Parametric Programming Approach for More Powerful and General Lasso Selective Inference

Vo Nguyen Le Duy
Nagoya Institute of Technology and RIKEN
duy.mllab.nit@gmail.com

Ichiro Takeuchi
Nagoya Institute of Technology and RIKEN
takeuchi.ichiro@nitech.ac.jp

Abstract

Selective Inference (SI) has been actively studied in the past few years for conducting inference on the features of linear models that are adaptively selected by feature selection methods such as Lasso. The basic idea of SI is to make inference conditional on the selection event. Unfortunately, the main limitation of the original SI approach for Lasso is that the inference is conducted not only conditional on the selected features but also on their signs — this leads to loss of power because of over-conditioning. Although this limitation can be circumvented by considering the union of such selection events for all possible combinations of signs, this is only feasible when the number of selected features is sufficiently small. To address this computational bottleneck, we propose a parametric programming-based method that can conduct SI without conditioning on signs even when we have thousands of active features. The main idea is to compute the continuum path of Lasso solutions in the direction of the selected test statistic, and identify the subset of the data space corresponding to the feature selection event by following the solution path. The proposed parametric programming-based method not only avoids the aforementioned computational bottleneck but also improves the performance and practicality of SI for Lasso in various respects. We conduct several experiments to demonstrate the effectiveness and efficiency of our proposed method.

1 Introduction

Reliable machine learning (ML), which is the problem of assessing the reliability of data-driven knowledge obtained by ML algorithms, is one of the most important issues in the ML community. Among various approaches for reliable ML, selective inference (SI) has been recognized as a new promising approach for assessing the statistical reliability of data-driven hypotheses selected by complex data analysis algorithms.

SI was first introduced as a statistical inference tool for the features selected by Lasso (Tibshirani, 1996). Although various properties of Lasso have been extensively studied in the past decades (see, e.g., Hastie et al. (2015)), exact statistical inference such as computing $p$-values or confidence intervals for adaptively selected features by Lasso has only recently begun to be actively studied in the context of SI (Lee et al., 2016, Fithian et al., 2014, Liu et al., 2018).

The main idea of SI is to make inference for the selected features conditional on the selection event, leading to exact valid inference on adaptively selected features by Lasso is possible in the sense that $p$-values for proper false positive rate control or confidence intervals with proper coverage guarantees can be obtained. After the seminal work (Lee et al., 2016), conditional inference-based SI has been actively studied and applied to various problems (Bachoc et al., 2014, Fithian et al., 2014, 2015, 2017, Tian and Taylor, 2018, Chen and Bien, 2019, Hyun et al., 2018, Bachoc et al., 2018, Charkhi and Claeskens, 2018, Loftus and Taylor, 2014, 2015, Panigrahi et al., 2016, Tibshirani et al., 2016, Yang et al., 2016, Suzumura et al., 2017, Tanizaki et al., 2020, Duy et al., 2020a, Sugiyama et al., 2020).

Existing works and their drawbacks. Let $A$ be a random variable indicating the set of the selected features by applying Lasso on any random data sample and $s$ be their signs. Then in the seminal work (Lee et al., 2016), the authors showed that the selection event $\{A = A_{\text{obs}}, s = s_{\text{obs}}\}$ is characterized as
a polytope in the data space, where \( A_{\text{obs}} \) and \( s_{\text{obs}} \) are the corresponding observations (see §2 for detailed setup), leading to the sampling distribution of the test-statistic in the form of a truncated Normal distribution. However, it is well-known that conditioning on the signs leads to low statistical power because of over-conditioning, which is widely recognized as a major drawback of the current Lasso SI approach.

Lee et al. (2016) also discussed the solution to overcome the drawback by conducting conditional inferences without sign event \( \{ A = A_{\text{obs}} \} \), which can be characterized by \( 2^{\lvert A_{\text{obs}} \rvert} \) polytopes. If the number of selected features \( \lvert A_{\text{obs}} \rvert \) is moderate (e.g., up to 15), it is feasible to consider all affine constraints of all these \( 2^{\lvert A_{\text{obs}} \rvert} \) polytopes. However, if \( \lvert A_{\text{obs}} \rvert \) is large, it is infeasible to enumerate the whole affine constraints for exponentially increasing number of polytopes.

Recently, Liu et al. (2018) have proposed two approaches, in which the problem settings are different from Lee et al. (2016), to improve the power. However, in their first approach, it is only applicable when the number of features \( p \) is smaller than the number of instances \( n \). In the second approach, they still consider an exponentially large number of all possible sign vectors. This paper is motivated by Section 6 of Liu et al. (2018), in which they provide a recipe for constructing more powerful conditional SI methods. In the other direction, Tian and Taylor (2018) and Ieradu and Shimodaira (2019) proposed methods using randomization. A drawback of these randomization-based approaches including simple data-splitting approach is that further randomness is added in both feature selection and inference stages.

