of the plots.

robust ones have been intensively studied in the context of robust statistics.

As illustrated in Figure 12, robust loss functions have two common properties. First, robust loss functions are essentially *non-convex* because they are designed to alleviate the effect of outliers [1]. If one uses a convex loss, even a single outlier can dominate the classification or regression result. Second, robust loss functions have *an additional parameter* for controlling the trade-off between robustness and efficiency. For example, in Ramp loss func-

[1] Xu et al. [41] introduced a robust learning approach that does not require non-convex optimization, in which non-convex loss functions are relaxed to be convex ones. In their approach, the resulting convex problems are formulated as semi-definite programs (SDPs), which are therefore inherently non-scalable. In our experience, the SDP method can be applied to datasets with up to a few hundred instances. Thus, we do not compare our method with the method in [41] in this paper.



(a) Robust loss function for SV classification (b) Robust loss function for SV regression

Figure 12: Robust loss functions for (a) SV classification and (b) SV regression. Note that these robust loss functions are essentially non-convex because they are designed to alleviate the effect of outliers. Furthermore, robust loss functions have additional hyper-parameter for controlling the trade-off between robustness and efficiency. For example, in Ramp loss function, one has to determine the breakpoint (the location of the orange circle) of the robust loss function. In plot (a), the breakpoint at margin -1.0 indicates that we regard the instances whose margin is smaller than -1.0 as outliers. Similarly, in plot (b), the breakpoints at ± 1.0 indicate that we regard the instances whose absolute residuals are greater than 1.0 as outliers.

tion in Figure 12(a), one must determine the breakpoint of the loss function (the location of the orange circle). Since such a tuning parameter governs the influence of outliers on the model, it must be carefully tuned based on the property of noise contained in the data set. Unless one has a clear knowledge about the properties of outliers, one has to tune the tuning parameter by using a model selection method such as cross-validation. These two properties suggest that one must solve many non-convex optimization problems for various values of the tuning parameter.

Unfortunately, existing non-convex optimization algorithms for robust SVM learning are not sufficiently efficient nor stable. The most common approach for solving non-convex optimization problems in robust SV classification and regression is *Difference of Convex (DC) programming* [37, 38, 39, 40, 42, 43], or its special form called *Concave Convex Procedure*

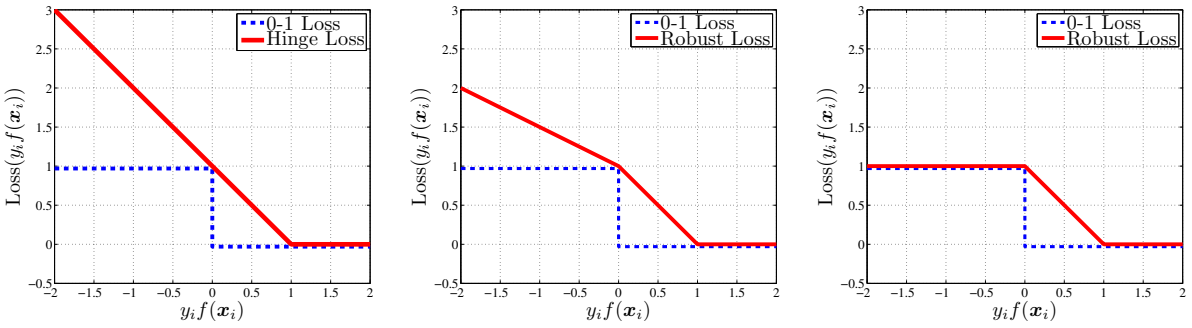
(CCCP) [52]. Since a solution obtained by these general non-convex optimization methods is only one of many local optimal solutions, one has to repeatedly solve the non-convex optimization problems from multiple different initial solutions. Furthermore, due to the non-convexity, robust SVM solutions with slightly different tuning parameter values can be significantly different, which makes the tuning-parameter selection problem highly unstable. To the best of our knowledge, there are no other existing studies that allows us to efficiently obtain stable sequence of multiple local optimal solutions for various tuning parameters values in the context of robust SV classification and regression.

8.2 Our contributions

In this paper, we address efficiency and stability issues of robust SV learning simultaneously by introducing a novel *homotopy* approach [2]. We use the parameterized formulations of robust SVC and SVR which bridge the standard SVM and fully robust SVM via a parameter that governs the influence of outliers. Our basic idea is to consider *homotopy* methods [48, 49, 50, 51] for tracing a path of solutions when the parameter is continuously changed. We call the parameter as the *homotopy parameter* and the path of solutions obtained by tracing the homotopy parameter as the *outlier path*. Figure13 and Figure14 illustrate how the robust loss functions for classification and regression problems can be gradually robustified, respectively.

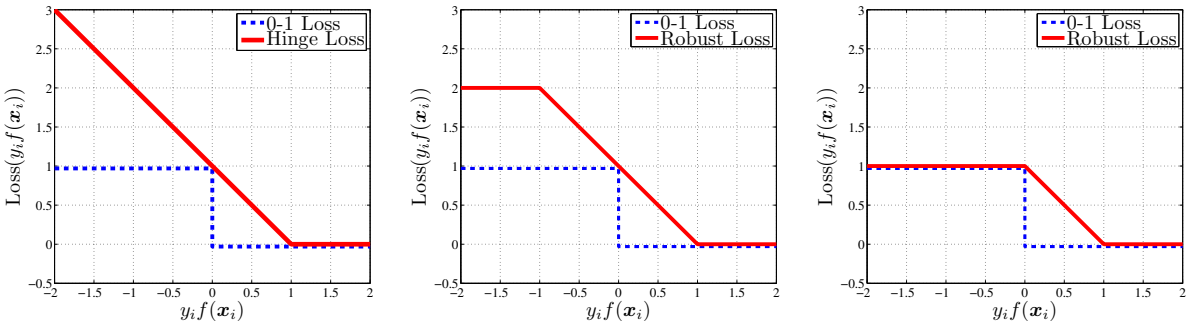
Our first technical contribution is in analyzing the properties of the outlier path for both classification and regression problems. In particular, we derive the necessary and sufficient conditions for SVC and SVR solutions to be locally optimal (note that the well-known Karush-Khun Tucker (KKT) conditions are only necessary, but not sufficient). Interestingly, the analyses indicate that the outlier paths contain a finite number of discontinuous points. To the best of our knowledge, the above property of robust learning has not been known previously.

[2] For regression problems, we study least absolute deviation (LAD) regression. It is straightforward to extend it to original SVR formulation with ε-insensitive loss function. In order to simplify the description, we often call LAD regression as SV regression (SVR). In what follows, we use the term SVM when we describe common properties of SVC and SVR.



$\theta = 1,\ s = 0$ $\theta = 0.5,\ s = 0$ $\theta = 0,\ s = 0$

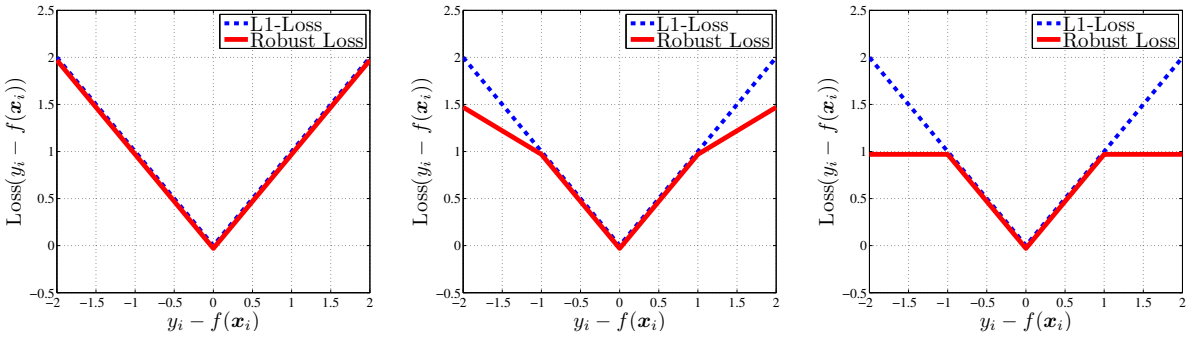
(a) Homotopy computation with decreasing θ from 1 to 0.



$\theta = 0,\ s = -\infty$ $\theta = 0,\ s = -1$ $\theta = 0,\ s = 0$

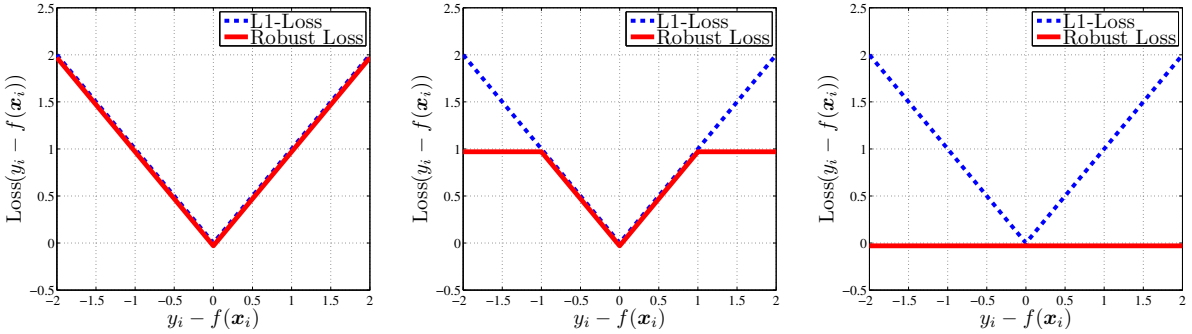
(b) Homotopy computation with increasing s from $-\infty$ to 0.

Figure 13: An illustration of the parameterized class of problems for robust SV classification (SVC). A loss function in each plot (red one) is characterized by two parameters $\theta \in [0, 1]$ and $s \in (-\infty, 0]$, which we call *robustness* parameters. When $\theta = 1$ or $s = -\infty$, the loss function is identical with the hinge loss function in standard SVM. On the other hand, when $\theta = 0$ and $s \neq -\infty$, the loss function is identical with the ramp loss function which has been used in several robust SV learning studies. In this paper, we consider a parameterized formulations of robust SVC which bridges the standard SVC and fully robust SVC via the homotopy parameters θ and/or s that governs the influence of outliers. The algorithm we introduce in this paper allows us to compute a path of solutions when we continuously change these homotopy parameters. In the top row and the bottom row, we consider paths of solutions with respect to the homotopy parameter θ and s, respectively.



$\theta = 1,\ |s| = 1$ $\theta = 0.5,\ |s| = 1$ $\theta = 0,\ |s| = 1$

(a) Homotopy computation with decreasing θ from 1 to 0.



$\theta = 0,\ |s| = \infty$ $\theta = 0,\ |s| = 1$ $\theta = 0,\ |s| = 0$

(b) Homotopy computation with decreasing s from $-\infty$ to 0.

Figure 14: An illustration of the parameterized class of problems for robust SV regression (SVR). A loss function in each plot (red one) is characterized by two parameters $\theta \in [0, 1]$ and $|s| \in [0, \infty)$, which we call *robustness* parameters. When $\theta = 1$ or $|s| = 0$, the loss function is identical with the loss function of least absolute deviation (LAD) regression. On the other hand, when $\theta = 0$ and $|s| \neq \infty$, the loss function is a robust function in which the influences of outliers are bounded. In this paper, we also consider a parameterized formulations of robust SVR which bridges the standard SVR and the robust SVR via the homotopy parameters θ and/or $|s|$ that governs the influence of outliers. The algorithm we introduce in this paper allows us to compute a path of solutions when we continuously change these homotopy parameters. In the top row and the bottom row, we consider paths of solutions with respect to the homotopy parameter θ and s, respectively.

Our second contribution is to develop an efficient algorithm for actually computing the outlier path based on the above theoretical investigation of the geometry of robust SVM solutions. Here, we use parametric programming technique [48, 49, 50, 51], which is often used for computing the *regularization path* in machine learning literature. The main technical challenge here is how to handle the discontinuous points in the outlier path. We overcome this difficulty by precisely analyzing the necessary and sufficient conditions for the local optimality. We develop an algorithm that can precisely detect such discontinuous points, and *jump* to find a strictly better local optimal solution.

Experimental results indicate that our proposed method can find better robust SVM solutions more efficiently than alternative method based on CCCP. We conjecture that there are two reasons why favorable results can be obtained with our method. At first, the outlier path shares similar advantage as *simulated annealing* [47]. Simulated annealing is known to find better local solutions in many non-convex optimization problems by solving a sequence of solutions along with so-called *temperature* parameter. If we regard the homotopy parameter as the temperature parameter, our outlier path algorithm can be interpreted as simulated annealing with infinitesimal step size (as we explain in the following sections, our algorithm represents the path of local solutions as a function of the homotopy parameter). Since our algorithm provides the path of local solutions, unlike other non-convex optimization algorithms such as CCCP, two solutions with slightly different homotopy parameter values tend to be similar, which makes the tuning parameter selection stable. According to our experiments, choice of the homotopy parameter is quite sensitive to the generalization performances. Thus, it is important to finely tune the homotopy parameter. Since our algorithm can compute the path of solutions, it is much more computationally efficient than running CCCP many times at different homotopy parameter values.

8.3 Organization of the paper

After we formulate robust SVC and SVR as parameterized optimization problems in § 9, we derive in § 10 the *necessary* and *sufficient* conditions for a robust SVM solution to be locally optimal, and show that there exist a finite number of discontinuous points in the local solution path. We then propose an efficient algorithm in § 11 that can precisely de-

tect such discontinuous points and *jump* to find a strictly better local optimal solution. In § 12, we experimentally demonstrate that our proposed method, named the *outlier path algorithm*, outperforms the existing robust SVM algorithm based on CCCP or DC programming. Finally, we conclude in § 13.

In this paper, we have extended our robusitification path framework to the regression problem, and many more experimental evaluations have been conducted. To the best of our knowledge, the homotopy method [48, 49, 50, 51] is first used in our preliminary conference paper in the context of robust learning, So far, homotopy-like methods have been (often implicitly) used for non-convex optimization problems in the context of sparse modeling [53, 54, 55] and semi-supervised learning [56].

