Statistica Sinica **29** (2019), 1607-1630
doi:https://doi.org/10.5705/ss.202015.0394

THE RESTRICTED CONSISTENCY PROPERTY OF LEAVE-n_v-OUT CROSS-VALIDATION FOR HIGH-DIMENSIONAL VARIABLE SELECTION

Yang Feng and Yi Yu

Columbia University and University of Bristol

Abstract: Cross-validation (CV) methods are popular for selecting the tuning parameter in high-dimensional variable selection problems. We show that a misalignment of the CV is one possible reason for its over-selection behavior. To fix this issue, we propose using a version of leave-n_v-out CV (CV(n_v)) to select the optimal model from a restricted candidate model set for high-dimensional generalized linear models. By using the same candidate model sequence and a proper order for the construction sample size n_c in each CV split, CV(n_v) avoids potential problems when developing theoretical properties. CV(n_v) is shown to exhibit the restricted model-selection consistency property under mild conditions. Extensive simulations and a real-data analysis support the theoretical results and demonstrate the performance of CV(n_v) in terms of both model selection and prediction.

Key words and phrases: Generalized linear models, leave-n_v-out cross-validation, restricted maximum likelihood estimators, restricted model-selection consistency, variable selection.

1. Introduction

Massive high-throughput data sets are becoming increasingly common as a result of technological advancements in many fields. Such data are characterized by a large number of variables p compared with the sample size n. For an overview of the many challenges associated with high-dimensional statistical modeling, refer to Fan and Lv (2010) and Bühlmann and van de Geer (2011).

A crucial goal in high-dimensional data analyses is to achieve a balance between the goodness-of-fit and the complexity of a model, because a model's predictive ability and interpretability are both important to practitioners in many scientific fields. A popular way to achieve this balance is to impose penalties on the model's complexity, which allows for simultaneous variable selection and parameter estimation in one step. This approach has been examined in numerous theoretical and numerical works. For example, Tibshirani (1996) proposed the

Lasso method, which is an ℓ_1 penalty, or equivalently, Chen and Donoho (1994) proposed the basis pursuit, Fan and Li (2001) proposed the smoothly clipped absolute deviation (SCAD) penalty, and Zhang (2010) proposed the minimax concave penalty (MCP).

An important aspect of penalization techniques is the tuning parameter, which determines the size of the penalty imposed. Over-penalization runs the risk of overlooking scientifically meaningful information; on the other hand, under-penalization may erroneously identify seemingly meaningful patterns that are actually the result of experimental noise. Therefore, it is critical to choose the tuning parameter with care.

There is an abundance of research on which information criteria to use to select the tuning parameter. These include the generalized cross-validation (CV) method (Tibshirani (1996); Wang, Li and Tsai (2007)), C_p (Efron et al. (2004)), extended Bayesian information criterion (EBIC) (Chen and Chen (2008); Luo and Chen (2014)), modified BIC (Wang, Li and Leng (2009)), and generalized information criterion (Zhang, Li and Tsai (2010); Fan and Tang (2013)). Other works propose selecting the tuning parameter by jointly estimating the regression coefficient and the standard deviation (Städler, Bühlmann and Van De Geer (2010); Sun and Zhang (2012)).

CV is a data-driven method and a popular way of selecting a tuning parameter. As such, a large amount of theoretical work has been done on using CV in fixed-dimensional linear regression models. For example, leave-one-out CV (CV(1)) has been shown to be asymptotically equivalent to the Akaike information criterion (AIC), C_p, jackknife method, and bootstrap method (Stone (1977); Efron (1983, 1986)). Shao (1993) proved the model-selection inconsistency of CV(1) for the fixed-dimensional linear regression model. In addition, for leave-n_v-out CV (CV(n_v)), the author gave the proper ratio of the size of the construction set to that of the validation set and showed that this ratio is necessary for model-selection consistency. Here, the construction and validation data sets refer to the subsets used to construct and validate the estimators in CV splits. However, K-fold CV, the most commonly used method, is well known for its conservativeness; that is, the corresponding estimator selects too many noise variables (Yu and Feng (2014b)). As mentioned in Zhang and Huang (2008), the theoretical justification for using a CV-based tuning parameter is unclear for model-selection purposes. Yu and Feng (2014b) proposed a modified CV for high-dimensional linear regression models and showed that it outperforms the regular K-fold CV in numerical experiments. Compared with Yu and Feng (2014b),

we adopt CV(n_v) for a sequence of candidate models from a complete data set. Then, we develop the restricted consistency results under the generalized linear model framework for high-dimensional variable selection.