Many machine learning tasks involve careful tuning of a regularization parameter \( \lambda \) that controls the balance between an empirical loss term and a regularization term, e.g., commonly by cross-validation (CV). However, most of the current Lasso SI methods assume a pre-specified \( \lambda \) and ignore the fact that \( \lambda \) is selected based on the data because the selection event of cross-validation is difficult to characterize. Loftus (2015) and Markovic et al. (2017) proposed solutions to incorporate CV event. However, the former requires additional conditioning on all intermediate models which leads to loss of power and the latter considers a randomization version of CV instead of the vanilla CV.

**Contribution.** Our contributions are as follows:

- In this paper, we propose a new SI approach based on parametric-programming (PP) (Ritter, 1984; Allgorver and George, 1993; Gal, 1995; Best, 1996), which we call PP-based SI, for resolving several major limitations of the seminal polytope-based SI proposed by Lee et al. (2016). The polytope-based SI is applicable when the selection event can be characterized as a polytope in the data space. Otherwise, the only way is to consider extra conditions, e.g., sign conditioning, so that the over-conditioned event is characterized as a polytope, which leads to loss of statistical power. In contrast, with the proposed PP-based SI, it is possible to characterize the selection event even if they cannot be described by a polytope.

- We introduce a method to compute the continuum path of Lasso solutions in the direction of interest, which is sub-sequently used to identify the exact sampling distribution of the test statistic with the minimum amount of conditioning. Therefore, the PP-based SI can fundamentally resolve the over-conditioning problem, which was a major concern in polytope-based SI, to achieve the high statistical power. Although the concept of PP has been used in various problems (Osborne et al., 2000; Efron and Tibshirani, 2004; Hastie et al., 2004; Rosset et al., 2006; Bach et al., 2009; Rosset and Zhu, 2007; Tsuda, 2007; Lee and Scott, 2007; Garrigues and Ghaoui, 2008; Takeuchi et al., 2009; Karasuyama and Takeuchi, 2010; Hocking et al., 2011; Karasuyama et al., 2012; Lee, 2019), this is the first study that introduces a piecewise-linear PP approach for characterizing the selection events in SI.

- Furthermore, by using PP-based SI, we can perform SI with minimal conditioning for regularizing parameter selection by cross-validation, which is complicated and was not possible with polytope-based SI. Besides, we show that our proposed method is general and can be applied in several settings as well as other selection models such as elastic net and interaction model.

Figure 1 shows the schematic illustration of the proposed method. For reproducibility, our implementation is available at https://github.com/vonguyenleduy/parametric_lasso_selective_inference.

**2 Problem Statement**

To formulate the problem, we consider a random response vector

\[ Y = (Y_1, ..., Y_n)^T \sim \mathcal{N}(\mu, \Sigma), \]  

where \( n \) is the number of instances, \( \mu \) is modeled as a linear function of \( p \) features \( x_1, ..., x_p \in \mathbb{R}^n \), and \( \Sigma \in \mathbb{R}^{n \times n} \) is a covariance matrix which is known or estimable from independent data. The goal is to statistically quantify the significance of the relation between the features and response while properly controlling the false positive rate. To achieve the goal,
Figure 1: Schematic illustration of the proposed method. By applying Lasso on the observed data $y^{\text{obs}}$, we obtain the observed active set $A_{\text{obs}}$. The statistical inference for each selected feature is conducted conditional on the subspace $Y$ whose data has the same active set as $y^{\text{obs}}$. We introduce a parametric programming method for characterizing the conditional data space $Y$ by searching on the parametrized line.

Since the hypothesis is generated from the data, selection bias exists. In order to correct the selection bias, we have to remove the information that has been used for initial hypothesis generating process. This is achieved by considering the sampling distribution of the test statistic $\eta_j^TY$ conditional on the selection event, i.e.,

$$\eta_j^TY \mid \{A(Y) = A(y^{\text{obs}}), q(Y) = q(y^{\text{obs}})\}, \quad (7)$$

where $q(Y) = (I_n - c\eta_j^TY)Y$ with $c = \Sigma\eta_j(\eta_j^T\Sigma\eta_j)^{-1}$. The second condition $q(Y) = q(y^{\text{obs}})$ indicates the component that is independent of the test statistic for a random vector $Y$ is the same as the one for $y^{\text{obs}}$. The $q(Y)$ corresponds to the component $z$ in the seminal paper (see [Lee et al. (2016)], Sec 5, Eq 5.2 and Theorem 5.2).