9 Parameterized Formulation of Robust SVM

In this section, we first formulate robust SVMs for classification and regression problems, which we denote by robust SVC (SV classification) and robust SVR (SV regression), respectively. Then, we use parameterized formulation both for robust SVC and SVR, where the parameter governs the influence of outlieres to the model. The problem is reduced to ordinary non-robust SVM at one end of the parameter, while the problem corresponds to fully-robust SVM at the other end of the parameter. In the following sections, we develop algorithms for computing the path of local optimal solutions when the parameter is changed form one end to the other.

9.1 Robust SV Classification

Let us consider a binary classification problem with n instances and d features. We denote the training set as $\{(\boldsymbol{x}_i, y_i)\}_{i\in\mathbb{N}_n}$ where $\boldsymbol{x}_i \in \mathcal{X}$ is the input vector in the input space $\mathcal{X} \subset \mathbb{R}^d$, $y_i \in \{-1, 1\}$ is the binary class label, and the notation $\mathbb{N}_n := \{1, \ldots, n\}$ represents the set of natural numbers up to n. We write the decision function as

$$f(\boldsymbol{x}) := \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}), \tag{25}$$

where $\boldsymbol{\phi}$ is the feature map implicitly defined by a kernel $\boldsymbol{K}$, $\boldsymbol{w}$ is a vector in the feature space, and $^\top$ denotes the transpose of vectors and matrices.

We introduce the following class of optimization problems *parameterized* by θ and s:

$$\min_{\boldsymbol{w}} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \ell(y_i f(\boldsymbol{x}_i);\ \theta, s), \tag{26}$$

where $C > 0$ is the regularization parameter which controls the balance between the first regularization term and the second loss term. The loss function ℓ is characterized by a pair of parameters $\theta \in [0, 1]$ and $s \leq 0$ as

$$\ell(z;\ \theta, s) := \begin{cases} [0, 1-z]_+, & z \geq s, \\ 1 - \theta z - s, & z < s, \end{cases} \tag{27}$$

where $[z]_+ := \max\{0, z\}$. We refer to θ and s as the *homotopy parameters*. Figure 13 shows the loss functions for several θ and s. Note that when $\theta = 1$ or $s = \infty$, the optimization problem (26) is equivalent to the standard formulation of SVC, in particular, so-called *soft-margin* SVC. In this case, the loss function is reduced to the well-known *hinge loss* function, which linearly penalizes $1 - y_i f(\boldsymbol{x}_i)$ when $y_i f(\boldsymbol{x}_i) \leq 1$, and gives no penalty for $y_i f(\boldsymbol{x}_i) > 1$ (the left-most column in Figure 13). The input vector $\boldsymbol{x}_i$ having $y_i f(\boldsymbol{x}_i) \leq 1$ at the final solution is called a support vector by which the resulting decision boundary is defined, while the other input vectors do not effect on the boundary because they have no effect on the objective function. The first homotopy parameter θ can be interpreted as the *weight* for an outlier: $\theta = 1$ indicates that the influence of an outlier is the same as an inlier, while $\theta = 0$ indicates that outliers are completely ignored. The second homotopy parameter $s \leq 0$ can be interpreted as the threshold for deciding outliers and inliers. When $\theta = 0$ and $s = 0$, the loss function is reduced to the *ramp loss* function (the right-most column in Figure 13) to which we have referred as fully robust SVM. A variety of robust loss functions, including the functions that we employed here, have appeared in [28].

In the following sections, we consider two types of homotopy methods. In the first method, we fix $s = 0$, and gradually change θ from 1 to 0 (see the top five plots in Figure 13). In the second method, we fix $\theta = 0$ and gradually change s from $-\infty$ to 0 (see the bottom five plots in Figure 13). The optimization (26) is a *convex* problem when $\theta = 1$ or $s = -\infty$, in

which ℓ is the hinge loss, while it is *non-convex* when $\theta = 0$ and $s = 0$, in which ℓ is the ramp loss. Therefore, each of the above two homotopy methods can be interpreted as the process of tracing a sequence of solutions when the optimization problem is gradually modified from convex to non-convex. By doing so, we expect to find good local optimal solutions because such a process can be interpreted as *simulated annealing* [47]. In addition, we can adaptively control the degree of robustness by selecting the best θ or s based on some model selection scheme.

9.2 Robust SV Regression

Let us next consider a regression problem. We denote the training set of the regression problem as $\{(\boldsymbol{x}_i, y_i)\}_{i \in \mathbb{N}_n}$, where the input $\boldsymbol{x}_i \in \mathcal{X}$ is the input vector as the classification case, while the output $y_i \in \mathbb{R}$ is a real scalar. We consider a regression function $f(\boldsymbol{x})$ in the form of (25). SV regression is formulated as

$$\min_{\boldsymbol{w}} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{i=1}^{n} \ell(y_i - f(\boldsymbol{x}_i);\ \theta, s), \tag{28}$$

where $C > 0$ is the regularization parameter, and the loss function ℓ is defined as

$$\ell(z;\ \theta, s) := \begin{cases} |z|, & |z| < s, \\ (|z| - s)\theta + s, & |z| \geq s. \end{cases} \tag{29}$$

The loss function in (29) has two parameters $\theta \in [0, 1]$ and $s \in [0, \infty)$ as the classification case. Figure 14 shows the loss functions for several θ and s.

10 Local Optimality

In order to use the homotopy approach, we need to clarify the continuity of the local solution path. To this end, we investigate several properties of local solutions of robust SVM, and derive the necessary and sufficient conditions. Interestingly, our analysis reveals that the local solution path has a finite number of *discontinuous* points. The theoretical results presented here form the basis of our novel homotopy algorithm given in the next section that can properly handle the above discontinuity issue. We first discuss the local optimality

of robust SVC in detail in § 10.1 and § 10.2, and then present the corresponding result of robust SVR briefly in § 10.3. Here we call a solution to be locally optimal if there is no strictly better feasible solutions in its neighborhood.

10.1 Conditionally Optimal Solutions (for Robust SVC)

The basic idea of our theoretical analysis is to reformulate the robust SVC learning problem as a combinatorial optimization problem. We consider a partition of the instances $\mathbb{N}_n := \{1, \ldots, n\}$ into two disjoint sets $\mathcal{I}$ and $\mathcal{O}$. The instances in $\mathcal{I}$ and $\mathcal{O}$ are defined as $\mathcal{I}$nliers and $\mathcal{O}$utliers, respectively. Here, we restrict that the margin $y_i f(\boldsymbol{x}_i)$ of an inlier should be larger than or equal to s, while that of an outlier should be smaller than or equal to [3] s. We denote the partition as $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\} \in 2^{\mathbb{N}_n}$, where $2^{\mathbb{N}_n}$ is the power set[4] of $\mathbb{N}_n$. Given a partition $\mathcal{P}$, the above restrictions define the feasible region of the solution f in the form of a convex polytope:

$$\mathrm{pol}(\mathcal{P}; s) := \left\{ f \;\middle|\; \begin{array}{ll} y_i f(\boldsymbol{x}_i) \geq s, & i \in \mathcal{I} \\ y_i f(\boldsymbol{x}_i) \leq s, & i \in \mathcal{O} \end{array} \right\}. \tag{30}$$

Using the notion of the convex polytopes, the optimization problem (26) can be rewritten as

$$\min_{\mathcal{P} \in 2^{\mathbb{N}_n}} \left(\min_{f \in \mathrm{pol}(\mathcal{P}; s)} J_{\mathcal{P}}(f; \theta) \right), \tag{31}$$

where the objective function $J_{\mathcal{P}}$ is defined as[5]

$$J_{\mathcal{P}}(f; \theta) := \frac{1}{2}||\boldsymbol{w}||_2^2 + C \left(\sum_{i \in \mathcal{I}} [1 - y_i f(\boldsymbol{x}_i)]_+ + \theta \sum_{i \in \mathcal{O}} [1 - y_i f(\boldsymbol{x}_i)]_+ \right).$$

Note that the right hand side depends on $\mathcal{P}$ though $\mathcal{I}$ and $\mathcal{O}$.

When the partition $\mathcal{P}$ is fixed, it is easy to confirm that the inner minimization problem of (31) is a convex problem.

[3] Note that an instance with the margin $y_i f(\boldsymbol{x}_i) = s$ can be the member of either $\mathcal{I}$ or $\mathcal{O}$.

[4] The power set means that there are 2^n patterns that each of the instances belongs to either $\mathcal{I}$ or $\mathcal{O}$.

[5] Note that we omitted the constant terms irrelevant to the optimization problem.

Definition 1 (Conditionally optimal solutions)**.** *Given a partition $\mathcal{P}$, the solution of the following convex problem is said to be the conditionally optimal solution:*

$$f^*_{\mathcal{P}} := \arg \min_{f \in \mathrm{pol}(\mathcal{P};s)} J_{\mathcal{P}}(f;\theta). \tag{32}$$

The formulation in (31) is interpreted as a combinatorial optimization problem of finding the best solution from all the 2^n conditionally optimal solutions $f^*_{\mathcal{P}}$ corresponding to all possible 2^n partitions[6].

Using the representer theorem or convex optimization theory, we can show that any conditionally optimal solution can be written as

$$f^*_{\mathcal{P}}(\boldsymbol{x}) := \sum_{j \in \mathbb{N}_n} \alpha^*_j y_j K(\boldsymbol{x}, \boldsymbol{x}_j), \tag{33}$$

where $\{\alpha^*_j\}_{j\in\mathbb{N}_n}$ are the optimal Lagrange multipliers. The following lemma summarizes the KKT optimality conditions of the conditionally optimal solution $f^*_{\mathcal{P}}$.

Lemma 5. *The KKT conditions of the convex problem (32) is written as*

$$y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) > 1 \;\Rightarrow\; \alpha^*_i = 0, \tag{34a}$$

$$y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) = 1 \;\Rightarrow\; \alpha^*_i \in [0, C], \tag{34b}$$

$$s < y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) < 1 \;\Rightarrow\; \alpha^*_i = C, \tag{34c}$$

$$y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) = s, i \in \mathcal{I} \;\Rightarrow\; \alpha^*_i \geq C, \tag{34d}$$

$$y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) = s, i \in \mathcal{O} \;\Rightarrow\; \alpha^*_i \leq C\theta, \tag{34e}$$

$$y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) < s \;\Rightarrow\; \alpha^*_i = C\theta. \tag{34f}$$

The proof is presented in § 14.1.

10.2 The necessary and sufficient conditions for local optimality (for Robust SVC)

From the definition of conditionally optimal solutions, it is clear that a local optimal solution must be conditionally optimal within the convex polytope $\mathrm{pol}(\mathcal{P};s)$. However, the condi-

[6] For some partitions $\mathcal{P}$, the convex problem (32) might not have any feasible solutions.

tional optimality does not necessarily indicate the local optimality as the following theorem suggests.

Theorem 6. *For any $\theta \in [0,1)$ and $s \leq 0$, consider the situation where a conditionally optimal solution $f^*_{\mathcal{P}}$ is at the boundary of the convex polytope* $\mathrm{pol}(\mathcal{P}; s)$*, i.e., there exists at least an instance such that $y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) = s$. In this situation, if we define a new partition $\tilde{\mathcal{P}} := \{\tilde{\mathcal{I}}, \tilde{\mathcal{O}}\}$ as*

$$\tilde{\mathcal{I}} \leftarrow \mathcal{I} \setminus \{i \in \mathcal{I} | y_i f^*(\boldsymbol{x}_i) = s\} \cup \{i \in \mathcal{O} | y_i f^*(\boldsymbol{x}_i) = s\}, \tag{35a}$$

$$\tilde{\mathcal{O}} \leftarrow \mathcal{O} \setminus \{i \in \mathcal{O} | y_i f^*(\boldsymbol{x}_i) = s\} \cup \{i \in \mathcal{I} | y_i f^*(\boldsymbol{x}_i) = s\}, \tag{35b}$$

then the new conditionally optimal solution $f^*_{\tilde{\mathcal{P}}}$ is strictly better than the original conditionally optimal solution $f^*_{\mathcal{P}}$, i.e.,

$$J_{\tilde{\mathcal{P}}}(f^*_{\tilde{\mathcal{P}}}; \theta) < J_{\mathcal{P}}(f^*_{\mathcal{P}}; \theta). \tag{36}$$

The proof is presented in § 14.2. Theorem 6 indicates that if $f^*_{\mathcal{P}}$ is at the boundary of the convex polytope $\mathrm{pol}(\mathcal{P}; s)$, i.e., if there is one or more instances such that $y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) = s$, then $f^*_{\mathcal{P}}$ is NOT locally optimal because there is a strictly better solution in the opposite side of the boundary.

The following theorem summarizes the necessary and sufficient conditions for local optimality. Note that, in non-convex optimization problems, the KKT conditions are necessary but not sufficient in general.

Theorem 7. *For $\theta \in [0,1)$ and $s \leq 0$,*

$$y_i f^*(\boldsymbol{x}_i) > 1 \quad \Rightarrow \quad \alpha^*_i = 0, \tag{37a}$$

$$y_i f^*(\boldsymbol{x}_i) = 1 \quad \Rightarrow \quad \alpha^*_i \in [0, C], \tag{37b}$$

$$s < y_i f^*(\boldsymbol{x}_i) < 1 \quad \Rightarrow \quad \alpha^*_i = C, \tag{37c}$$

$$y_i f^*(\boldsymbol{x}_i) < s \quad \Rightarrow \quad \alpha^*_i = C\theta, \tag{37d}$$

$$y_i f^*(\boldsymbol{x}_i) \neq s, \qquad \forall i \in \mathbb{N}_n, \tag{37e}$$

are necessary and sufficient for f^* to be locally optimal.

The proof is presented in § 14.3. The condition (37e) indicates that the solution at the boundary of the convex polytope is not locally optimal. Figure 15 illustrates when a conditionally optimal solution can be locally optimal with a certain θ or s.

Theorem 7 suggests that, whenever the local solution path computed by the homotopy approach encounters a boundary of the current convex polytope at a certain θ or s, the solution is not anymore locally optimal. In such cases, it is better to search a local optimal solution at that θ or s, and restart the local solution path from the new one. In other words, the local solution path has *discontinuity* at that θ or s. Fortunately, Theorem 6 tells us how to handle such a situation. If the local solution path arrives at the boundary, it can *jump* to the new conditionally optimal solution $f^*_{\tilde{\mathcal{P}}}$ which is located on the opposite side of the boundary. This jump operation is justified because the new solution is shown to be strictly better than the previous one. Figure 15 (c) and (d) illustrate such a situation.