Another related method is the relaxed Lasso (Meinshausen (2007)). This is a two-stage method, with the penalty in the second stage operating only on those variables selected in the first stage. The author conjectures that the K-fold CV of this two-step method will achieve model-selection consistency. In contrast to Meinshausen (2007), we study the theoretical behavior of CV(n_v). In particular, we focus mainly on model selection, rather than on proposing a variant of the Lasso procedure. We also provide a rigorous discussion of the asymptotic behavior of the CV.

This study offers two main contributions to the literature. First, we investigate the advantages and drawbacks of the CV methods commonly used for tuning parameter selection in penalized estimation methods. Second, we examine CV(n_v), showing that it is consistent, in a restricted sense, for a wide range of penalty functions in the high-dimensional generalized linear model framework.

We use the following notation throughout this paper. For a p-dimensional vector $\boldsymbol{\beta}$ and an $n \times p$-dimensional matrix A, suppose s is a subset of $\{1, \ldots, n\}$, and α is a subset of $\{1, \ldots, p\}$. Then $\boldsymbol{\beta}_\alpha$ represents the subvector of $\boldsymbol{\beta}$ corresponding to α, A_s represents the submatrix of A corresponding to rows with indices in s, and A^α represents the submatrix of A corresponding to columns with indices in α. Let $|s|$ represent the cardinality of set s. In addition, define the ℓ_0, ℓ_1, and ℓ_2 norms of $\boldsymbol{\beta}$ as $\|\boldsymbol{\beta}\|_0 = \sum_{j=1}^p 1\{\beta_j \neq 0\}$, $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^p |\beta_j|$, and $\|\boldsymbol{\beta}\| = [\sum_{j=1}^p \beta_j^2]^{1/2}$, respectively. Let g_1 and g_2 be two functions of n. We use $g_1(n) = \Theta(g_2(n))$ to represent that they are asymptotically of the same order; that is, there exist positive constants c_1 and c_2, such that

$$c_1 \leq \liminf_n \frac{g_1(n)}{g_2(n)} \leq \limsup_n \frac{g_1(n)}{g_2(n)} \leq c_2.$$

The rest of the paper is organized as follows. We introduce the generalized linear model setup and discuss K-fold CV in Section 2. Motivated by the problems associated with K-fold CV, in Section 3, we introduce CV(n_v) for high-dimensional variable selection, and show that it can achieve restricted model-selection consistency. Next, Section 4 discusses relevant theory, after which we present our simulation studies in Section 5 and a real-data analysis in Section 6 to compare CV(n_v) with other CV methods and popular information criteria. We conclude the paper with a short discussion in Section 7. All technical details are collected in the supplementary material.

2. Model Setup and K-fold CV

2.1. Model setup

Suppose we have n independent and identically distributed (i.i.d.) observation pairs $(\boldsymbol{x}_i, y_i)$, for $i = 1, \ldots, n$, where $\boldsymbol{x}_i$ is a p-dimensional predictor and y_i is the response. For generalized linear models, we assume the conditional distribution of y, given $\boldsymbol{x}$, belongs to an exponential family with a canonical link and the canonical parameter $\theta = \boldsymbol{x}^\top \boldsymbol{\beta}$; that is, it has the following density function:

$$f(y; \boldsymbol{x}, \boldsymbol{\beta}) = c(y, \phi) \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} \right),$$

where $\phi \in (0, \infty)$ is the dispersion parameter, and the functions $a(\cdot)$, $b(\cdot)$, and $c(\cdot, \cdot)$ are known and vary across models. Let $\boldsymbol{\beta}^o$ be the true regression parameter, with $\|\boldsymbol{\beta}^o\|_0 = d_o$. In the high-dimensional setting, p may well exceed n, but d_o is usually assumed to be strictly upper-bounded by n (i.e., $d_o < n$). Up to an affine transformation with $\theta_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}$, the log-likelihood divided by the sample size is given by

$$\ell(\boldsymbol{\beta}) = n^{-1} \sum_{i=1}^{n} \{ y_i \theta_i - b(\theta_i) \}. \tag{2.1}$$

Minimizing the penalized negative log-likelihood function leads to the following estimator:

$$\hat{\boldsymbol{\beta}}(\lambda) = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \{ -\ell(\boldsymbol{\beta}) + p_{\lambda,\gamma}(\boldsymbol{\beta}) \}, \tag{2.2}$$

where $p_{\lambda,\gamma}(\cdot)$ is the penalty function.