Once the selection event is identified, we can easily compute the pivotal quantity

$$P_{j_1}^{z} \mu, \eta_j^T \Sigma\eta_j(\eta_j^TY) \mid \{A(Y) = A(y^{\text{obs}}), q(Y) = q(y^{\text{obs}})\}, \quad (8)$$

which is the c.d.f. of the truncated Normal distribution with mean $\eta_j^T \mu$, variance $\eta_j^T \Sigma \eta_j$, and the truncation region $Z$ which is calculated based on the selection event. The pivotal quantity is crucial for calculating p-value and confidence interval. Based on the pivotal quantity, we can consider selective type I error or selective p-value ([Fithian et al. (2014)]) in the form of

$$P_{j_1}^{\text{selective}} = 2 \min\{\pi_j, 1 - \pi_j\}, \quad (9)$$

where $\pi_j = 1 - F_{0,\eta_j^T \Sigma \eta_j}(\eta_j^TY)$, which is valid in the sense that

$$\text{Pr}_{H_{0,j}}(P_{j_1}^{\text{selective}} < \alpha) = \alpha, \forall \alpha \in [0, 1].$$

Furthermore, to obtain $1-\alpha$ confidence interval for any $\alpha \in [0, 1]$, by inverting the pivotal quantity in Equation (8), we can find the smallest and largest values of
\( \eta_j^\top \mu \) such that the value of pivotal quantity remains in the interval \( \left[ \frac{s}{2}, 1 - \frac{s}{2} \right] \) \cite{Lee et al. 2016}.

However, the main challenge is that characterizing \( A(Y) = A(y_{\text{obs}}) \) in Equation (7) is intractable because we have to consider \( 2^{|A(y_{\text{obs}})|} \) possible sign vectors. To overcome this issue, \cite{Lee et al. 2016} consider inference conditional not only on the selected features but also on their signs. Unfortunately, additionally considering the signs leads to low statistical power because of over-conditioning.

In the next section, we will introduce a method for identifying the minimum amount of conditioning \( \{A(Y) = A(y_{\text{obs}}), q(Y) = q(y_{\text{obs}})\} \), which leads to high statistical power. The main idea is to compute the path of Lasso solutions in the direction of interest \( \eta_j \). By focusing on the line along \( \eta_j \), we can skip majority of the polytopes that do not affect the truncated Normal sampling distribution because they do not intersect with this line. In other words, we can skip majority of combinations of signs that never appear when applying Lasso to the data on the line.

### 3 Proposed Method

In this section, we propose a parametric programming approach for characterizing conditioning event in Equation (7). The schematic illustration is shown in Figure 4.

#### 3.1 Conditional Data Space Characterization

Let us define the set of \( y \in \mathbb{R}^n \) which satisfies the conditions in Equation (7) as

\[
\mathcal{Y} = \{ y \in \mathbb{R}^n \mid A(y) = A(y_{\text{obs}}), q(y) = q(y_{\text{obs}}) \}. \tag{10}
\]

According to the second condition, the data in \( \mathcal{Y} \) is restricted to a line (see Sec 6 in Liu et al. \cite{2018} and Pithian et al. \cite{2014}). Therefore, the set \( \mathcal{Y} \) can be re-written, using a scalar parameter \( z \in \mathbb{R} \), as

\[
\mathcal{Y} = \{ y(z) = a + bz \mid z \in \mathcal{Z} \}, \tag{11}
\]

where \( a = q(y_{\text{obs}}), b = \Sigma \eta_j (\eta_j^\top \Sigma \eta_j)^{-1} \), and

\[
\mathcal{Z} = \{ z \in \mathbb{R} \mid A(y(z)) = A(y_{\text{obs}}) \}. \tag{12}
\]

Now, let us consider a random variable \( z \in \mathbb{R} \) and its observation \( z_{\text{obs}} \in \mathbb{R} \), which satisfy \( Y = a + bZ \) and \( y_{\text{obs}} = a + bZ_{\text{obs}} \). The conditional inference in Equation (7) is re-written as the problem of characterizing the sampling distribution of

\[
Z \mid \{ Z \in \mathcal{Z} \}. \tag{13}
\]

Since \( Z \sim \mathcal{N}(0, \Sigma \eta_j) \) under the null hypothesis, the law of \( Z \mid Z \in \mathcal{Z} \) follows a truncated Normal distribution. Once the truncation region \( \mathcal{Z} \) is identified, the pivotal quantity in Equation (7) is equal to \( F_{\mathcal{Z}, \Sigma \eta_j}(Z) \), and can be easily obtained. Thus, the remaining task is to characterize \( \mathcal{Z} \).