10.3 Local optimality of SV Regression

In order to derive the necessary and sufficient conditions of the local optimality in robust SVR, with abuse of notation, let us consider a partition of the instances $\mathbb{N}_n$ into two disjoint sets $\mathcal{I}$ and $\mathcal{O}$, which represent inliers and outliers, respectively. In regression problems, an instance $(\boldsymbol{x}_i, y_i)$ is regarded as an outlier if the absolute residual $|y_i - f(\boldsymbol{x}_i)|$ is sufficiently large. Thus, we define inliers and outliers of regression problem as

$$\begin{aligned} \mathcal{I} &:= \{i \in \mathbb{N}_n |\ |y_i - f(\boldsymbol{x}_i)| \leq s\}, \\ \mathcal{O} &:= \{i \in \mathbb{N}_n |\ |y_i - f(\boldsymbol{x}_i)| \geq s\}. \end{aligned}$$

Given a partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\} \in 2^{\mathbb{N}_n}$, the feasible region of the solution f is represented as a convex polytope:

$$\mathrm{pol}(\mathcal{P}; s) := \left\{ f \;\middle|\; \begin{array}{ll} |y_i - f(\boldsymbol{x}_i)| \leq s, & i \in \mathcal{I}, \\ |y_i - f(\boldsymbol{x}_i)| \geq s, & i \in \mathcal{O} \end{array} \right\}. \tag{38}$$

Then, as in the classification case, the optimization problem (28) can be rewritten as

$$\min_{\mathcal{P}} \left(\min_{f \in \mathrm{pol}(\mathcal{P};s)} J_{\mathcal{P}}(f;\theta) \right), \tag{39}$$

where the objective function $J_{\mathcal{P}}$ is defined as

$$J_{\mathcal{P}}(f;\theta) := \frac{1}{2}||\boldsymbol{w}||_2^2 + C \left(\sum_{i \in \mathcal{I}} |y_i - f(\boldsymbol{x}_i)| + \theta \sum_{i \in \mathcal{O}} |y_i - f(\boldsymbol{x}_i)| \right).$$

Since the inner problem of (39) is a convex problem, any conditionally optimal solution can be written as

$$f^*_{\mathcal{P}}(\boldsymbol{x}) := \sum_{j \in \mathbb{N}_n} \alpha^*_j K(\boldsymbol{x}, \boldsymbol{x}_j). \tag{40}$$

The KKT conditions of $f^*_{\mathcal{P}}(\boldsymbol{x})$ are written as

$$|y_i - f^*_{\mathcal{P}}(\boldsymbol{x}_i)| = 0 \;\Rightarrow\; 0 \leq |\alpha^*_i| \leq C, \tag{41a}$$

$$0 \leq |y_i - f^*_{\mathcal{P}}(\boldsymbol{x}_i)| < s \;\Rightarrow\; |\alpha^*_i| = C, \tag{41b}$$

$$|y_i - f^*_{\mathcal{P}}(\boldsymbol{x}_i)| = s, i \in \mathcal{I} \;\Rightarrow\; |\alpha^*_i| \geq C, \tag{41c}$$

$$|y_i - f^*_{\mathcal{P}}(\boldsymbol{x}_i)| = s, i \in \mathcal{O} \;\Rightarrow\; |\alpha^*_i| \leq \theta C, \tag{41d}$$

$$|y_i - f^*_{\mathcal{P}}(\boldsymbol{x}_i)| > s \;\Rightarrow\; |\alpha^*_i| = \theta C. \tag{41e}$$

Based on the same discussion as § 10.2, the necessary and sufficient conditions for the local optimality of robust SVR are summarized as the following theorem:

Theorem 8. *For $\theta \in [0, 1)$ and $s \geq 0$,*

$$|y_i - f^*_{\mathcal{P}}(\boldsymbol{x}_i)| = 0 \;\Rightarrow\; 0 \leq |\alpha^*_i| \leq C, \tag{42a}$$

$$0 \leq |y_i - f^*_{\mathcal{P}}(\boldsymbol{x}_i)| < s \;\Rightarrow\; |\alpha^*_i| = C, \tag{42b}$$

$$|y_i - f^*_{\mathcal{P}}(\boldsymbol{x}_i)| > s \;\Rightarrow\; |\alpha^*_i| = \theta C, \tag{42c}$$

$$|y_i - f^*_{\mathcal{P}}(\boldsymbol{x}_i)| \neq s. \tag{42d}$$

are necessary and sufficient for f^* to be locally optimal.

We omit the proof of this theorem because they can be easily derived in the same way as Theorem 7.

Algorithm 2 Outlier Path Algorithm

1: Initialize the solution f by solving the standard SVM.

2: Initialize the partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\}$ as follows:

$$\begin{aligned} \mathcal{I} &\leftarrow \{i \in \mathbb{N}_n | y_i f(\boldsymbol{x}_i) \leq s\}, \\ \mathcal{O} &\leftarrow \{i \in \mathbb{N}_n | y_i f(\boldsymbol{x}_i) > s\}. \end{aligned}$$

3: $\theta \leftarrow 1$ for OP-θ; $s \leftarrow \min_{i \in \mathbb{N}_n} y_i f(\boldsymbol{x}_i)$ for OP-s.

4: **while** $\theta > 0$ for OP-θ; $s < 0$ for OP-s **do**

5: **if** $(y_i f(\boldsymbol{x}_i) \neq s \ \forall \ i \in \mathbb{N}_n)$ **then**

6: Run C-step.

7: **else**

8: Run D-step.

9: **end if**

10: **end while**

11 Outlier Path Algorithm

Based on the analysis presented in the previous section, we develop a novel homotopy algorithm for robust SVM. We call the proposed method the *outlier-path (OP)* algorithm. For simplicity, we consider homotopy path computation involving either θ or s, and denote the former as OP-θ and the latter as OP-s. OP-θ computes the local solution path when θ is gradually decreased from 1 to 0 with fixed $s = 0$, while OP-s computes the local solution path when s is gradually increased from $-\infty$ to 0 with fixed $\theta = 0$.

The local optimality of robust SVM in the previous section shows that the path of local optimal solutions has finite discontinuous points that satisfy (37e) or (42d). Below, we introduce an algorithm that appropriately handles those discontinuous points. In this section, we only describe the algorithm for robust SVC. All the methodologies described in this section can be easily extended to robust SVR counterpart.

11.1 Overview

The main flow of the OP algorithm is described in Algorithm 2. The solution f is initialized by solving the standard (convex) SVM, and the partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\}$ is defined to satisfy the constraints in (30). The algorithm mainly switches over the two steps called the *continuous step (C-step)* and the *discontinuous step (D-step)*.

In the C-step (Algorithm 3), a continuous path of local solutions is computed for a sequence of gradually decreasing θ (or increasing s) within the convex polytope $\mathrm{pol}(\mathcal{P}; s)$ defined by the current partition $\mathcal{P}$. If the local solution path encounters a boundary of the convex polytope, i.e., if there exists at least an instance such that $y_i f(\boldsymbol{x}_i) = s$, then the algorithm stops updating θ (or s) and enters the D-step.

In the D-step (Algorithm 4), a better local solution is obtained for fixed θ (or s) by solving a convex problem defined over another convex polytope in the opposite side of the boundary (see Figure 15(d)). If the new solution is again at a boundary of the new polytope, the algorithm repeatedly calls the D-step until it finds the solution in the strict interior of the current polytope.

The C-step can be implemented by any homotopy algorithms for solving a sequence of quadratic problems (QP). In OP-θ, the local solution path can be exactly computed because the path within a convex polytope can be represented as piecewise-linear functions of the homotopy parameter θ. In OP-s, the C-step is trivial because the optimal solution is shown to be constant within a convex polytope. In § 11.2 and § 11.3, we will describe the details of our implementation of the C-step for OP-θ and OP-s, respectively.

In the D-step, we only need to solve a single quadratic problem (QP). Any QP solver can be used in this step. We note that the *warm-start* approach [57] is quite helpful in the D-step because the difference between two conditionally optimal solutions in adjacent two convex polytopes is typically very small. In § 11.4, we describe the details of our implementation of the D-step. Figure 16 illustrates an example of the local solution path obtained by OP-θ.

11.2 Continuous-Step for OP-θ

In the C-step, the partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\}$ is fixed, and our task is to solve a sequence of convex quadratic problems (QPs) parameterized by θ within the convex polytope $\mathrm{pol}(\mathcal{P}; s)$. It has been known in optimization literature that a certain class of parametric convex QP can be

Algorithm 3 Continuous Step (C-step)

1: **while** $(y_i f(\boldsymbol{x}_i) \neq s \ \forall \ i \in \mathbb{N}_n)$ **do**

2: Solve the sequence of convex problems,

$$\min_{f \in \mathrm{pol}(\mathcal{P};s)} J_{\mathcal{P}}(f;\theta),$$

for gradually decreasing θ in OP-θ or gradually increasing s in OP-s.

3: **end while**

Algorithm 4 Discontinuous Step (D-step)

1: Update the partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\}$ as follows:

$$\begin{aligned}\mathcal{I} &\leftarrow \mathcal{I} \setminus \{i \in \mathcal{I} | y_i f(\boldsymbol{x}_i) = s\} \cup \{i \in \mathcal{O} | y_i f(\boldsymbol{x}_i) = s\},\\ \mathcal{O} &\leftarrow \mathcal{O} \setminus \{i \in \mathcal{O} | y_i f(\boldsymbol{x}_i) = s\} \cup \{i \in \mathcal{I} | y_i f(\boldsymbol{x}_i) = s\}.\end{aligned}$$

2: Solve the following convex problem for fixed θ and s:

$$\min_{f \in \mathrm{pol}(\mathcal{P};s)} J_{\mathcal{P}}(f;\theta).$$

exactly solved by exploiting the piecewise linearity of the solution path [51]. We can easily show that the local solution path of OP-θ within a convex polytope is also represented as a piecewise-linear function of θ. The algorithm presented here is similar to the regularization path algorithm for SVM given in [58].

Let us consider a partition of the inliers in $\mathcal{I}$ into the following three disjoint sets:

$$\begin{aligned}\mathcal{R} &:= \{i | 1 < y_i f(\boldsymbol{x}_i)\},\\ \mathcal{E} &:= \{i | y_i f(\boldsymbol{x}_i) = 1\},\\ \mathcal{L} &:= \{i | s < y_i f(\boldsymbol{x}_i) < 1\}.\end{aligned}$$

For a given fixed partition $\{\mathcal{R}, \mathcal{E}, \mathcal{L}, \mathcal{O}\}$, the KKT conditions of the convex problem (32) indicate that

$$\alpha_i = 0 \ \forall \ i \in \mathcal{R}, \quad \alpha_i = C \ \forall \ i \in \mathcal{L}, \quad \alpha_i = C\theta \ \forall \ i \in \mathcal{O}.$$

The KKT conditions also imply that the remaining Lagrange multipliers $\{\alpha_i\}_{i\in\mathcal{E}}$ must satisfy the following linear system of equations:

$$y_i f(\boldsymbol{x}_i) = \sum_{j\in\mathbb{N}_n} \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 \ \forall\ i \in \mathcal{E}$$

$$\Leftrightarrow \quad \boldsymbol{Q}_{\mathcal{E}\mathcal{E}}\boldsymbol{\alpha}_{\mathcal{E}} = \mathbf{1} - \boldsymbol{Q}_{\mathcal{E}\mathcal{L}}\mathbf{1}C - \boldsymbol{Q}_{\mathcal{E}\mathcal{O}}\mathbf{1}C\theta, \tag{43}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ is an $n \times n$ matrix whose $(i,j)^{\text{th}}$ entry is defined as $Q_{ij} := y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Here, a notation such as $\boldsymbol{Q}_{\mathcal{E}\mathcal{L}}$ represents a submatrix of $\boldsymbol{Q}$ having only the rows in the index set $\mathcal{E}$ and the columns in the index set $\mathcal{L}$. By solving the linear system of equations (43), the Lagrange multipliers $\alpha_i, i \in \mathbb{N}_n$, can be written as an affine function of θ.

Noting that $y_i f(\boldsymbol{x}_i) = \sum_{j\in\mathbb{N}_n} \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is also represented as an affine function of θ, any changes of the partition $\{\mathcal{R}, \mathcal{E}, \mathcal{L}\}$ can be exactly identified when the homotopy parameter θ is continuously decreased. Since the solution path linearly changes for each partition of $\{\mathcal{R}, \mathcal{E}, \mathcal{L}\}$, the entire path is represented as a continuous piecewise-linear function of the homotopy parameter θ. We denote the points in $\theta \in [0, 1)$ at which members of the sets $\{\mathcal{R}, \mathcal{E}, \mathcal{L}\}$ change as *break-points* θ_{BP}.

Using the piecewise-linearity of $y_i f(\boldsymbol{x}_i)$, we can also identify when we should switch to the D-step. Once we detect an instance satisfying $y_i f(\boldsymbol{x}_i) = s$, we exit the C-step and enter the D-step.

11.3 Continuous-Step for OP-s

Since θ is fixed to 0 in OP-s, the KKT conditions (34) yields

$$\alpha_i = 0 \ \forall\ i \in \mathcal{O}.$$

This means that outliers have no influence on the solution and thus the conditionally optimal solution $f^*_{\mathcal{P}}$ does not change with s as long as the partition $\mathcal{P}$ is unchanged. The only task in the C-step for OP-s is therefore to find the next s that changes the partition $\mathcal{P}$. Such s can be simply found as

$$s \ \leftarrow \ \min_{i\in\mathcal{L}} y_i f(\boldsymbol{x}_i).$$

11.4 Discontinuous-Step (for both OP-θ and OP-s)

As mentioned before, any convex QP solver can be used for the D-step. When the algorithm enters the D-step, we have the conditionally optimal solution $f^*_{\mathcal{P}}$ for the partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\}$. Our task here is to find another conditionally optimal solution $f^*_{\tilde{\mathcal{P}}}$ for $\tilde{\mathcal{P}} := \{\tilde{\mathcal{I}}, \tilde{\mathcal{O}}\}$ given by (35).

Given that the difference between the two solutions $f^*_{\mathcal{P}}$ and $f^*_{\tilde{\mathcal{P}}}$ is typically small, the D-step can be efficiently implemented by a technique used in the context of incremental learning [59].