Given subset $s \subset \{1, \ldots, n\}$, the log-likelihood function evaluated on the subset s is

$$\ell^{(s)}(\boldsymbol{\beta}) = (|s|)^{-1} \sum_{i \in s} \{ y_i \theta_i - b(\theta_i) \}. \tag{2.3}$$

Then, the corresponding minimizer of the penalized negative log-likelihood is

$$\hat{\boldsymbol{\beta}}^{(s)}(\lambda) = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \{ -\ell^{(s)}(\boldsymbol{\beta}) + p_{\lambda,\gamma}(\boldsymbol{\beta}) \}. \tag{2.4}$$

In this study, we consider only separable sparsity-inducing penalties; that is, there exists a non-negative function $\rho(\cdot)$, such that for any vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$, the penalty function $p_{\lambda,\gamma}(\cdot)$ satisfies

$$p_{\lambda,\gamma}(\boldsymbol{\beta}) = \sum_{j=1}^{p} \rho(|\beta_j|; \lambda, \gamma), \tag{2.5}$$

where λ and γ are the parameters of the penalty function, and the minimizer

of the penalized negative log-likelihood leads to a sparse solution. Both convex and folded-concave penalties can be written in the form of (2.5). For convex penalties, such as the Lasso method (Tibshirani (1996)), $\gamma = \infty$, whereas for folded-concave penalties, $0 < \gamma < \infty$. In the penalty function given in (2.5), γ is a parameter controlling the concavity of the penalty. Here, we focus on the collection of solutions as λ changes, with γ fixed.

A popular class of algorithms used to solve (2.2) are the path algorithms. Many path algorithms have been proposed, including forward regression, stepwise regression, lars (Efron et al. (2004)), glmpath (Park and Hastie (2007)), glmnet (Friedman et al. (2010)), ncvreg (Breheny and Huang (2011)), and apple (Yu and Feng (2014a)), among others. In a path algorithm, a collection of (usually sparse) estimators $\{\hat{\boldsymbol{\beta}}_r, r = 1, \ldots, R\}$ is generated, where R represents the total number of candidate estimators. Then, the best estimate $\hat{\boldsymbol{\beta}}_{\hat{r}}$ is chosen from the R candidates according to certain criteria.

2.2. CV

There are many different versions of CV. Thus, to avoid ambiguity, we describe K-fold CV using glmnet and ncvreg in the penalized negative log-likelihood context in Algorithm 1.

Algorithm 1 K-fold CV for a typical path algorithm.

Input: The complete data set $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$, a path algorithm.

Output: The optimal location $\hat{r}$ and its corresponding solution $\hat{\boldsymbol{\beta}}_{\hat{r}}$.

1: Using the complete data set, generate a data-driven penalty parameter sequence $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_R\}$. Compute the solution path $\{\hat{\boldsymbol{\beta}}_r, r = 1, \ldots, R\}$, where $\hat{\boldsymbol{\beta}}_r = \hat{\boldsymbol{\beta}}(\lambda_r)$.

2: Randomly divide the data set into K folds, and denote the index of each fold as s_k, for $k = 1, \ldots, K$, where $s_{(-k)} = \{1, \ldots, n\} \setminus s_k$.

3: For each fold $k = 1, \ldots, K$:

a) Using the construction data in $s_{(-k)}$, generate its own penalty parameter sequence $\boldsymbol{\lambda}^{(-k)} = \{\lambda_1^{(-k)}, \ldots, \lambda_R^{(-k)}\}$.

b) Compute the corresponding solution path $\{\hat{\boldsymbol{\beta}}_r^{(-k)}, r = 1, \ldots, R\}$, where $\hat{\boldsymbol{\beta}}_r^{(-k)} = \hat{\boldsymbol{\beta}}^{(s_{(-k)})}(\lambda_r^{(-k)})$ is the penalized estimator defined in (2.4) with penalty parameter $\lambda_r^{(-k)}$.

c) Evaluate the prediction performance of $\{\hat{\boldsymbol{\beta}}_r^{(-k)}, r = 1, \ldots, R\}$ on the validation data in s_k using the negative log-likelihood function. The resulting values are denoted by $\{L_r^k, r = 1, \ldots, R\}$, where $L_r^k = -\ell^{(s_k)}(\hat{\boldsymbol{\beta}}_r^{(-k)})$, as defined in (2.3).