#### Characterization of truncation region \( \mathcal{Z} \).

Let us introduce the optimization problem (2) with parametrized response vector \( y(z) \) for identifying the minimum amount of conditioning lighter, we write

\[
\hat{\beta}(z) = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \|y(z) - X\beta\|_2^2 + \lambda\|\beta\|_1. \tag{14}
\]

The subdifferential of the \( \ell_1 \)-norm at \( \hat{\beta}(z) \) is defined as follows:

\[
\partial \|\hat{\beta}(z)\|_1 = s(z): \begin{cases} s_j(z) = \text{sign}(\hat{\beta}_j(z)) & \text{if } \hat{\beta}_j(z) \neq 0 \\ s_j(z) \in [-1, 1] & \text{if } \hat{\beta}_j(z) = 0, \end{cases}
\]

where we denote \( s(z) = \text{sign}(\hat{\beta}(z)) \). Then, for any \( z \) in \( \mathbb{R} \), the optimality condition is given by

\[
X^\top \left( X\hat{\beta}(z) - y(z) \right) + \lambda s(z) = 0, \tag{15}
\]

\( s(z) \in \partial \|\hat{\beta}(z)\|_1 \). To construct the truncation region \( \mathcal{Z} \) in Equation (12), we have to 1) compute the entire path of \( \hat{\beta}(z) \), and 2) identify the set of intervals of \( z \) on which \( A(y(z)) = A(y_{\text{obs}}) \). However, it seems intractable to compute \( \hat{\beta}(z) \) for infinitely many values of \( z \in \mathbb{R} \). Our main idea to overcome this difficulty is to propose a parametric programming method for efficiently computing a finite number of “transition points” at which the active set changes.

#### 3.2 A Piecewise Linear Homotopy

We now derive the main technique. We show that \( \hat{\beta}(z) \) is a piecewise linear function of \( z \). To make the notation lighter, we write \( A_z = A(y(z)) \), and we denote the set of inactive features as \( A_z^c \).

**Lemma 1.** Consider two real values \( z' \) and \( z \) (\( z' > z \)). Suppose \( |s_j(z)| < 1 \) for all \( j \in A_z^c \), \( |s_j(z')| < 1 \) for all \( j \in A_z^c \), and \( X_{A_z}^\top X_{A_z} \) is invertible. If \( \hat{\beta}_{A_z}(z) \) and \( \hat{\beta}_{A_z}(z') \) have the same active set and the same signs, then we have

\[
\hat{\beta}_{A_z}(z') - \hat{\beta}_{A_z}(z) = \psi_{A_z}(z) \times (z' - z), \tag{16}
\]

\[
\lambda s_{A_z^c}(z') - \lambda s_{A_z^c}(z) = \gamma_{A_z^c}(z') \times (z' - z), \tag{17}
\]

where \( \psi_{A_z}(z) = (X_{A_z}^\top X_{A_z})^{-1}X_{A_z}^\top b \), and \( \gamma_{A_z^c}(z) = X_{A_z^c}^\top b - X_{A_z^c}^\top X_{A_z} \psi_{A_z}(z) \).

**Proof.** From the optimality conditions of the Lasso, we have

\[
X_{A_z}^\top X_{A_z} \hat{\beta}_{A_z}(z) - X_{A_z}^\top y(z) + \lambda s_{A_z}(z) = 0, \tag{18}
\]

\[
X_{A_z^c}^\top X_{A_z^c} \hat{\beta}_{A_z^c}(z') - X_{A_z^c}^\top y(z') + \lambda s_{A_z^c}(z') = 0. \tag{19}
\]
Then, by subtracting (18) from (19) and \( A_z = A_{z'} \), we have
\[
\begin{align*}
\hat{\beta}_{A_z}(z') - \hat{\beta}_{A_z}(z) &= (X_{A_z}^T X_{A_z})^{-1} X_{A_z}^T (y(z') - y(z)) \\
&= (X_{A_z}^T X_{A_z})^{-1} X_{A_z}^T (a + b z' - a - b z) \\
&= (X_{A_z}^T X_{A_z})^{-1} X_{A_z}^T b \times (z' - z).
\end{align*}
\]
Thus, we achieve Equation (16). Next, from the optimality conditions of the Lasso, we also have
\[
\begin{align*}
-X_{A_z}^T X_{A_z} \hat{\beta}_{A_z}(z) + X_{A_z}^T y(z) = \lambda s_{A_z}(z), \quad (20) \\
-X_{A_z}^T X_{A_z} \hat{\beta}_{A_z}(z') + X_{A_z}^T y(z') = \lambda s_{A_z}(z'). \quad (21)
\end{align*}
\]
Similarly, by subtracting (20) from (21) and \( A_z = A_{z'} \), we can easily achieve Equation (17). \( \square \)