Let us define

$$\Delta_{\mathcal{I}\to\mathcal{O}} := \{i \in \mathcal{I} \mid y_i f_{\mathcal{P}}(\boldsymbol{x}_i) = s\},$$
$$\Delta_{\mathcal{O}\to\mathcal{I}} := \{i \in \mathcal{O} \mid y_i f_{\mathcal{P}}(\boldsymbol{x}_i) = s\}.$$

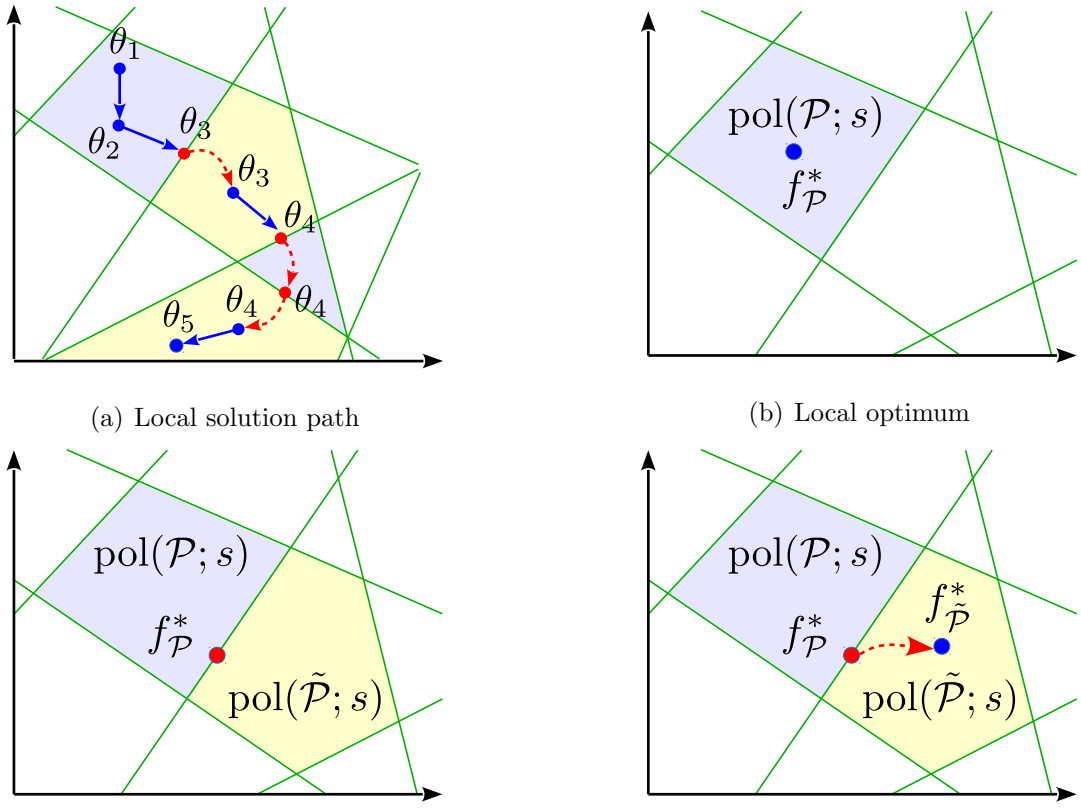
Then, we consider the following parameterized problem with parameter $\mu \in [0, 1]$:

$$f_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i; \mu) := f_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i) + \mu \Delta f_i \ \forall \ i \in \mathbb{N}_n,$$

where

$$\Delta f_i := y_i \left[\begin{array}{cc} \boldsymbol{K}_{i,\Delta_{\mathcal{I}\to\mathcal{O}}} & \boldsymbol{K}_{i,\Delta_{\mathcal{O}\to\mathcal{I}}} \end{array} \right] \left[\begin{array}{c} \boldsymbol{\alpha}^{(\mathrm{bef})}_{\Delta_{\mathcal{I}\to\mathcal{O}}} - \mathbf{1}C\theta \\ \boldsymbol{\alpha}^{(\mathrm{bef})}_{\Delta_{\mathcal{O}\to\mathcal{I}}} - \mathbf{1}C \end{array} \right],$$

and $\boldsymbol{\alpha}^{(\mathrm{bef})}$ be the corresponding $\boldsymbol{\alpha}$ at the beginning of the D-Step. We can show that $f_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i; \mu)$ is reduced to $f_{\mathcal{P}}(\boldsymbol{x}_i)$ when $\mu = 1$, while it is reduced to $f_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i)$ when $\mu = 0$ for all $i \in \mathbb{N}_n$. By using a similar technique to incremental learning [59], we can efficiently compute the path of solutions when μ is continuously changed from 1 to 0. This algorithm behaves similarly to the C-step in OP-θ. The implementation detail of the D-step is described in § 15.



(a) Local solution path

(b) Local optimum

(c) Not local optimum

(d) Local optimum

Figure 15: Solution space of robust SVC. (a) The arrows indicate a local solution path when θ is gradually moved from θ_1 to θ_5 (see § 11 for more details). (b) $f^*_{\mathcal{P}}$ is locally optimal if it is at the strict interior of the convex polytope $\mathrm{pol}(\mathcal{P}; s)$. (c) If $f^*_{\mathcal{P}}$ exists at the boundary, then $f^*_{\mathcal{P}}$ is feasible, but not locally optimal. A new convex polytope $\mathrm{pol}(\tilde{\mathcal{P}}; s)$ defined in the opposite side of the boundary is shown in yellow. (d) A strictly better solution exists in $\mathrm{pol}(\tilde{\mathcal{P}}; s)$.

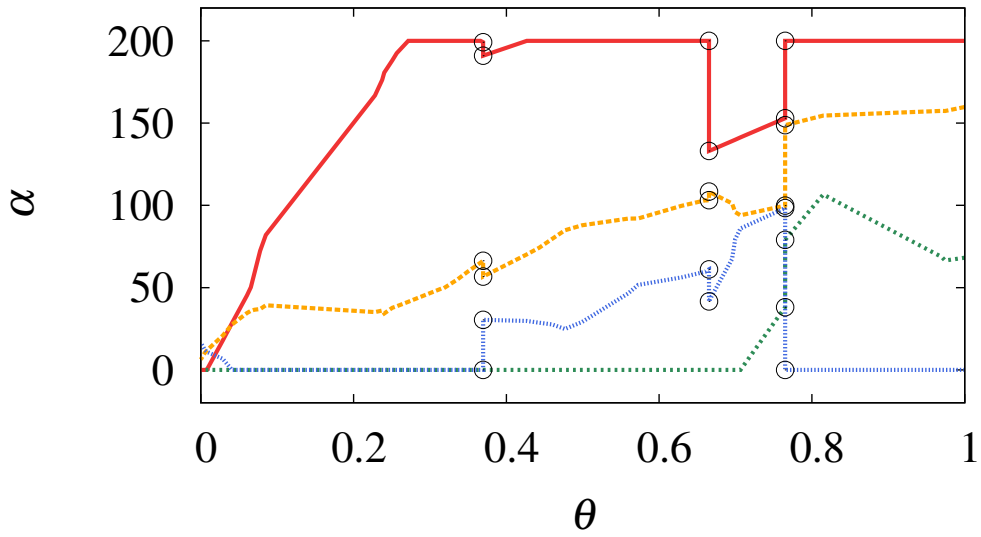


Figure 16: An example of the local solution path by OP-θ on a simple toy data set (with $C = 200$). The paths of four Lagrange multipliers $\alpha_1^*, \cdots, \alpha_4^*$ are plotted in the range of $\theta \in [0, 1]$. Open circles represent the discontinuous points in the path. In this simple example, we had experienced three discontinuous points at $\theta = 0.37, 0.67$ and 0.77.

12 Numerical Experiments

In this section, we compare the proposed outlier-path (OP) algorithm with conventional concave-convex procedure (CCCP) [52] because, in most of the existing robust SVM studies, non-convex optimization for robust SVM training are solved by CCCP or a variant called difference of convex (DC) programming [37, 38, 39, 40, 42, 43].

12.1 Setup

We used several benchmark data sets listed in Tables 5 and 6. We randomly divided data set into training (40%), validation (30%), and test (30%) sets for the purposes of optimization, model selection (including the selection of θ or s), and performance evaluation, respectively. For robust SVC, we randomly flipped 15%, 20%, 25% of the labels in the training and the validation data sets. For robust SVR, we first preprocess the input and output variables; each input variable was normalized so that the minimum and the maximum values are -1 and $+1$, respectively, while the output variable was standardized to have mean zero and variance one. Then, for the 5%, 10%, 15% of the training and the validation instances, we added an uniform noise $U(-2, 2)$ to input variable, and a Gaussian noise $N(0, 10^2)$ to output variable, where $U(a, b)$ denotes the uniform distribution between a and b and $N(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2.

12.2 Generalization Performance

First, we compared the generalization performance. We used the linear kernel and the radial basis function (RBF) kernel defined as $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)$, where γ is a kernel parameter fixed to $\gamma = 1/d$ with d being the input dimensionality. Model selection was carried out by finding the best hyper-parameter combination that minimizes the validation error. We have a pair of hyper-parameters in each setup. In all the setups, the regularization parameter C was chosen from $\{0.01, 0.1, 1, 10, 100\}$, while the candidates of the homotopy parameters were chosen as follows:

- In OP-θ, the set of break-points $\theta_{BP} \in [0, 1]$ was considered as the candidates (note that

Table 5: Benchmark data sets for robust SVC experiments

Data		n	d
D1	BreastCancerDiagnostic	569	30
D2	AustralianCredit	690	14
D3	GermanNumer	1000	24
D4	SVMGuide1	3089	4
D5	spambase	4601	57
D6	musk	6598	166
D7	gisette	6000	5000
D8	w5a	9888	300
D9	a6a	11220	122
D10	a7a	16100	122

$n = \#$ of instances, $d =$ input dimension

Table 6: Benchmark data sets for robust SVR experiments

Data		n	d
D1	bodyfat	252	14
D2	yacht_hydrodynamics	308	6
D3	mpg	392	7
D4	housing	506	13
D5	mg	1385	6
D6	winequality-red	1599	11
D7	winequality-white	4898	11
D8	space_ga	3107	6
D9	abalone	4177	8
D10	cpusmall	8192	12
D11	cadata	20640	8

$n = \#$ of instances, $d =$ input dimension

the local solutions at each break-point have been already computed in the homotopy computation).

- In OP-s, the set of break-points in $[s_C, 0]$ was used as the candidates for robust SVC, where

$$s_C := \min_{i \in \mathbb{N}_n} y_i f_{\mathrm{SVC}}(\boldsymbol{x}_i)$$

with f_{SVC} being the ordinary non-robust SVC. For robust SVR, the set of break-points in $[s_R, 0.2S_R]$ was used as the candidates, where

$$s_R := \max_{i \in \mathbb{N}_n} |y_i - f_{\mathrm{SVR}}(\boldsymbol{x}_i)|$$

with f_{SVC} being the ordinary non-robust SVR.

- In CCCP-θ, the homotopy parameter θ was selected from

$$\theta \in \{1, 0.75, 0.5, 0.25, 0\}.$$

- In CCCP-s, the homotopy parameter s was selected from

$$s \in \{s_C, 0.75s_C, 0.5s_C, 0.25s_C, 0\}$$

for robust SVC, while it was selected from

$$s \in \{s_R, 0.8s_R, 0.6s_R, 0.4s_R, 0.2s_R\}$$

for robust SVR.

Note that both OP and CCCP were initialized by using the solution of standard SVM.

Table 7 represents the average and the standard deviation of the test errors on 10 different random data splits. The other results on different noise levels summarized in § 16. These results indicate that our proposed OP algorithm tends to find better local solutions and the degree of robustness was appropriately controlled.

12.3 Computation Time

Second, we compared the computational costs of the entire model-building process of each method. The results are shown in Figure 17. Note that the computational cost of the OP algorithm does not depend on the number of hyper-parameter candidates of θ or s, because the entire path of local solutions has already been computed with the infinitesimal resolution in the homotopy computation. On the other hand, the computational cost of CCCP depends on the number of hyper-parameter candidates. In our implementation of CCCP, we used the warm-start approach, i.e., we initialized CCCP with the previous solution for efficiently

computing a sequence of solutions. The results indicate that the proposed OP algorithm enables stable and efficient control of robustness, while CCCP suffers a trade-off between model selection performance and computational costs.

12.4 Stability of Concave-Convex Procedure (CCCP)

Finally, we empirically investigate the stability of CCCP algorithm. For simplicity, we only considered a linear classification problem on BreastCancerDiagnostic data with the homotopy parameters $\theta = 0$ and $s = 0$. Remember that, $\theta = 1$ corresponds to the hinge loss function for the standard convex SVM, while $\theta = 0$ corresponds to fully robust ramp loss function. Here, we used warm-start approaches for obtaining a robust SVM solution for $\theta = 0$ by considering a sequence of θ values: $\theta_{(\ell)} := 1 - \frac{\ell}{L}, \ell = 0, 1, \ldots, L$ (another homotopy parameter s was fixed to be 0). Specifically, we started from the standard convex SVM solution with $\theta_{(0)} = 1$, and used it as a warm-start initial solution for $\theta_{(1)}$, and used it as a warm-start initial solution for $\theta_{(2)}$, and so on.

Table 8 shows the average and the standard deviation of the test errors on 30 different random data splits for the three warm-start approaches with $L = 5, 10, 15$ (the performances of CCCP in Table 7 are the results with $L = 5$). The results indicate that the performances (for $\theta = 0$) were slightly better, i.e., better local optimal solutions were obtained when the length of the sequence L is larger. We conjecture that it is because the warm-start approach can be regarded as a simulated annealing, and the performances were better when the annealing step size ($= 1/L$) is smaller.

It is important to note that, if we consider the above warm-start approach with very large L, we would be able to obtain similar results as the proposed homotopy approach. In other words, the proposed homotopy approach can be considered as an efficient method for conducting simulated annealing with infinitesimally small annealing step size. Table 9 shows the same results as Table 7 for three different CCCP approaches with $L = 5, 10, 15$. As we discussed above, the performances tend to be better when L is large, although the computational cost also increases when L gets large.