4: Calculate the average criterion values $\{L_r, r = 1, \ldots, R\}$, where $L_r = K^{-1} \sum_{k=1}^{K} L_r^k$. Let $\hat{r} = \arg\min_{r=1,\ldots,R} L_r$.

In Algorithm 1, to compare the performance of $\{\hat{\boldsymbol{\beta}}_r, r = 1, \ldots, R\}$, we average the prediction performance on the corresponding validation set s_k over the K folds using the estimator $\hat{\boldsymbol{\beta}}_r^{(-k)}$ from the construction set $s_{(-k)}$. However, there is no guarantee that we are averaging across the same models or the same tuning

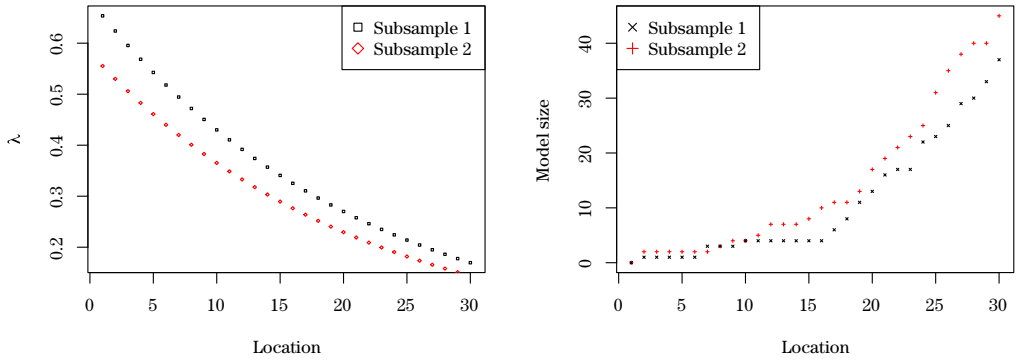

Figure 1. An example of K-fold CV.

parameters across different folds. In path algorithms, the tuning parameters are determined by the construction data set, and the estimators are determined by the tuning parameters and the construction data set.

Remark 1. Other path algorithms, including lars and glmpath, do not start with a sequence of data-driven penalty parameters. Instead, they proceed by adaptively adding/deleting one predictor at a time from the model, and then provide the corresponding $\hat{\boldsymbol{\beta}}_r$ after each operation. Note that a solution $\hat{\boldsymbol{\beta}}_r$ from such path algorithms implies a certain value of λ_r in (2.2). As a result, the preceding discussion on the averaging process applies to these algorithms as well.

Remark 2. In practice, it is common to use the tuning parameter sequence $\boldsymbol{\lambda}$ generated by the complete data set in all splits. Although this guarantees the alignment of the tuning parameters across different splits, it results in misalignment in terms of model sequences. This can cause additional problems because a desirable tuning parameter should be a function of the sample size. Furthermore, it is difficult to link a chosen tuning parameter from the splits with its performance for the complete data set.

We conduct a simple simulation for a high-dimensional linear regression example using five-fold CV. In Figure 1, we show the results for two construction data sets when performing this CV. In the left panel, we show the first 30 values of λ on each path, with the x-axis showing the location indices. The right panel presents the sequences of the model sizes against their locations on the solution paths. The CV averages the models across different splits. However, as shown in Figure 1, the corresponding λ-sequences and model-size sequences are quite different for the two splits. As a result, it is difficult to derive a theoretical justi-

fication for either the model selection or the tuning parameter selection property of the CV-tuned estimator. Further numerical results on the alignment issue of the CV are discussed in Section 5.1.

3. Leave-n_v-out CV

Now that we have a better understanding of the issues associated with CV, we propose a version of CV(n_v). We first introduce several key concepts related to model selection and CV(n_v) in Section 3.1, and then point out its major differences from CV in Section 3.2. In Section 3.3, we show that CV(n_v) is *restricted model-selection consistent* (defined formally in Section 3.1) under mild technical conditions in the generalized linear model framework for both convex and folded-concave penalties.