**Remark 1.** In this paper, we assume the uniqueness of the Lasso solution \( \hat{\beta}(z) \) for all \( z \in \mathbb{R} \) as well as \( |s_j(z)| < 1 \) for all \( j \in A_z \) and the invertibility of \( X_{A_z}^T X_{A_z} \). These assumptions are justified by assuming the columns of \( X \) are in general position \cite{Tibshirani2013}. Parametric programming methods for handling the rare cases where these assumptions are not satisfied have been studied, e.g., in \cite{Best1996}, and can be applied to our problem setup. In practice, when the design matrix is not in general position, it is also common to introduce an additional ridge penalty term, resulting in the elastic net \cite{Zou2005}. Our proposed method can be extended to the elastic net case (see Appendix 2.1 for the details).

**Computation of the transition point.** From Lemma 1, the solution \( \hat{\beta}(z) \) is a linear function of \( z \) until \( z \) reaches a transition point at which either an element of \( \hat{\beta}(z) \) becomes zero or a component of \( s(z) \) becomes one in absolute value. We now introduce how the transition point is identified.

**Lemma 2.** Let \( z \) be a real value such that \( \max_{j \in A_z^c} |s_j(z)| < 1 \). Then, \( A_{z'} = A_z \), \( \max_{j \in A_z^c} |s_j(z')| < 1 \), and \( s(z) = s(z') \) for any real value \( z' \) in the interval \([z, z + t_z)\), where \( z + t_z \) is the value of transition point,
\[
\begin{align*}
t_z &= \min \{t_1, t_2\}, \quad (22) \\
t_1 &= \min_{j \in A_z} \left( -\frac{\hat{\beta}_j(z)}{\psi_j(z)} \right)_{++}, \quad (23) \\
t_2 &= \min_{j \in A_z^c} \left( \frac{\lambda \text{sign}(\gamma_j(z)) - s_j(z)}{\gamma_j(z)} \right)_{++}. \quad (24)
\end{align*}
\]
Here, we use the convention that for any \( m \in \mathbb{R} \), \( (m)_{++} = m \) if \( m > 0 \), and \( (m)_{++} = \infty \) otherwise.

**Proof.** From Equation (16), we can see that \( \hat{\beta}_{A_z}(z) \) is a function of \( z \). For a real value \( z \), there exists \( t_z \) such that for any real value \( z' \) in \([z, z + t_z)\), all elements of \( \beta_{A_z}(z') \) remain the same signs with \( \beta_{A_z}(z) \). Similarly, from Equation (17), we can see that \( s_{A_z}(z) \) is a function of \( z \). Then, for a real value \( z \), there exists \( t_z \) such that for any real value \( z' \) in \([z, z + t_z)\), all elements of \( s_{A_z}(z') \) are smaller than 1 in absolute value. Finally, by taking \( t_z = \min\{t_1, t_2\} \), we obtain the interval in which the active set and signs of Lasso solution remain the same. The remaining task is to compute \( t_1 \) and \( t_2 \). We defer the detailed derivations of \( t_1 \) and \( t_2 \) to the Appendix 2.1. \( \square \)

**3.3 Algorithm**

In this section, we show the detailed algorithm of our proposed parametric programming method. In Algorithm 1 for feature selection step, we simply apply Lasso to the data \((X, y^{\text{obs}})\), and obtain the active set \( A_{\text{obs}} \). Then, we conduct SI for each selected feature. For testing \( \beta_j, j \in A_{\text{obs}} \), we first obtain the direction of interest \( \eta_j \), which can be easily computed as in Equation (14). Second, the main task is to compute the solution path of \( \hat{\beta}(z) \) in Equation (14) for the parametrized response vector \( y(z) \), where, note that, the parametrized solution \( \hat{\beta}(z) \) are different among different \( j \in A_{\text{obs}} \) since the direction of interest \( \eta_j \) depends on \( j \). This task can be done by Algorithm 2. Finally, after having the path, we can easily obtain truncation region \( Z \) which is used to compute selective p-value or selective confidence interval.