Table 7: The mean of test error by 0-1 loss and standard deviation (linear, robust SVC). The noise level is 15% in SVC, while it is 5% in SVR. The bold face indicate the better method in terms of the test error.

Data	C-SVC	CCCP-θ	OP-θ	CCCP-s	OP-s
D1	.056(.016)	.050(.014)	**.049(.016)**	.055(.018)	**.050(.016)**
D2	.151(.018)	**.145(.007)**	.151(.018)	**.145(.007)**	.152(.010)
D3	.281(.028)	.270(.033)	.270(.023)	**.262(.013)**	.266(.013)
D4	.066(.007)	.047(.007)	.047(.005)	.053(.010)	**.042(.006)**
D5	.108(.010)	.088(.009)	.088(.009)	.088(.010)	**.084(.007)**
D6	.072(.005)	**.058(.006)**	.064(.003)	.061(.007)	**.060(.003)**
D7	.185(.013)	.184(.010)	.184(.010)	.184(.010)	.184(.010)
D8	.020(.002)	.020(.003)	.020(.002)	.021(.003)	**.020(.003)**
D9	.173(.004)	.181(.009)	**.173(.005)**	.165(.004)	**.164(.004)**
D10	.173(.008)	.176(.006)	**.173(.007)**	**.160(.004)**	.161(.005)

The mean of test error by 0-1 loss and standard deviation (RBF, robust SVC).

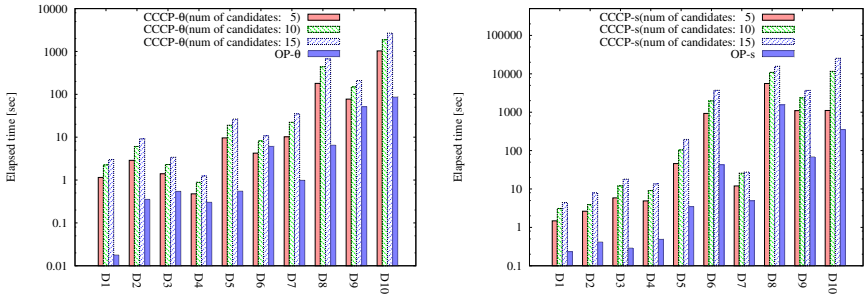
D1	.055(.017)	.043(.022)	**.042(.017)**	**.037(.016)**	.038(.013)
D2	.149(.010)	.148(.010)	**.147(.010)**	.146(.013)	**.142(.013)**
D3	.276(.024)	.267(.026)	**.266(.024)**	.271(.015)	**.261(.020)**
D4	.052(.009)	.048(.009)	**.044(.006)**	.047(.008)	**.040(.005)**
D5	.117(.012)	.109(.013)	**.107(.012)**	.107(.011)	**.094(.008)**
D6	.046(.007)	.045(.007)	.045(.007)	.045(.007)	**.043(.006)**
D7	.044(.003)	.044(.003)	.044(.003)	.044(.003)	.044(.003)
D8	.022(.003)	.022(.003)	.022(.003)	.022(.003)	**.021(.002)**
D9	.169(.003)	.170(.005)	**.169(.004)**	.168(.005)	**.162(.003)**
D10	.163(.003)	.163(.003)	.163(.003)	.162(.002)	**.160(.004)**

The mean of L_1 test error and standard deviation (linear, robust SVR).

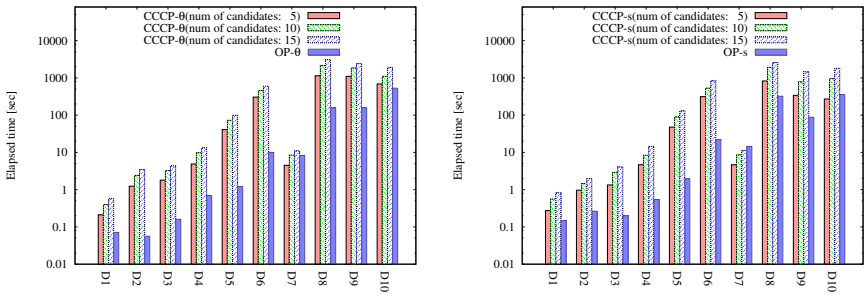
D1	.442(.324)	.337(.347)	**.319(.353)**	.414(.341)	**.276(.321)**
D2	.470(.053)	.487(.086)	**.474(.087)**	.490(.108)	**.484(.104)**
D3	.414(.038)	.351(.025)	**.350(.036)**	.414(.105)	**.372(.043)**
D4	.548(.180)	.520(.193)	**.510(.146)**	**.562(.210)**	.596(.297)
D5	.539(.019)	.531(.019)	**.530(.017)**	.539(.024)	**.529(.018)**
D6	.685(.028)	.664(.026)	**.655(.027)**	**.685(.044)**	.686(.040)
D7	.700(.016)	.691(.017)	**.685(.017)**	.698(.022)	**.692(.014)**
D8	.582(.027)	.583(.042)	**.570(.031)**	.589(.035)	**.569(.028)**
D9	.518(.015)	.510(.019)	**.501(.021)**	.522(.026)	**.516(.019)**
D10	.281(.021)	**.278(.016)**	.279(.016)	.269(.018)	.269(.021)
D11	.494(.010)	.488(.011)	**.487(.012)**	.492(.009)	.492(.008)

The mean of L_1 test error and standard deviation (RBF, robust SVR).

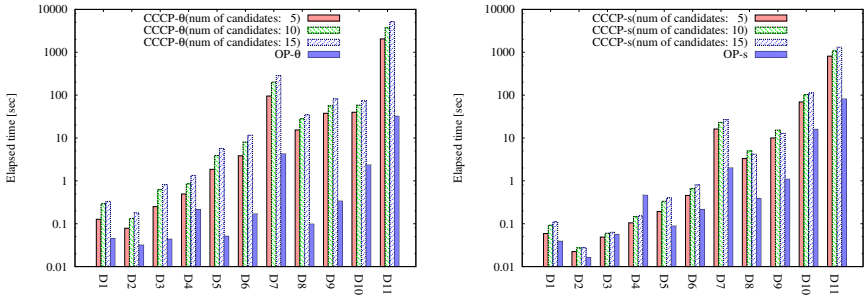
D1	.077(.049)	.069(.054)	**.065(.056)**	.070(.053)	**.051(.040)**
D2	.357(.059)	.346(.045)	**.339(.045)**	.332(.038)	**.327(.040)**
D3	.337(.052)	**.299(.021)**	.302(.019)	.296(.022)	**.295(.022)**
D4	.390(.046)	.350(.025)	**.349(.023)**	.357(.022)	**.343(.024)**
D5	.513(.024)	.519(.028)	**.504(.018)**	.515(.024)	**.503(.019)**
D6	.641(.028)	.640(.015)	**.635(.017)**	.634(.022)	**.631(.017)**
D7	.671(.011)	.669(.009)	.669(.007)	.674(.011)	**.671(.009)**
D8	.528(.027)	.504(.027)	**.496(.024)**	.511(.018)	**.510(.020)**
D9	.488(.012)	.490(.016)	**.486(.012)**	.484(.013)	**.482(.014)**
D10	.198(.015)	.198(.027)	**.196(.025)**	.194(.015)	**.189(.017)**
D11	.456(.016)	.441(.005)	.441(.006)	**.444(.015)**	.446(.015)



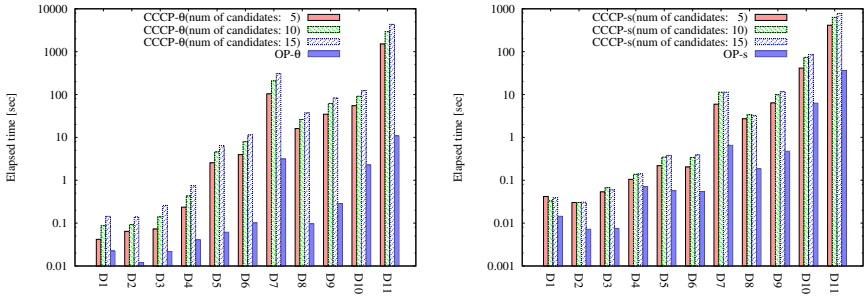
(a) Elapsed time for CCCP and OP (linear, robust SVC)



(b) Elapsed time for CCCP and OP (RBF, robust SVC)



(c) Elapsed time for CCCP and OP (linear, robust SVR)



(d) Elapsed time for CCCP and OP (RBF, robust SVR)

Figure 17: Elapsed time when the number of (θ, s)-candidates is increased. Changing the number of hyper-parameter candidates affects the computation time of CCCP, but not OP because the entire path of solutions is computed with the infinitesimal resolution.

Table 8: The mean of test error by 0-1 loss and standard deviation when the number of hyperparameter candidates is increased.

$\theta(L=5)$	test error	$\theta(L=10)$	test error	$\theta(L=15)$	test error
.800	.051(.015)	.900	.052(.016)	.933	.052(.016)
.600	.047(.014)	.800	.051(.015)	.867	.052(.015)
.400	.047(.013)	.700	.049(.015)	.800	.051(.015)
.200	.043(.015)	.600	.048(.014)	.733	.050(.014)
.000	.043(.013)	.500	.047(.014)	.667	.049(.014)
		.400	.046(.013)	.600	.048(.014)
		.300	.044(.015)	.533	.048(.015)
		.200	.043(.013)	.467	.047(.013)
		.100	.041(.013)	.400	.046(.013)
		.000	.041(.012)	.333	.045(.014)
				.267	.043(.012)
				.200	.042(.013)
				.133	.041(.012)
				.067	.040(.012)
				.000	.040(.013)

Table 9: The mean of test error by 0-1 loss and standard deviation (linear, robust SVC). The noise level is 15% in SVC, while it is 5% in SVR. The bold face indicate the better than CCCP-θ^5 or CCCP-s^5.

Data	CCCP-θ^5	CCCP-θ^{10}	CCCP-θ^{15}	OP-θ	CCCP-s^5	CCCP-s^{10}	CCCP-s^{15}	OP-s
D1	.050(.014)	**.049(.019)**	**.048(.017)**	**.049(.016)**	.055(.018)	**.045(.012)**	**.046(.009)**	**.050(.016)**
D2	.145(.007)	.151(.019)	.151(.019)	.151(.018)	.145(.007)	.154(.018)	.154(.018)	.152(.010)
D3	.270(.033)	**.265(.014)**	**.266(.019)**	.270(.023)	.262(.013)	.277(.023)	.270(.020)	.266(.013)
D4	.047(.007)	.048(.005)	.049(.005)	.047(.005)	.053(.010)	**.046(.004)**	**.046(.004)**	**.042(.006)**
D5	.088(.009)	**.087(.007)**	.088(.009)	.088(.009)	.088(.010)	**.087(.012)**	**.086(.010)**	**.084(.007)**
D6	.058(.006)	.065(.005)	.065(.005)	.064(.003)	.061(.007)	.061(.004)	**.060(.003)**	**.060(.003)**
D7	.184(.010)	.184(.010)	.184(.010)	.184(.010)	.184(.010)	.184(.009)	.184(.008)	.184(.010)
D8	.020(.003)	.020(.002)	.020(.002)	.020(.002)	.021(.003)	**.020(.002)**	**.020(.002)**	**.020(.003)**
D9	.181(.009)	**.173(.004)**	**.172(.004)**	**.173(.005)**	.165(.004)	**.162(.003)**	**.163(.003)**	**.164(.004)**
D10	.176(.006)	**.175(.006)**	**.174(.007)**	**.173(.007)**	.160(.004)	.160(.003)	.162(.003)	.161(.005)
The mean of test error by 0-1 loss and standard deviation (RBF, robust SVC).								
D1	.043(.022)	.043(.013)	**.042(.015)**	**.042(.017)**	.037(.016)	.047(.016)	.048(.014)	.038(.013)
D2	.148(.010)	.152(.019)	.152(.020)	**.147(.010)**	.146(.013)	.146(.021)	.146(.021)	**.142(.013)**
D3	.267(.026)	.271(.022)	.269(.026)	**.266(.024)**	.271(.015)	**.268(.015)**	**.268(.016)**	**.261(.020)**
D4	.048(.009)	**.045(.006)**	**.045(.005)**	**.044(.006)**	.047(.008)	**.044(.004)**	**.044(.006)**	**.040(.005)**
D5	.109(.013)	.110(.010)	.110(.010)	**.107(.012)**	.107(.011)	**.097(.009)**	**.099(.009)**	**.094(.008)**
D6	.045(.007)	.051(.004)	.051(.004)	.045(.007)	.045(.007)	.049(.004)	.051(.005)	**.043(.006)**
D7	.044(.003)	.044(.003)	.044(.003)	.044(.003)	.044(.003)	.044(.003)	.044(.003)	.044(.003)
D8	.022(.003)	**.021(.002)**	**.021(.002)**	.022(.003)	.022(.003)	.022(.003)	**.021(.003)**	**.021(.002)**
D9	.170(.005)	**.169(.003)**	**.169(.003)**	**.169(.004)**	.168(.005)	**.164(.003)**	**.165(.004)**	**.162(.003)**
D10	.163(.003)	.163(.003)	.163(.003)	.163(.003)	.162(.002)	**.160(.004)**	**.161(.004)**	**.160(.004)**
The mean of L_1 test error and standard deviation (linear, robust SVR).								
D1	.337(.347)	.343(.349)	**.317(.329)**	**.319(.353)**	.414(.341)	.421(.346)	**.400(.371)**	**.276(.321)**
D2	.487(.086)	**.486(.083)**	**.477(.080)**	**.474(.087)**	.490(.108)	**.489(.110)**	**.489(.110)**	**.484(.104)**
D3	.351(.025)	.356(.030)	.354(.027)	**.350(.036)**	.414(.105)	**.405(.085)**	**.413(.083)**	**.372(.043)**
D4	.520(.193)	**.484(.117)**	.532(.270)	**.510(.146)**	.562(.210)	**.539(.128)**	**.559(.193)**	.596(.297)
D5	.531(.019)	.534(.020)	.532(.013)	**.530(.017)**	.539(.024)	**.533(.018)**	**.537(.011)**	**.529(.018)**
D6	.664(.026)	**.662(.028)**	.668(.025)	**.655(.027)**	.685(.044)	**.676(.034)**	**.682(.037)**	.686(.040)
D7	.691(.017)	**.688(.016)**	**.687(.016)**	**.685(.017)**	.698(.022)	**.693(.013)**	.700(.021)	**.692(.014)**
D8	.583(.042)	**.573(.028)**	**.580(.039)**	**.570(.031)**	.589(.035)	**.581(.032)**	**.582(.037)**	**.569(.028)**
D9	.510(.019)	**.506(.022)**	.510(.022)	**.501(.021)**	.522(.026)	**.518(.025)**	.523(.023)	**.516(.019)**
D10	.278(.016)	.279(.016)	.283(.019)	.279(.016)	.269(.018)	.274(.020)	.271(.020)	.269(.021)
D11	.488(.011)	.489(.010)	**.487(.010)**	**.487(.012)**	.492(.009)	**.491(.010)**	.492(.009)	.492(.008)
The mean of L_1 test error and standard deviation (RBF, robust SVR).								
D1	.069(.054)	**.066(.056)**	**.061(.052)**	**.065(.056)**	.070(.053)	**.069(.054)**	.070(.055)	**.051(.040)**
D2	.346(.045)	**.341(.044)**	**.341(.044)**	**.339(.045)**	.332(.038)	**.324(.026)**	**.324(.026)**	**.327(.040)**
D3	.299(.021)	.299(.019)	.300(.020)	.302(.019)	.296(.022)	.296(.020)	**.295(.022)**	**.295(.022)**
D4	.350(.025)	**.348(.024)**	**.346(.025)**	**.349(.023)**	.357(.022)	**.356(.022)**	.358(.023)	**.343(.024)**
D5	.519(.028)	**.508(.019)**	**.501(.019)**	**.504(.018)**	.515(.024)	**.506(.021)**	**.506(.021)**	**.503(.019)**
D6	.640(.015)	**.637(.020)**	.643(.021)	**.635(.017)**	.634(.022)	.634(.020)	**.633(.019)**	**.631(.017)**
D7	.669(.009)	.669(.008)	**.668(.008)**	.669(.007)	.674(.011)	.675(.010)	**.672(.010)**	**.671(.009)**
D8	.504(.027)	.504(.028)	**.500(.025)**	**.496(.024)**	.511(.018)	.514(.019)	.512(.017)	**.510(.020)**
D9	.490(.016)	**.487(.013)**	**.486(.010)**	**.486(.012)**	.484(.013)	.486(.013)	.484(.014)	**.482(.014)**
D10	.198(.027)	**.189(.022)**	**.196(.023)**	**.196(.025)**	.194(.015)	**.191(.017)**	**.192(.015)**	**.189(.017)**
D11	.441(.005)	.441(.005)	**.440(.006)**	.441(.006)	.444(.015)	.445(.015)	.448(.016)	.446(.015)