3.1. Key concepts

From the solution path $\{\hat{\boldsymbol{\beta}}_r, r = 1, \ldots, R\}$ in Algorithm 1, we obtain a corresponding path of models $\mathcal{A} = \{\alpha_r, r = 1, \ldots, R\}$, where $\alpha_r = \{j \in \{1, \ldots, p\} : (\hat{\beta}_r)_j \neq 0\}$ denotes the indices with nonzero coefficient estimates. Similar to Shao (1993), we divide $\mathcal{A}$ into two disjoint subsets: $\mathcal{A}_c$, and its complement $\mathcal{A} \setminus \mathcal{A}_c$, where $\mathcal{A}_c = \{\alpha \in \mathcal{A} : (X^\alpha)\boldsymbol{\beta}^o_\alpha = X\boldsymbol{\beta}^o\}$. Next, we provide three definitions, which constitute the fundamental concepts of this study.

Definition 1 (True model)**.** *The true model is defined as* $\mathcal{O} = \{j : \beta^o_j \neq 0\}$.

Here, for any estimated model $\hat{\mathcal{O}}$, we define its false negative (FN) as $|\mathcal{O} \setminus \hat{\mathcal{O}}|$ and its false positive (FP) as $|\hat{\mathcal{O}} \setminus \mathcal{O}|$. Then, for the models in $\mathcal{A}_c$, FN $= 0$, and for the models in $\mathcal{A} \setminus \mathcal{A}_c$, FN > 0.

Definition 2 (Optimal model set)**.** *Let* $d_* = \min_{\alpha \in \mathcal{A}_c} |\alpha|$. *Define the optimal model set as* $\alpha_* = \{\alpha \in \mathcal{A}_c : |\alpha| = d_*\}$.

When $|\alpha_*| = 1$, there is only one optimal model. Thus, with a slight abuse of notation, we call α_* the optimal model. The optimal models can be different from the true model, and they are the sparsest models without FNs.

Remark 3. For any model $\alpha \in \mathcal{A}$, define its fitted risk as follows:

$$R(\alpha) = \sup_{\boldsymbol{x} \in \mathbb{R}^p : \|\boldsymbol{x}\| = 1} (\boldsymbol{x}^\top_\alpha \boldsymbol{\beta}^o_\alpha - \boldsymbol{x}^\top \boldsymbol{\beta}^o)^2 = \left\|\boldsymbol{\beta}^o_{-\alpha}\right\|^2.$$

It is obvious that if $\alpha \in \mathcal{A}_c$, then $R(\alpha) = 0$; otherwise, $R(\alpha) > 0$.

We now demonstrate the differences between the true model and the optimal model (set) using a toy example. In a linear regression setting, assume that the

true regression coefficient $\boldsymbol{\beta}^o \in \mathbb{R}^{100}$, where $\beta_j^o = 1$, for $j = 1, \ldots, 5$, and $\beta_j^o = 0$, for $j = 6, \ldots, 100$. Then, the true model $\mathcal{O} = \{1, 2, 3, 4, 5\}$. The candidate models are given as follows: $\alpha_1 = \{1, 2, 3\}$, $\alpha_2 = \{1, 2, 3, 4\}$, $\alpha_3 = \{1, 2, 3, 4, 5, 6\}$, and $\alpha_4 = \{1, 2, 3, 4, 5, 6, 7\}$. Note that the true model is not among the candidate models. Both models α_1 and α_2 miss at least one important variable, with $R(\alpha_1) = 2$ and $R(\alpha_2) = 1$. The true model is a subset of both α_3 and α_4, and $R(\alpha_3) = R(\alpha_4) = 0$. In this situation, $\alpha_3, \alpha_4 \in \mathcal{A}_c$. From the definition of the optimal model (set), we know α_3 is the optimal model because it contains fewer FPs than α_4 does. As a result, it is reasonable to focus on the optimal model (set) when the true model is unavailable.

Definition 3 (Restricted model-selection consistency)**.** *We say that a method has the restricted model-selection consistency property if the selected model $\hat{\alpha}_n$ satisfies*

$$\lim_{n\to\infty} \text{pr}\{\hat{\alpha}_n \in \alpha_*\} = 1.$$

Here, we do not require a specific path algorithm, but start with a collection of candidate models. As a result, in Definition 3, *restricted model-selection consistency* means that the selected model is in the optimal model set with probability tending to one. This differs from model-selection consistency, which means $\lim_{n\to\infty} \text{pr}\{\hat{\alpha}_n = \mathcal{O}\} = 1$ in our setup. However, the two properties coincide when the true model is an available candidate (i.e. $\mathcal{O} \in \mathcal{A}$).