In Algorithm 2 a sequence of transition points are computed one by one. The algorithm is initialized at \( z_k = z_{\min}, k = 0 \). At each \( z_k \), the task is to find the next transition point \( z_{k+1} \), where the active set changes. This task can be done by computing the step size in Algorithm 3. This step is repeated until
Algorithm 2 compute_solution_path
Input: X, λ, a, b, [z_min, z_max]
1: Initialization: k = 0, z_k = z_min, T = z_k
2: while z_k < z_max do
3: y(z_k) = a + b z_k
4: t_{z_k}, \hat{\beta}(z_k), A_{z_k} \leftarrow \text{compute_step_size}(X, y(z_k), \lambda)
5: z_{k+1} = z_k + t_{z_k}, T = T \cup \{z_{k+1}\} (z_{k+1} is the value of the next transition point)
6: k := k + 1
7: end while
Output: \{\hat{\beta}(z_k)\}_{z_k \in \mathcal{T}}, \{A_{z_k}\}_{z_k \in \mathcal{T}}

Algorithm 3 compute_step_size
Input: X, y(z), λ
1: Compute primal/dual Lasso solution \hat{\beta}(z), \hat{s}(z) for data (X, y(z))
2: Obtain active set \mathcal{A}_z = \{j : \hat{\beta}_j(z) \neq 0\}
3: Compute \psi_{A_z}(z), \gamma_{A_z}(z) \leftarrow \text{Lemma 1}
4: t_z^1, t_z^2 \leftarrow \text{Equations (23) and (24) in Lemma 2}
5: t_z = \min\{t_z^1, t_z^2\}
Output: t_z, \hat{\beta}(z), \mathcal{A}_z

z_k > z_max. The algorithm returns the sequences of Lasso solutions and transition points.

Choice of [z_min, z_max]. Under the normality, very positive and negative values of z do not affect the inference. Therefore, it is reasonable to consider range of values, e.g., [-20σ, 20σ] [Liu et al., 2018], where σ is the standard deviation of the sampling distribution of test statistic.

3.4 Characterization of CV-based Tuning Parameter Selection Event

In this section, we introduce a new way to characterize the minimal selection event that λ is chosen based on the data, e.g., via cross-validation, which is complicated and thus none of the currently available Lasso SI methods can handle. Given a set of regularization parameter candidates \Lambda, we denote \mathcal{V}(y^{obs}) = \lambda^{obs} \in \Lambda is the event that \lambda^{obs} is selected when performing validation on y^{obs}. The conditional inference on selected feature j when applying Lasso on \{X, y^{obs}\} is then defined as

$$\eta_j^Y | \{A(Y) = A(y^{obs}), \mathcal{V}(Y) = \mathcal{V}(y^{obs}), q(Y) = q(y^{obs})\}. \quad (25)$$

The conditional inference on selected feature j when applying Lasso on \{X, y^{obs}\} is then defined as

$$\eta_j^Y | \{A(Y) = A(y^{obs}), \mathcal{V}(Y) = \mathcal{V}(y^{obs}), q(Y) = q(y^{obs})\}. \quad (25)$$

The validation error is defined as \mathcal{E}_\lambda(z) = \frac{1}{2} ||y_{val}(z) - X_{val} \hat{\beta}_\lambda(z)||_2^2. Then, we can re-defined \mathcal{Z}_2 = \{z \in \mathbb{R} | \mathcal{E}_{\lambda^{obs}}(z) \leq \mathcal{E}_\lambda(z) for any \mathcal{\lambda} \in \Lambda\}. Since \hat{\beta}_\lambda(z) is a piecewise-linear function of z and y_{val}(z) is a linear function of z, the validation error \mathcal{E}_\lambda(z) is a piecewise-quadratic function of z. Now, for each \mathcal{\lambda} \in \Lambda, we have a corresponding piecewise-quadratic function of z. Finally, we can identify \mathcal{Z}_2 by finding the intervals of z in which the validation error \mathcal{E}_{\lambda^{obs}}(z) corresponding to \lambda^{obs} is minimum among a set of piecewise-quadratic functions.

3.5 The Generality of the Proposed Method

Since we can efficiently compute the path of Lasso solutions, our proposed method is flexible and can be easily extended to various respects. In [Liu et al., 2018], the main limitations are their method can not be applied when p > n, or requires huge computation time. With our method, all these limitations are resolved. We provide detailed discussions and solutions in Appendices 6.2.2 and 6.2.3. Besides, we also apply the proposed method to other respects, which can not be solved by the methods in [Lee et al., 2016] and [Liu et al., 2018], including characterizing the minimum amount of conditioning in elastic net [Zou and Hastie, 2005] (Appendix 6.2.1), marginal model (Appendix 6.2.4), and interaction model (Appendix 6.2.3).

4 Experiment

In this section, we will demonstrate the performance of the proposed method. Here, we present the main results. Several additional experiments can be found in Appendix 6.3.
4.1 Experimental Setup

We executed the code on Intel(R) Xeon(R) CPU E5-2687W v4 @ 3.00GHz.