13 Conclusion

In this paper, we proposed a novel robust SVM learning algorithm based on the homotopy approach that allows efficient computation of the sequence of local optimal solutions when the influence of outliers is gradually decreased. The algorithm is built on our theoretical findings about the geometric property and the optimality conditions of local solutions of robust SVM. Experimental results indicate that our algorithm tends to find better local solutions possibly due to the simulated annealing-like effect and the stable control of robustness. One of the important future works is to adopt scalable homotopy algorithms [55] or approximate parametric programming algorithms [60] for further improving the computational efficiency.

14 Proofs

14.1 Proof of Lemma 5

The proof can be directly derived based on standard convex optimization theory [61] because the optimization problem (32) is convex if the partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\}$ is fixed.

We denote a n dimensional vector as $\mathbf{1}, \mathbf{1}_\mathcal{I}$ or $\mathbf{1}_\mathcal{O} \in \mathbb{R}^n$: the all elements of $\mathbf{1}$ are one, while $\mathbf{1}_\mathcal{I}$ or $\mathbf{1}_\mathcal{O}$ indicates the all elements corresponding to $\mathcal{I}$ or $\mathcal{O}$ are one and the others are zero, and it can be obtained as $\mathbf{1} := \mathbf{1}_\mathcal{I} + \mathbf{1}_\mathcal{O}$. In this proof, we use this notation for any vectors such as $\boldsymbol{\alpha} := \boldsymbol{\alpha}_\mathcal{I} + \boldsymbol{\alpha}_\mathcal{O}$ and $\boldsymbol{y} := \boldsymbol{y}_\mathcal{I} + \boldsymbol{y}_\mathcal{O}$.

We rewrite the optimization problem in the conditionally optimal solution (32) as

$$
\begin{aligned}
&\arg\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i\in\mathcal{I}}[1 - y_i f(\boldsymbol{x}_i)]_+ + C\theta\sum_{i\in\mathcal{O}}[1 - y_i f(\boldsymbol{x}_i)]_+ \\
:= &\arg\min_{\boldsymbol{w},\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\mathbf{1}_\mathcal{I}^\top\boldsymbol{\xi}_\mathcal{I} + C\theta\mathbf{1}_\mathcal{O}^\top\boldsymbol{\xi}_\mathcal{O} \ \text{ subject to} \\
&\qquad \mathbf{1}_\mathcal{I} - \boldsymbol{y}_\mathcal{I} \circ (\boldsymbol{\Phi}\boldsymbol{w})_\mathcal{I} \leq \boldsymbol{\xi}_\mathcal{I},\ \boldsymbol{\xi}_\mathcal{I} \geq \mathbf{0},\ \boldsymbol{y}_\mathcal{I} \circ (\boldsymbol{\Phi}\boldsymbol{w})_\mathcal{I} \geq s\mathbf{1}_\mathcal{I}, \\
&\qquad \mathbf{1}_\mathcal{O} - \boldsymbol{y}_\mathcal{O} \circ (\boldsymbol{\Phi}\boldsymbol{w})_\mathcal{O} \leq \boldsymbol{\xi}_\mathcal{O},\ \boldsymbol{\xi}_\mathcal{O} \geq \mathbf{0},\ \boldsymbol{y}_\mathcal{O} \circ (\boldsymbol{\Phi}\boldsymbol{w})_\mathcal{O} \leq s\mathbf{1}_\mathcal{O},
\end{aligned}
$$

where $\boldsymbol{\Phi}\boldsymbol{w}$ represents the decision function $f_\mathcal{P}(\boldsymbol{x}_i) := (\boldsymbol{\Phi}\boldsymbol{w})_i$ as defined (25), and $\circ$ indicates the element-wise product of two vectors.

Let us define Lagrangian as

$$L := \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\mathbf{1}_{\mathcal{I}}^\top \boldsymbol{\xi}_{\mathcal{I}} + C\theta \mathbf{1}_{\mathcal{O}}^\top \boldsymbol{\xi}_{\mathcal{O}} + (\boldsymbol{\alpha}_{\mathcal{I}} + \boldsymbol{\alpha}_{\mathcal{O}})^\top (\mathbf{1} - \boldsymbol{y} \circ (\boldsymbol{\Phi}\boldsymbol{w}) - \boldsymbol{\xi}) - (\boldsymbol{\eta}_{\mathcal{I}} + \boldsymbol{\eta}_{\mathcal{O}})^\top \boldsymbol{\xi} - (\boldsymbol{\nu}_{\mathcal{I}} - \boldsymbol{\nu}_{\mathcal{O}})^\top (\boldsymbol{y} \circ (\boldsymbol{\Phi}\boldsymbol{w}) - s\mathbf{1}).$$

Using the convex optimization theory [61], the optimality conditions are

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{w}} &= \boldsymbol{w}^* - \boldsymbol{\Phi}^\top (\boldsymbol{y} \circ (\boldsymbol{\alpha}^* + \boldsymbol{\nu}_{\mathcal{I}}^* - \boldsymbol{\nu}_{\mathcal{O}}^*)) = \mathbf{0}, \\
\frac{\partial L}{\partial \boldsymbol{\xi}_I} = C\mathbf{1}_{\mathcal{I}} - \boldsymbol{\alpha}_{\mathcal{I}}^* - \boldsymbol{\eta}_{\mathcal{I}}^* = \mathbf{0},\ \frac{\partial L}{\partial \boldsymbol{\xi}_O} &= C\theta\mathbf{1}_{\mathcal{O}}^* - \boldsymbol{\alpha}_{\mathcal{O}}^* - \boldsymbol{\eta}_{\mathcal{O}}^* = \mathbf{0}, \\
\mathbf{1}_{\mathcal{I}} - \boldsymbol{y}_{\mathcal{I}} \circ (\boldsymbol{\Phi}\boldsymbol{w}^*)_{\mathcal{I}} \leq \boldsymbol{\xi}_{\mathcal{I}}^*,\ \boldsymbol{\xi}_{\mathcal{I}}^* \geq \mathbf{0},\ \boldsymbol{y}_{\mathcal{I}} &\circ (\boldsymbol{\Phi}\boldsymbol{w}^*)_{\mathcal{I}} \geq s\mathbf{1}_{\mathcal{I}}, \\
\mathbf{1}_{\mathcal{O}} - \boldsymbol{y}_{\mathcal{O}} \circ (\boldsymbol{\Phi}\boldsymbol{w}^*)_{\mathcal{O}} \leq \boldsymbol{\xi}_{\mathcal{O}}^*,\ \boldsymbol{\xi}_{\mathcal{O}}^* \geq \mathbf{0},\ \boldsymbol{y}_{\mathcal{O}} &\circ (\boldsymbol{\Phi}\boldsymbol{w}^*)_{\mathcal{O}} \leq s\mathbf{1}_{\mathcal{O}}, \\
\boldsymbol{\alpha}_{\mathcal{I}}^* \geq \mathbf{0},\ \boldsymbol{\alpha}_{\mathcal{O}}^* \geq \mathbf{0},\ \boldsymbol{\eta}_{\mathcal{I}}^* \geq \mathbf{0},\ \boldsymbol{\eta}_{\mathcal{O}}^* \geq \mathbf{0},\ \boldsymbol{\nu}_{\mathcal{I}}^* &\geq \mathbf{0},\ \boldsymbol{\nu}_{\mathcal{O}}^* \geq \mathbf{0}, \\
\boldsymbol{\alpha}_{\mathcal{I}}^{*\top} (\mathbf{1}_{\mathcal{I}} - \boldsymbol{y}_{\mathcal{I}} \circ (\boldsymbol{\Phi}\boldsymbol{w}^*)_{\mathcal{I}} - \boldsymbol{\xi}_{\mathcal{I}}^*) = \mathbf{0},\ \boldsymbol{\eta}_{\mathcal{I}}^* \circ \boldsymbol{\xi}_{\mathcal{I}}^* &= \mathbf{0}, \\
\boldsymbol{\alpha}_{\mathcal{O}}^{*\top} (\mathbf{1}_{\mathcal{O}} - \boldsymbol{y}_{\mathcal{O}} \circ (\boldsymbol{\Phi}\boldsymbol{w}^*)_{\mathcal{O}} - \boldsymbol{\xi}_{\mathcal{O}}^*) = \mathbf{0},\ \boldsymbol{\eta}_{\mathcal{O}}^* \circ \boldsymbol{\xi}_{\mathcal{O}}^* &= \mathbf{0}, \\
\boldsymbol{\nu}_{\mathcal{I}}^{*\top} (\boldsymbol{y}_{\mathcal{I}} \circ (\boldsymbol{\Phi}\boldsymbol{w}^*)_{\mathcal{I}} - s\mathbf{1}_{\mathcal{I}}) &= \mathbf{0}, \\
\boldsymbol{\nu}_{\mathcal{O}}^{*\top} (\boldsymbol{y}_{\mathcal{O}} \circ (\boldsymbol{\Phi}\boldsymbol{w}^*)_{\mathcal{O}} - s\mathbf{1}_{\mathcal{O}}) &= \mathbf{0}.
\end{aligned}$$

We rewrite the multipliers $\boldsymbol{\alpha}^*$ as $\boldsymbol{\alpha}^* \leftarrow \boldsymbol{\alpha}^* + \boldsymbol{\nu}_{\mathcal{I}}^* - \boldsymbol{\nu}_{\mathcal{O}}^*$, and then we obtain $\boldsymbol{w}^* = \boldsymbol{\Phi}^\top (\boldsymbol{y} \circ \boldsymbol{\alpha}^*)$. Since both $\boldsymbol{\nu}_{\mathcal{I}}^*$ and $\boldsymbol{\nu}_{\mathcal{O}}^*$ are zero when $\boldsymbol{y} \circ (\boldsymbol{\Phi}\boldsymbol{w}^*) \neq s\mathbf{1}$, the optimality conditions can be rewritten as

$$\begin{aligned}
y_i f_{\mathcal{P}}^*(\boldsymbol{x}_i) > 1 \ &\Rightarrow\ \alpha_i^* = 0, \\
y_i f_{\mathcal{P}}^*(\boldsymbol{x}_i) = 1 \ &\Rightarrow\ \alpha_i^* \in [0, C], \\
s < y_i f_{\mathcal{P}}^*(\boldsymbol{x}_i) < 1 \ &\Rightarrow\ \alpha_i^* = C, \\
y_i f_{\mathcal{P}}^*(\boldsymbol{x}_i) < s \ &\Rightarrow\ \alpha_i^* = C\theta,
\end{aligned}$$

while $\boldsymbol{\nu}_i^*$ can be positive when $\boldsymbol{y}_i (\boldsymbol{\Phi}\boldsymbol{w}^*)_i = s$, and thus the following holds:

$$\begin{aligned}
y_i f_{\mathcal{P}}^*(\boldsymbol{x}_i) = s, i \in \mathcal{I} \ &\Rightarrow\ \alpha_i^* \geq C, \\
y_i f_{\mathcal{P}}^*(\boldsymbol{x}_i) = s, i \in \mathcal{O} \ &\Rightarrow\ \alpha_i^* \leq C\theta,
\end{aligned}$$

we finalized the proof. **Q.E.D.**

14.2 Proof of Theorem 6

Although $f^*_{\mathcal{P}}$ is a feasible solution, it is not a local optimum for $\theta \in [0, 1)$ and $s \leq 0$ because

$$\alpha_i \leq C\theta \quad \text{for} \quad i \in \tilde{\mathcal{I}} \cap \mathcal{O}, \tag{44a}$$

$$\alpha_i \geq C \quad \text{for} \quad i \in \tilde{\mathcal{O}} \cap \mathcal{I}, \tag{44b}$$

violate the KKT conditions (34) for $\tilde{\mathcal{P}}$. These feasibility and *sub*-optimality indicates that

$$J_{\tilde{\mathcal{P}}}(f^*_{\tilde{\mathcal{P}}}; \theta) < J_{\mathcal{P}}(f^*_{\mathcal{P}}; \theta), \tag{45}$$

we arrive at (36). **Q.E.D.**

14.3 Proof of Theorem 7

Sufficiency: If (37e) is true, i.e., if there are NO instances with $y_i f^*_{\mathcal{P}}(\vec{x}_i) = s$, then any convex problems defined by different partitions $\tilde{\mathcal{P}} \neq \mathcal{P}$ do not have feasible solutions in the neighborhood of $f^*_{\mathcal{P}}$. This means that if $f^*_{\mathcal{P}}$ is a conditionally optimal solution, then it is locally optimal. (37a)-(37d) are sufficient for $f^*_{\mathcal{P}}$ to be conditionally optimal for the given partition $\mathcal{P}$. Thus, (37) is sufficient for $f^*_{\mathcal{P}}$ to be locally optimal.