3.2. Methodology

The detailed CV(n_v) algorithm for the high-dimensional penalized regression is described in Algorithm 2. The main idea is to use the complete data set to derive the collection of solutions and the corresponding model sequence. The problem of selecting the optimal solution is then reduced to choosing the optimal model. In this sense, we recast the tuning parameter selection problem for high-dimensional generalized linear models to one of model selection for low-dimensional generalized linear models. Here, the different splits are the same. Therefore, the averaging has intuitive meaning.

Another key ingredient of CV(n_v) is the choice of n_c and n_v, that is, the sample sizes of the construction and validation subsets, respectively. Following Shao (1993, 1996), we choose n_c and n_v such that $n_c/n \to 0$ and $n_c \to \infty$ as $n \to \infty$. This differs from the K-fold CV methods, where a larger proportion of data is used for construction and a smaller proportion is used for validation. Next, we briefly explain the intuitive reasoning behind the specific splitting of

Algorithm 2 CV(n_v) for a typical path algorithm.

Input: The complete data set $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$, a path algorithm.
Output: The optimal location $\hat{r}$ and its corresponding solution $\tilde{\boldsymbol{\beta}}_{\hat{r}}$.

1: Compute the solution path $\{\hat{\boldsymbol{\beta}}_r, r = 1, \ldots, R\}$ using a given path algorithm with the complete data set. Obtain a sequence of models $\{\alpha_1, \ldots, \alpha_R\}$, where α_r is the support of $\hat{\boldsymbol{\beta}}_r$.
2: Independently draw validation sets $\{s_k, k = 1, \ldots, K\}$, where $s_k \subset \{1, \ldots, n\}$ and $|s_k| = n_v$. Let $s_{(-k)} = \{1, \ldots, n\} \setminus s_k$ represent the corresponding construction set $s_{(-k)}$, with $|s_{(-k)}| = n_c$.
3: For each $k = 1, \ldots, K$:

a) Using the construction data in $s_{(-k)}$, compute the collection of solutions $\{\tilde{\boldsymbol{\beta}}_r^{(-k)}, r = 1, \ldots, R\}$, where

$$\tilde{\boldsymbol{\beta}}_r^{(-k)} = \underset{\substack{\boldsymbol{\beta} \in \mathbb{R}^p, \\ \boldsymbol{\beta}_{(-\alpha_r)} = \mathbf{0}}}{\arg\min} \{-\ell^{(s_{-k})}(\boldsymbol{\beta})\}, \tag{3.1}$$

where $\ell^{(s_{(-k)})}(\cdot)$ is defined in (2.3).

b) Evaluate the prediction performance of $\{\tilde{\boldsymbol{\beta}}_r^{(-k)}, r = 1, \ldots, R\}$ on the validation set s_k using the negative log-likelihood function. The resulting values are denoted by $\{L_r^k, r = 1, \ldots, R\}$, where $L_r^k = -\ell^{(k)}(\tilde{\boldsymbol{\beta}}_r^{(-k)})$.

4: Calculate the average criterion value $\{L_r, r = 1, \ldots, R\}$, where $L_r = K^{-1} \sum_{k=1}^K L_r^k$. Set $\hat{r} = \arg\min_{r \in \{1, \ldots, R\}} L_r$, along with its corresponding solution $\tilde{\boldsymbol{\beta}}_{\hat{r}}$, as in (3.1).

the sample. Note that the purpose of CV is to select the best model from the candidates. As a result, in addition to having an accurate estimation for each model (when $n_c \to \infty$), perhaps more importantly, we need a sufficiently large ($n_c/n \to 0$) validation set in order to detect the subtle differences between the models. This is particularly challenging in the high-dimensional settings because there are many possible candidate models. The popular ten-fold CV, for example, only uses 1/10 of the data for the validation set, which has been shown to be too small for the purpose of model selection.