Methods for comparison. We show the false positive rates (FPRs), true positive rates (TPRs) and confidence intervals (CIs) for the following cases of conditional inferences:

- **TN-A**: conditional inference without sign conditioning, which is mainly focused in this paper,

  \[\eta_j^Y | \{A(Y) = A_{\text{obs}}, q(Y) = q(y_{\text{obs}})\}.\]

- **TN-As**: conditional inference with additional sign conditioning, which is mainly focused in Lee et al. (2016),

  \[\eta_j^Y | \{A(Y) = A_{\text{obs}}, s = s_{\text{obs}}, q(Y) = q(y_{\text{obs}})\},\]

where \(s\) is the sign vector of Lasso solutions on \(Y\), and \(s_{\text{obs}}\) is the sign vector of the Lasso solutions on \(y_{\text{obs}}\).

We also show the FPRs, TPRs and CIs of data splitting (DS) method (Cox, 1975), which is the commonly used procedure for the purpose of selection bias correction. In this approach, the data is randomly divided in two halves — one half is used for model selection and the other half is used for inference.

**Synthetic data generation.** We generated \(n\) outcomes as \(y_i = x_i^T \beta + \varepsilon_i, \ i = 1, ..., n\), where \(x_i \sim N(0, I_p)\) in which \(p = 5\), and \(\varepsilon_i \sim N(0, 1)\). Here, we assume that the variance of the noise is known. In practice, the variance can be estimated from independent data. We set the regularization parameter \(\lambda = 1\) and significance level \(\alpha = 0.05\). We used Bonferroni correction to account for the multiplicity in all the experiments. If we test \(m\) selected features (hypotheses) at the same time, then the Bonferroni correction would test each individual hypothesis at \(\alpha^* = \alpha/m\). For the FPR experiments, all elements of \(\beta\) were set to 0 and we set \(n \in \{100, 200, 300, 400, 500\}\). For the TPR experiments, the first two elements of \(\beta\) were set to 0.25. We ran 100 trials for each \(n \in \{50, 100, 150, 200\}\), and we repeated this experiments 10 times. For the experiments of CIs, we set \(n = 100, p = 10\), and the first 5 elements of \(\beta\) were set to 0.25.

**Definition of TPR.** In SI, we only conduct statistical testing when there is at least one hypothesis discovered by the algorithm. Therefore, the definition of TPR, which can be also called conditional power, is as follows:

\[
\text{TPR} = \frac{\# \text{correctly detected & rejected}}{\# \text{correctly detected}},
\]

Figure 2: Results of false positive rate (FPR) control, true positive rate (TPR) and confidence interval (CI), where \# correctly detected is the number of truly positive features selected by the algorithm (e.g., Lasso) and \# rejected is the number of truly positive features whose null hypothesis is rejected by SI.

4.2 Numerical Results

**The results of FPRs, TPRs and CIs.** The results of FPR and TPR are shown in Figures 2a and 2b. In three cases, the FPRs are properly controlled under the significance level \(\alpha\). Regarding the TPR comparison, it is obvious that TN-A has the highest power. In regard to CI experiments, we note that the number of selected features between Lasso and DS can be different. Therefore, for a fair comparison, we only consider the features that are selected in both methods. In our experiments, since 9 features were selected by the Lasso in the cases of TN-A and TN-As while only 8 features were selected in the case of DS, we only show the 95% CI of the features that are selected in both cases in Figure 2. The lengths of CI obtained by TN-A are almost the shortest. We repeated this experiment 100 times and showed the boxplot of the lengths of the confidence intervals in Figure 2c. In summary, the CI results are consistent with the TPR results, i.e., TN-A has the shortest length of CI which indicates it has the highest power.

**The results when accounting CV selection event.** We also demonstrate the TPRs and the lengths of CIs between the case when \(\lambda = 2^0\) is fixed and \(\lambda\) is selected from the set \(\Lambda_1 = \{2^{-1}, 2^0, 2^1\}\) or \(\Lambda_2 = \{2^{-10}, 2^{-9}, ..., 2^9, 2^{10}\}\). We show that the TPR tends to decrease when increasing the size of \(\Lambda\) as
Parametric Programming Approach for More Powerful and General Lasso Selective Inference

Figure 3: Demonstration of TPR when accounting cross-validation selection event.

Figure 4: Demonstration of CI length when considering cross-validation selection event.

shown in Figure 3. This is due to the fact that when we increase the size of $\lambda$, we have to condition on more information which leads to shorter truncation interval and results low TPR. The TPR results are consistent with the CI results shown in Figure 4 in which the length of CI is longer when increasing the size of $\lambda$. Besides, we also conducted TPR comparison between our method and the over-conditioning version proposed in Loftus (2015). The results are shown in Figure 5. Our method has higher power since we can characterize minimum amount of conditioning.