Necessity: From Theorem 6, if there exists an instance such that $y_i f^*_{\mathcal{P}}(\vec{x}_i) = s$, then $f^*_{\mathcal{P}}$ is a feasible but not locally optimal. Then (37e) is necessary for $f^*_{\mathcal{P}}$ to be locally optimal. In addition, (37a)-(37d) are also necessary for local optimality, because of every local optimal solutions are conditionally optimal for the given partition $\mathcal{P}$. Thus, (37) is necessary for $f^*_{\mathcal{P}}$ to be locally optimal.

Q.E.D.

15 Implementation of D-step

In D-step, we work with the following convex problem

$$f^*_{\tilde{\mathcal{P}}} := \arg \min_{f \in \mathrm{pol}(\tilde{\mathcal{P}}; s)} J_{\tilde{\mathcal{P}}}(f; \theta). \tag{46}$$

where, $\tilde{\mathcal{P}}$ is updated from $\mathcal{P}$ as (35).

Let us define a partition $\Pi := \{\mathcal{R}, \mathcal{E}, \mathcal{L}, \tilde{\mathcal{I}}', \tilde{\mathcal{O}}', \hat{\mathcal{O}}''\}$ of $\mathbb{N}_n$ such that

$$
\begin{aligned}
i \in \mathcal{R} &\;\Rightarrow\; y_i f(\boldsymbol{x}_i) > 1, && \text{(47a)}\\
i \in \mathcal{E} &\;\Rightarrow\; y_i f(\boldsymbol{x}_i) = 1, && \text{(47b)}\\
i \in \mathcal{L} &\;\Rightarrow\; s < y_i f(\boldsymbol{x}_i) < 1, && \text{(47c)}\\
i \in \tilde{\mathcal{I}}' &\;\Rightarrow\; y_i f(\boldsymbol{x}_i) = s \text{ and } i \in \tilde{\mathcal{I}}, && \text{(47d)}\\
i \in \tilde{\mathcal{O}}' &\;\Rightarrow\; y_i f(\boldsymbol{x}_i) = s \text{ and } i \in \tilde{\mathcal{O}}, && \text{(47e)}\\
i \in \tilde{\mathcal{O}}'' &\;\Rightarrow\; y_i f(\boldsymbol{x}_i) < s. && \text{(47f)}
\end{aligned}
$$

If we write the conditionally optimal solution as

$$
f^*_{\tilde{\mathcal{P}}}(x) := \sum_{j \in \mathbb{N}_n} \alpha^*_j y_j K(x, \boldsymbol{x}_j), \tag{48}
$$

$\{\alpha^*_j\}_{j \in \mathbb{N}_n}$ must satisfy the following KKT conditions

$$
\begin{aligned}
y_i f^*_{\tilde{\mathcal{P}}}(\vec{x}_i) > 1 &\;\Rightarrow\; \alpha^*_i = 0, && \text{(49a)}\\
y_i f^*_{\tilde{\mathcal{P}}}(\vec{x}_i) = 1 &\;\Rightarrow\; \alpha^*_i \in [0, C], && \text{(49b)}\\
s < y_i f^*_{\tilde{\mathcal{P}}}(\vec{x}_i) < 1 &\;\Rightarrow\; \alpha^*_i = C, && \text{(49c)}\\
y_i f^*_{\tilde{\mathcal{P}}}(\vec{x}_i) = s, i \in \tilde{\mathcal{I}}' &\;\Rightarrow\; \alpha^*_i \geq C, && \text{(49d)}\\
y_i f^*_{\tilde{\mathcal{P}}}(\vec{x}_i) = s, i \in \tilde{\mathcal{O}}' &\;\Rightarrow\; \alpha^*_i \leq C\theta, && \text{(49e)}\\
y_i f^*_{\tilde{\mathcal{P}}}(\vec{x}_i) < s, i \in \tilde{\mathcal{O}}'' &\;\Rightarrow\; \alpha^*_i = C\theta. && \text{(49f)}
\end{aligned}
$$

At the beginning of the D-step, $f^*_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i)$ violates the KKT conditions by

$$
\Delta f_i := y_i \begin{bmatrix} \boldsymbol{K}_{i,\Delta_{\mathcal{I}\to\mathcal{O}}} & \boldsymbol{K}_{i,\Delta_{\mathcal{O}\to\mathcal{I}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}^{(\mathrm{bef})}_{\Delta_{\mathcal{I}\to\mathcal{O}}} - \mathbf{1}C\theta \\ \boldsymbol{\alpha}^{(\mathrm{bef})}_{\Delta_{\mathcal{O}\to\mathcal{I}}} - \mathbf{1}C \end{bmatrix}.
$$

where $\boldsymbol{\alpha}^{(\mathrm{bef})}$ is the corresponding $\boldsymbol{\alpha}$ at the beginning of the D-step, while $\Delta_{\mathcal{I}\to\mathcal{O}}$ and $\Delta_{\mathcal{O}\to\mathcal{I}}$ denote the difference in $\tilde{\mathcal{P}}$ and $\mathcal{P}$ defined as

$$
\begin{aligned}
\Delta_{\mathcal{I}\to\mathcal{O}} &:= \{i \in \mathcal{I} \mid y_i f_{\mathcal{P}}(\boldsymbol{x}_i) = s\},\\
\Delta_{\mathcal{O}\to\mathcal{I}} &:= \{i \in \mathcal{O} \mid y_i f_{\mathcal{P}}(\boldsymbol{x}_i) = s\}.
\end{aligned}
$$

Then, we consider the following another parameterized problem with a parameter $\mu \in [0, 1]$:

$$f_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i; \mu) := f_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i) + \mu \Delta f_i \ \forall \ i \in \mathbb{N}_n.$$

In order to always satisfy the KKT conditions for $f_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i; \mu)$, we solve the following linear system

$$\boldsymbol{Q}_{\mathcal{A},\mathcal{A}} \begin{bmatrix} \boldsymbol{\alpha}_{\mathcal{E}} \\ \boldsymbol{\alpha}_{\tilde{\mathcal{I}}'} \\ \boldsymbol{\alpha}_{\tilde{\mathcal{O}}'} \end{bmatrix} = \begin{bmatrix} \boldsymbol{1} \\ \boldsymbol{s} \\ \boldsymbol{s} \end{bmatrix} - \boldsymbol{Q}_{\mathcal{A},\mathcal{L}} \boldsymbol{1} C - \boldsymbol{Q}_{\mathcal{A},\tilde{\mathcal{O}}''} \boldsymbol{1} C \theta$$
$$- \begin{bmatrix} \boldsymbol{Q}_{\mathcal{A},\Delta_{\mathcal{I}\to\mathcal{O}}} & \boldsymbol{Q}_{\mathcal{A},\Delta_{\mathcal{O}\to\mathcal{I}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}^{(\mathrm{bef})}_{\Delta_{\mathcal{I}\to\mathcal{O}}} - \boldsymbol{1} C \theta \\ \boldsymbol{\alpha}^{(\mathrm{bef})}_{\Delta_{\mathcal{O}\to\mathcal{I}}} - \boldsymbol{1} C \end{bmatrix} \mu,$$

where $\mathcal{A} := \{\mathcal{E}, \tilde{\mathcal{I}}', \tilde{\mathcal{O}}'\}$. This linear system can also be solved by using the piecewise-linear parametric programming while the scalar parameter μ is continuously moved from 1 to 0.

In this parametric problem, we can show that $f^*_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i; \mu) = f^*_{\mathcal{P}}(\boldsymbol{x}_i)$ if $\mu = 1$ and $f^*_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i; \mu) = f^*_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i)$ if $\mu = 0$ for all $i \in \mathbb{N}_n$.

Since the number of elements in $\Delta_{\mathcal{I}\to\mathcal{O}}$ and $\Delta_{\mathcal{O}\to\mathcal{I}}$ are typically small, the D-step can be efficiently implemented by a technique used in the context of incremental learning [59].

16 Generalization Performance on Different Noise Levels

Tables 10 and 11 represent the average and the standard deviation of the test errors on 10 different random data splits with more higher noise level. It seems that our proposed OP algorithm tends to find better local solutions even if the noisy level is more higher.

Acknowledgments

I would like to express my deep sense of appreciation to my current supervisor Prof. Ichiro Takeuchi. He has been a great mentor throughout my PhD. Most of my publications that

are essential parts of this thesis have never completed without his help. In addition, I would like to give my thanks to all current and past members of Prof. Takeuchi lab. for their participation in constructing fruitful discussion and meeting.

In particular, I would like to thank Kazuya Nakagawa at Takeuchi lab., I have benefited from his numerous insights and countless discussions. Moreover, I would like to thank Assoc. Prof. Masayuki Karasuyama and Asst. Prof. Yuta Umezu at Nagoya Institute of Technology, Prof. Koji Tsuda and Prof. Masashi Sugiyama at University of Tokyo. They gave me thoughtful and detailed feedback on my researches.

Table 10: The mean of test error by 0-1 loss and standard deviation (linear, robust SVC). The noise level is 20% in SVC, while it is 10% in SVR. The numbers in bold face indicate the better method in terms of the test error.

Data	C-SVC	CCCP-θ	OP-θ	CCCP-s	OP-s
D1	.053(.013)	.052(.022)	**.048(.024)**	.052(.018)	**.048(.021)**
D2	.152(.018)	.152(.018)	.152(.018)	**.157(.021)**	.163(.013)
D3	.277(.025)	.274(.020)	**.266(.029)**	.279(.026)	**.268(.024)**
D4	.077(.015)	.051(.014)	**.044(.012)**	.054(.015)	**.043(.009)**
D5	.124(.014)	.102(.014)	**.099(.013)**	.099(.009)	**.097(.010)**
D6	.067(.003)	.064(.003)	**.063(.004)**	.064(.002)	**.063(.004)**
D7	.218(.012)	.218(.012)	.218(.012)	**.216(.010)**	.217(.010)
D8	.021(.003)	.021(.003)	.021(.003)	.022(.003)	.022(.002)
D9	.177(.014)	.178(.013)	.178(.014)	**.163(.004)**	.165(.005)
D10	.187(.018)	.188(.018)	.188(.019)	.163(.004)	**.159(.004)**

The mean of test error by 0-1 loss and standard deviation (RBF, robust SVC).

D1	.046(.009)	**.044(.012)**	.048(.024)	**.044(.013)**	.050(.031)
D2	.157(.023)	.161(.023)	.161(.023)	**.161(.028)**	.166(.026)
D3	.279(.027)	.274(.029)	**.271(.023)**	.283(.022)	**.280(.021)**
D4	.058(.011)	.050(.006)	**.046(.012)**	.049(.012)	**.043(.008)**
D5	.129(.016)	.124(.015)	.124(.015)	.126(.016)	**.105(.009)**
D6	.048(.004)	.049(.004)	.049(.004)	.049(.005)	**.048(.006)**
D7	.045(.003)	.046(.003)	.046(.003)	.045(.003)	.045(.003)
D8	.024(.003)	.023(.003)	.023(.003)	.024(.003)	.024(.003)
D9	.169(.004)	.169(.004)	**.168(.005)**	.164(.005)	**.163(.003)**
D10	.163(.004)	.163(.004)	.163(.003)	.160(.004)	**.159(.004)**

The mean of L_1 test error and standard deviation (linear, robust SVR).

D1	.551(.233)	.246(.293)	**.225(.271)**	**.357(.264)**	.421(.273)
D2	.547(.183)	**.496(.074)**	.511(.108)	.542(.186)	**.486(.041)**
D3	.493(.105)	**.440(.125)**	.447(.122)	.586(.319)	**.568(.304)**
D4	.546(.079)	.473(.088)	**.469(.086)**	.476(.067)	**.472(.072)**
D5	.561(.034)	.549(.029)	**.543(.029)**	**.547(.028)**	.548(.027)
D6	.759(.042)	.706(.053)	**.696(.059)**	.712(.047)	**.703(.043)**
D7	.710(.008)	.704(.010)	**.693(.016)**	.719(.025)	**.717(.024)**
D8	.595(.030)	.571(.014)	**.568(.012)**	.583(.021)	**.576(.017)**
D9	.548(.035)	.530(.026)	.530(.026)	.539(.034)	**.535(.037)**
D10	.301(.018)	**.290(.016)**	.292(.016)	**.288(.013)**	.290(.021)
D11	.505(.020)	.501(.008)	.501(.007)	.502(.016)	**.501(.016)**

The mean of L_1 test error and standard deviation (RBF, robust SVR).

D1	.087(.040)	.064(.051)	**.059(.048)**	.074(.047)	**.067(.044)**
D2	.394(.049)	.369(.049)	**.361(.051)**	.377(.055)	**.332(.062)**
D3	.374(.039)	**.345(.044)**	.346(.043)	.342(.043)	**.328(.029)**
D4	.394(.059)	.399(.050)	**.393(.048)**	.385(.049)	**.381(.046)**
D5	.524(.015)	.520(.015)	.520(.013)	**.516(.022)**	.518(.026)
D6	.667(.027)	**.655(.025)**	.656(.023)	.668(.028)	**.661(.024)**
D7	.676(.011)	.674(.008)	**.673(.008)**	.674(.008)	**.672(.007)**
D8	.560(.018)	.541(.022)	**.535(.026)**	.542(.015)	**.539(.027)**
D9	.499(.016)	.493(.014)	**.491(.012)**	.495(.017)	**.492(.015)**
D10	.228(.030)	.220(.017)	**.219(.017)**	**.207(.010)**	.208(.010)
D11	.466(.018)	**.453(.020)**	.454(.020)	.466(.018)	**.464(.017)**

Table 11: The mean of test error by 0-1 loss and standard deviation (linear, robust SVC). The noise level is 25% in SVC, while it is 15% in SVR. The numbers in bold face indicate the better method in terms of the test error.