We now present the behavior of CV(n_v) when n_v varies using a simulation study. In Figure 2, we present the average FP and FN of CV(n_v) with a wide range of n_c in linear and logistic regression problems with $n = 500$ and $p = 1{,}000$. The remaining settings are as shown in Example 1. From Figure 2, it is clear that, in all cases, a larger order of n_c results in more FPs, but the fewer FNs. For the linear regression, $n_c = \lceil n^{1/2} \rceil$ performs best, whereas $n_c = \lceil n^{3/4} \rceil$ performs best for the logistic regression. The behaviors for the linear regression and logistic regression vary because when the covariates and the coefficients are the same, the logistic regression needs a larger sample size to fit the model well than the linear regression does. Intuitively, under the canonical link, the Fisher information for generalized linear models can be written as $1/a(\phi) X^\top W X$, where ϕ is the dispersion parameter. For the logistic regression, $W = \mathrm{diag}\{\pi_1(1-\pi_1), \ldots, \pi_n(1-\pi_n)\}$, where $\pi_i = \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})/(1 + \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})) < 1$ in non-degenerate

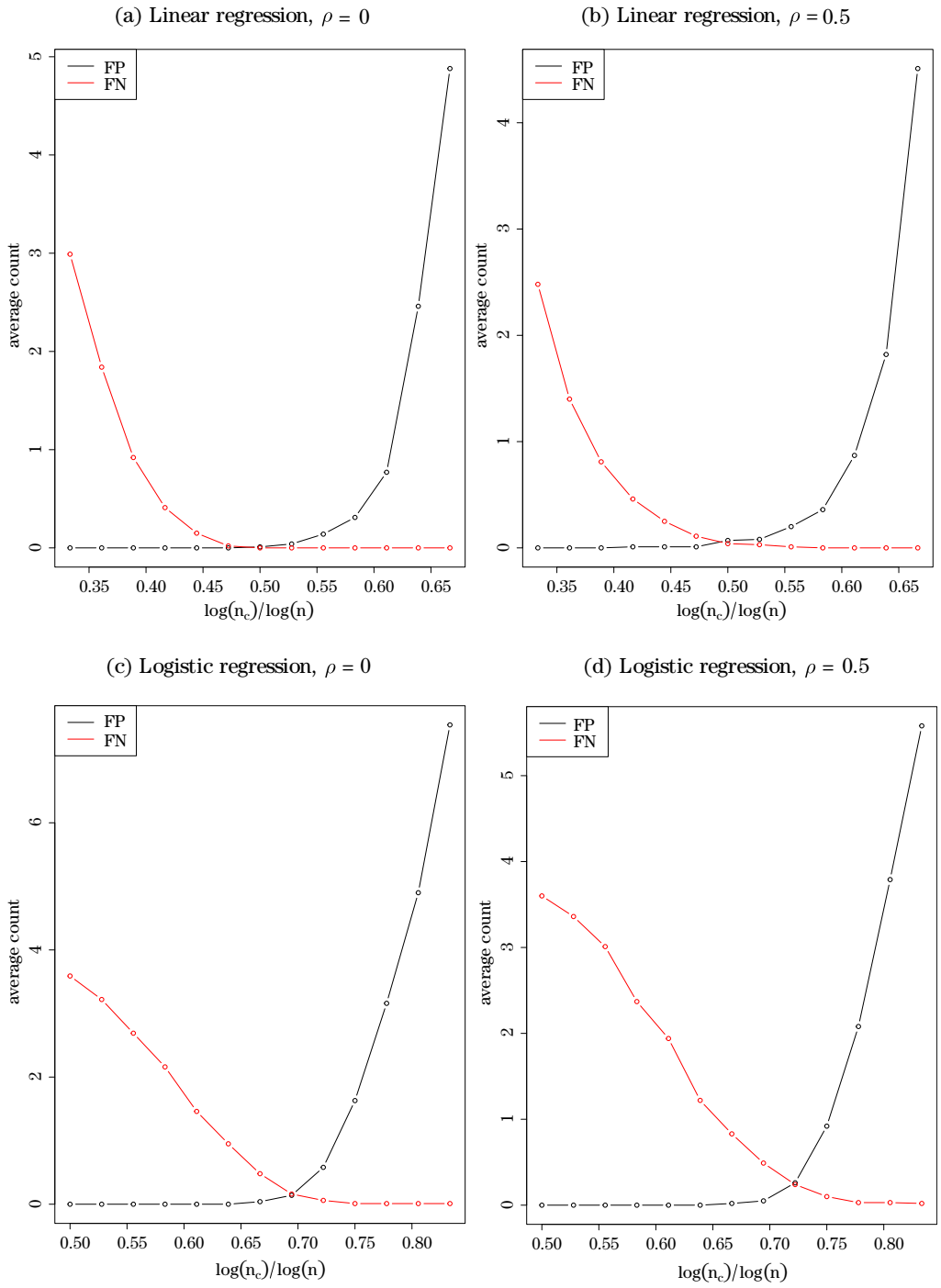

Figure 2. The FP and FN of the CV(n_v) for different n_c-values in Example 1.

cases, whereas for the linear regression, $W = I_n$. This indicates that the logistic regression always has less information than the linear regression does. Thus, compared with the linear regression, we need a larger sample size for the logistic regression in order to have the same level of estimation accuracy.