The efficiency of the proposed method. In Lee et al. (2016), the authors mentioned the naive way to remove sign conditioning by enumerating all possible combination of signs $2^{\mid A_{obs}\mid}$ which is only feasible when $\mid A_{obs}\mid$ is small. On the left-hand side of Figure 6 we show the efficiency of our method compared to the naive way of removing sign conditioning. On the right-hand side of Figure 6 the Lasso SI without conditioning on signs can be done even when thousands of features are selected.

In Figure 8 we show the boxplot of the actual number of intervals of $z$ that we encountered on the line when constructing the truncation region $Z$. This indicates that the number of polytopes intersecting the line $z$ that we need to consider is much smaller than $2^{\mid A_{obs}\mid}$, which is considered in Lee et al. (2016)—this is the reason why the proposed approach can resolve all major limitations of the current SI method, making Lasso SI more powerful and practical.

We did not compare the computational time between the proposed method TN-A and the over-conditioning version TN-As because TN-As is obviously faster than TN-A but it has lower power than TN-A. Our main purpose is to demonstrate that the proposed method...
Figure 7: Comparison between the proposed method and methods in Liu et al. (2018), in which an exponentially increasing number of all possible sign combinations are still required.

Figure 8: Number of encountered intervals on the line.

not only has high statistical power but also has practically computational costs.

We note that, in the worst-case, the complexity of the proposed method still grows exponentially. This is a common issue in other parametric programming applications such as regularization paths. However, fortunately, it has been well-recognized that this worst case rarely happens in practice, and our experiments suggest that this also applies to PP-based SI.

The robustness of the proposed method in terms of the FPR control. We demonstrate the robustness of our method in terms of the FPR control by considering the following cases:

- Non-normal noise: we consider the noise following Laplace distribution, skew normal distribution (skewness coefficient 10), and $t_{20}$ distribution.
- Unknown $\sigma^2$: we also consider the case when the variance is estimated from the data.

We generated $n$ outcomes as $y_i = x_i^T \beta + \varepsilon_i$, $i = 1, \ldots, n$, where $p = 5$, $x_i \sim \mathcal{N}(0, I_p)$, and $\varepsilon_i$ follows Laplace distribution, skew normal distribution, or $t_{20}$ distribution with zero mean and standard deviation was set to 1. In the case of estimated $\sigma^2$, $\varepsilon_i \sim \mathcal{N}(0, 1)$. We set all elements of $\beta$ to 0, and set $\lambda = 0.5$. For each case, we ran 1,200 trials for each $n \in \{100, 200, 300, 400\}$. We confirmed that our method still maintains good performance on FPR control. The results are shown in Appendix 6.3.

4.3 Results on Real-World Datasets

We demonstrate the efficiency of the proposed method by applying it on high-dimensional real-world bioinformatics related datasets, which is available at http://www.coepra.org/CoEPrA_regr.html In datasets 1 and 3, $n$ is the number of nona-peptides. Each amino acid in a nona-peptide is described by 643 descriptors, for a total of $p = 643 \times 9 = 5787$ descriptors. In dataset 2, $n$ is the number of octa-peptides. Each amino acid in an octa-peptide is described by 643 descriptors, for a total of $p = 643 \times 8 = 5144$ descriptors. For these experiments, we used elastic net instead of Lasso to obtain large $|A_{\text{obs}}|$. The results are shown in Table 1.

Table 1: Results on high-dimensional real-world bioinformatics related datasets.

|     | n   | p    | $|A_{\text{obs}}|$ | Avg. Time (s) |
|-----|-----|------|---------------------|--------------|
| Dataset 1 | 89  | 5787 | 600                 | 0.374        |
| Dataset 2 | 76  | 5144 | 621                 | 0.344        |
| Dataset 3 | 133 | 5787 | 660                 | 0.342        |

The time shown in the table is the average time to compute $p$-value for a selected feature.

5 Conclusion

In this paper, we have proposed a general method for characterizing the selection event of Lasso SI by introducing piecewise-linear parametric programming approach. With the proposed method, we can conduct a powerful SI by conditioning only on the selected features without the need of enumerating all possible sign vectors. Besides, we also introduced a new way to characterizing the cross-validation based tuning parameter selection. The proposed method not only overcomes the drawbacks of current Lasso SI methods but also improves the performance and practicality of SI for Lasso in various respects. Our idea is general and can be applied to circumvent several drawbacks of all the methods that are based on the current SI framework. We conducted experiments on both synthetic and real-world datasets to demonstrate the effectiveness and efficiency of our proposed method.
Acknowledgements

This work was partially supported by MEXT KAKENHI (20H00601, 16H06538), JST CREST (JP-MJCR1502), RIKEN Center for Advanced Intelligence Project, and RIKEN Junior Research Associate Program.

References