Data	C-SVC	CCCP-θ	OP-θ	CCCP-s	OP-s
D1	.072(.028)	.065(.028)	**.064(.031)**	**.066(.024)**	.068(.029)
D2	.151(.018)	.151(.018)	.151(.018)	**.153(.019)**	.154(.019)
D3	.275(.020)	**.272(.018)**	.274(.015)	**.271(.017)**	.275(.013)
D4	.088(.014)	.060(.011)	**.057(.010)**	.062(.011)	**.046(.005)**
D5	.128(.013)	.105(.013)	**.103(.011)**	.103(.014)	**.096(.014)**
D6	.071(.003)	.067(.005)	**.066(.006)**	.067(.004)	**.066(.005)**
D7	.241(.009)	**.239(.009)**	.240(.009)	**.239(.009)**	.241(.010)
D8	.022(.003)	.022(.003)	.022(.003)	.022(.003)	.022(.003)
D9	.201(.019)	.201(.019)	.201(.018)	.169(.007)	**.168(.007)**
D10	.198(.015)	**.198(.014)**	.199(.015)	.169(.004)	**.165(.004)**

The mean of test error by 0-1 loss and standard deviation (RBF, robust SVC).

D1	.061(.030)	**.057(.019)**	.062(.021)	.063(.023)	**.061(.025)**
D2	.151(.018)	.152(.018)	**.151(.018)**	.152(.023)	**.149(.023)**
D3	.277(.020)	.275(.018)	.275(.017)	.270(.022)	**.263(.017)**
D4	.065(.010)	.052(.009)	.052(.007)	.055(.008)	**.044(.007)**
D5	.131(.012)	.125(.014)	**.123(.011)**	.125(.010)	**.106(.007)**
D6	.059(.007)	.058(.007)	**.057(.006)**	.061(.007)	.061(.007)
D7	.050(.007)	.049(.006)	.049(.006)	.049(.005)	.049(.006)
D8	.025(.003)	.024(.003)	.024(.003)	.025(.003)	**.024(.003)**
D9	.172(.007)	.173(.007)	**.172(.006)**	.168(.004)	**.166(.006)**
D10	.167(.005)	.168(.005)	.168(.005)	**.164(.004)**	.166(.007)

The mean of L_1 test error and standard deviation (linear, robust SVR).

D1	.581(.275)	**.421(.326)**	.499(.372)	.524(.383)	**.462(.289)**
D2	.607(.405)	.584(.411)	**.570(.415)**	**.745(.572)**	.762(.548)
D3	.547(.190)	.531(.210)	**.492(.186)**	.716(.360)	**.640(.242)**
D4	.637(.206)	**.531(.212)**	.631(.386)	**.752(.501)**	.767(.550)
D5	.557(.025)	.547(.016)	**.542(.023)**	.568(.025)	**.551(.019)**
D6	.757(.038)	.672(.054)	**.671(.064)**	.722(.064)	**.711(.047)**
D7	.720(.016)	.710(.016)	**.703(.024)**	.713(.017)	.713(.017)
D8	.618(.051)	.585(.025)	**.580(.022)**	.599(.018)	**.596(.023)**
D9	.561(.025)	**.515(.020)**	.519(.021)	.552(.047)	**.540(.036)**
D10	.303(.017)	.285(.014)	**.283(.012)**	**.282(.018)**	.283(.017)
D11	.511(.009)	.503(.012)	.503(.010)	.512(.023)	**.508(.023)**

The mean of L_1 test error and standard deviation (RBF, robust SVR).

D1	.237(.151)	.164(.140)	**.153(.138)**	.202(.163)	**.190(.155)**
D2	.392(.067)	.355(.059)	**.348(.072)**	**.353(.059)**	.358(.056)
D3	.387(.036)	**.333(.037)**	.339(.042)	.340(.037)	**.325(.039)**
D4	.435(.069)	.428(.077)	**.420(.075)**	**.415(.072)**	.416(.058)
D5	.532(.016)	.532(.017)	**.529(.020)**	.525(.018)	**.523(.020)**
D6	.665(.034)	.648(.036)	**.644(.033)**	.649(.036)	**.643(.034)**
D7	.700(.027)	.687(.024)	**.683(.025)**	.682(.013)	.682(.016)
D8	.553(.025)	.547(.020)	**.544(.019)**	.545(.026)	**.537(.025)**
D9	.499(.008)	**.490(.011)**	.492(.012)	.498(.015)	**.494(.013)**
D10	.243(.021)	.241(.016)	**.238(.015)**	.228(.023)	.228(.022)
D11	.482(.024)	**.463(.020)**	.464(.020)	.470(.011)	.470(.011)

References

[1] T. A. Manolio, F. S. Collins, N. J. Cox, D. B. Goldstein, L. A. Hindorff, D. J. Hunter, M. I. McCarthy, E. M. Ramos, L. R. Cardon, A. Chakravarti *et al.*, "Finding the missing heritability of complex diseases," *Nature*, vol. 461, no. 7265, pp. 747–753, 2009.

[2] T. A. Manolio and F. S. Collins, "Genes, environment, health, and disease: facing up to complexity." *Human heredity*, vol. 63, no. 2, pp. 63–66, 2006.

[3] H. J. Cordell, "Detecting gene–gene interactions that underlie human diseases," *Nature Reviews Genetics*, vol. 10, no. 6, pp. 392–404, 2009.

[4] J. Fan and J. Lv, "Sure independence screening for ultrahigh dimensional feature space," *Journal of The Royal Statistical Society B*, vol. 70, pp. 849–911, 2008.

[5] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on.* IEEE, 1993, pp. 40–44.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1996.

[7] J. D. Lee, D. L. Sun, Y. Sun, J. E. Taylor *et al.*, "Exact post-selection inference, with application to the lasso," *The Annals of Statistics*, vol. 44, no. 3, pp. 907–927, 2016.

[8] W. Fithian, D. Sun, and J. Taylor, "Optimal inference after model selection," *arXiv preprint arXiv:1410.2597*, 2014.

[9] W. Fithian, J. Taylor, R. Tibshirani, and R. Tibshirani, "Selective sequential model selection," *arXiv preprint arXiv:1512.02565*, 2015.

[10] X. Tian and J. Taylor, "Asymptotics of selective inference," *Scandinavian Journal of Statistics*, vol. 44, no. 2, pp. 480–499, 2017.

[11] X. Tian, J. Taylor *et al.*, "Selective inference with a randomized response," *The Annals of Statistics*, vol. 46, no. 2, pp. 679–710, 2018.

[12] R. J. Tibshirani, A. Rinaldo, R. Tibshirani, L. Wasserman *et al.*, "Uniform asymptotic inference and the bootstrap after model selection," *The Annals of Statistics*, vol. 46, no. 3, pp. 1255–1287, 2018.

[13] J. Taylor and R. Tibshirani, "Post-selection inference for-penalized likelihood models," *Canadian Journal of Statistics*, vol. 46, no. 1, pp. 41–61, 2018.

[14] F. Yang, R. F. Barber, P. Jain, and J. Lafferty, "Selective inference for group-sparse linear models," in *Advances in Neural Information Processing Systems*, 2016, pp. 2469–2477.

[15] R. F. Barber, E. J. Candès *et al.*, "A knockoff filter for high-dimensional selective inference," *The Annals of Statistics*, vol. 47, no. 5, pp. 2504–2537, 2019.

[16] J. J. Heckman, "Sample selection bias as a specification error," *Econometrica: Journal of the econometric society*, pp. 153–161, 1979.

[17] Y. Benjamini and D. Yekutieli, "False discovery rate–adjusted multiple confidence intervals for selected parameters," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 71–81, 2005.

[18] J. D. Storey *et al.*, "The positive false discovery rate: a bayesian interpretation and the q-value," *The Annals of Statistics*, vol. 31, no. 6, pp. 2013–2035, 2003.

[19] J. D. Lee and J. E. Taylor, "Exact post model selection inference for marginal screening," in *Advances in Neural Information Processing Systems*, 2014, pp. 136–144.

[20] R. J. Tibshirani, J. Taylor, R. Lockhart, and R. Tibshirani, "Exact post-selection inference for sequential regression procedures," *Journal of the American Statistical Association*, vol. 111, no. 514, pp. 600–620, 2016.

[21] K. Nakagawa, S. Suzumura, M. Karasuyama, K. Tsuda, and I. Takeuchi, "Safe pattern pruning: An efficient approach for predictive pattern mining," in *Proceedings of the 22nd*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016, pp. 1785–1794.

[22] H. Saigo, T. Uno, and K. Tsuda, "Mining complex genotypic features for predicting hiv-1 drug resistance," *Bioinformatics*, vol. 23, no. 18, pp. 2455–2462, 2007.

[23] K. Tsuda, "Entire regularization paths for graph data," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 919–926.

[24] S. Morishita, "Computing optimal hypotheses efficiently for boosting," in *Progress in Discovery Science*. Springer, 2002, pp. 471–481.

[25] K. Liu, J. Markovic, and R. Tibshirani, "More powerful post-selection inference, with application to the lasso," *arXiv preprint arXiv:1801.09037*, 2018.

[26] M. G. G'Sell, S. Wager, A. Chouldechova, and R. Tibshirani, "Sequential selection procedures and false discovery rate control," *Journal of the royal statistical society: series B (statistical methodology)*, vol. 78, no. 2, pp. 423–444, 2016.

[27] S.-Y. Rhee, M. J. Gonzales, R. Kantor, B. J. Betts, J. Ravela, and R. W. Shafer, "Human immunodeficiency virus reverse transcriptase and protease sequence database," *Nucleic acids research*, vol. 31, no. 1, pp. 298–303, 2003.

[28] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *The Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2003.

[29] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pp. 144–152, 1992.

[30] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[31] V. N. Vapnik, *Statistical Learning Theory*. Wiley Inter-Science, 1998.

[32] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, vol. 3, 2003, pp. 912–919.

[33] X. Zhu, "Semi-supervised learning literature survey," 2005.

[34] M. C. Yuen, I. King, and K. S. Leung, "A survey of crowdsourcing systems," in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on.* IEEE, 2011, pp. 766–773.

[35] K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," *RN*, vol. 15, p. 01, 2015.

[36] H. Masnadi-Shiraze and N. Vasconcelos, "Functional gradient techniques for combining hypotheses," in *Advances in Large Margin Classifiers.* MIT Press, 2000, pp. 221–246.

[37] X. Shen, G. Tseng, X. Zhang, and W. H. Wong, "On ψ-learning," *Journal of the American Statistical Association*, vol. 98, no. 463, pp. 724–734, 2003.

[38] N. Krause and Y. Singer, "Leveraging the margin more carefully," in *Proceedings of the 21st International Conference on Machhine Learning*, 2004, pp. 63–70.

[39] Y. Liu, X. Shen, and H. Doss, "Multicategory ψ-learning and support vector machine: Computational tools," *Journal of Computational and Graphical Statistics*, vol. 14, pp. 219–236, 2005.

[40] Y. Liu and X. Shen, "Multicategory ψ-learning," *Journal of the American Statistical Association*, vol. 101, p. 98, 2006.

[41] L. Xu, K. Crammer, and D. Schuurmans, "Robust support vector machine traiing via convex outlier ablation," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2006.

[42] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Trading convexity for scalability," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 201–208.

[43] Y. Wu and Y. Liu, "Robust truncated hinge loss support vector machines," *Journal of the American Statistical Association*, vol. 102, pp. 974–983, 2007.

[44] H. Masnadi-Shirazi and N. Vasconcelos, "On the design of loss functions for classification: theory, robustness to outliers, and savageboost," in *Advances in Neural Information Processing Systems*, vol. 22, 2009, pp. 1049–1056.

[45] Y. Freund, "A more robust boosting algorithm," *arXiv:0905.2138*, 2009.

[46] Y. Yu, M. Yang, L. Xu, M. White, and D. Schuurmans, "Relaxed clipping: a global training method for robust regression and classification," in *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[47] J. Hromkovic, *Algorithmics for Hard Problems*. Springer, 2001.

[48] E. L. Allgower and K. George, "Continuation and path following," *Acta Numerica*, vol. 2, pp. 1–63, 1993.

[49] T. Gal, *Postoptimal Analysis, Parametric Programming, and Related Topics*. Walter de Gruyter, 1995.

[50] K. Ritter, "On parametric linear and quadratic programming problems," *mathematical Programming: Proceedings of the International Congress on Mathematical Programming*, pp. 307–335, 1984.

[51] M. J. Best, "An algorithm for the solution of the parametric quadratic programming problem," *Applied Mathemetics and Parallel Computing*, pp. 57–76, 1996.

[52] A. L. Yuille and A. Rangarajan, "The concave-convex procedure (cccp)," in *Advances in Neural Information Processing Systems*, vol. 14, 2002.

[53] C. H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *Annals of Statistics*, vol. 38, pp. 894–942, 2010.

[54] R. Mazumder, J. H. Friedman, and T. Hastie, "Sparsenet: coordinate descent with non-convex penalties," *Journal of the American Statistical Association*, vol. 106, pp. 1125–1138, 2011.

[55] H. Zhou, A. Armagan, and D. B. Dunson, "Path following and empirical Bayes model selection for sparse regression," *arXiv:1201.3528*, 2012.

[56] K. Ogawa, M. Imamura, I. Takeuchi, and M. Sugiyama, "Infinitesimal annealing for training semi-supervised support vector machines," in *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[57] D. DeCoste and K. Wagstaff, "Alpha seeding for support vector machines," in *Proceeding of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

[58] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, "The entire regularization path for the support vector machine," *Journal of Machine Learning Research*, vol. 5, pp. 1391–415, 2004.

[59] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in Neural Information Processing Systems*, 2001, vol. 13, pp. 409–415.

[60] J. Giesen, M. Jaggi, and S. Laue, "Approximating parameterized convex optimization problems," *ACM Transactions on Algorithms*, vol. 9, 2012.

[61] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.