We conclude that in order to achieve the restricted model-selection consistency property, a small n_c rate should be chosen, as long as the size of the construction sample is large enough to provide accurate estimates. Despite the above comparison, the optimal n_c rates may change for different settings. However, CV(n_v), with a wide range of n_c values, always outperforms the ten-fold CV, as well as the AIC and BIC.

In contrast to Shao (1993, 1996), we study a high-dimensional variable selection problem, which leads to fundamental technical differences. We allow the number of candidate models to diverge, as stated in Condition 3 below, whereas in Shao (1993, 1996), this quantity is a fixed constant.

4. Theory

Before presenting the theory, we introduce several conditions.

Condition 1. *The set $\mathcal{A}_c$ is nonempty.*

Condition 1 is usually satisfied when the penalty parameter λ is sufficiently small, and it ensures that the problem we are trying to solve is not degenerate.

Condition 2 (Beta-min)**.** *For the true model $\mathcal{O}$, let $\sigma^2 = \operatorname{var}(y)$. Here, we assume*

$$\beta_* = \min_{j \in \mathcal{O}} \left|\beta_j^o\right| \gg \sigma \sqrt{\frac{\log p}{n}}.$$

Condition 2 is common in the high-dimensional sparse recovery literature and guarantees that the signal variables are detectable from the noise variables. If $p = O(\exp(n^a))$, $0 < a < 1$, then $\beta_* = \Theta(1)$ is sufficient to satisfy this condition. In fact, β_* can tend to zero slowly as n and p diverge.

Condition 3 (Candidate set)**.** *Denote $d_{\max} = \max_{\alpha \in \mathcal{A}_c} |\alpha|$, $d^* = \max\{d_{\max} - d_*, d_*\}$. Assume $n_c d^* \ll n$ and*

$$R = o\left(\exp\left(\frac{n}{n_c d^*}\right)\right). \tag{4.1}$$

Condition 3 ensures that the candidate set is well behaved. We allow the number of candidate models to diverge, as long as $n_c d^* \ll n$. For instance, if d^* is bounded and $n_c = O(n^{1/2})$, then $R = o(\exp(n^{1/2}))$.

In the fixed-p scenario, the candidate set can be all possible 2^p models. When we allow both p and n to diverge, we are aware that the number of the candidate models increases. However, in practice, this is usually a fixed number, say, $R = 100$, in the default setting in the glmnet package in R. We can control an increasing number of candidate models by exploiting concentration inequalities. Condition 3 gives the limit of this quantity.

Condition 4 (Generalized linear model properties)**.** *(i) Assume that $b(\cdot)$ has continuous first-, second- and third-order derivatives $\dot{b}(\cdot)$, $\ddot{b}(\cdot)$, and $\dddot{b}(\cdot)$, respectively; in addition, $\ddot{b}(\cdot) > 0$; (ii) there exists a function $h(\cdot)$ and $\epsilon_0 > 0$ such that, for any $\alpha \in \mathcal{A}_c$ and $\boldsymbol{\eta}_\alpha \in \{\boldsymbol{\zeta}_\alpha : \|\boldsymbol{\zeta}_\alpha - \boldsymbol{\beta}_\alpha\| \le \epsilon_0\}$, we have $E(h(\boldsymbol{x})) < \infty$, $E(h_\alpha(\boldsymbol{x}_\alpha)) < \infty$, $\|\ddot{b}(\boldsymbol{x}_\alpha^\top \boldsymbol{\eta}_\alpha)\|^2 \le h_\alpha(\boldsymbol{x}_\alpha)$, and $\|\dddot{b}(\boldsymbol{x}_\alpha^\top \boldsymbol{\eta}_\alpha)\|^2 \le h_\alpha(\boldsymbol{x}_\alpha)$, where $h_\alpha(\cdot)$ is the function $h(\cdot)$ restricted to the subspace spanned by $\boldsymbol{x}_\alpha$.*

This is a mild condition for generalized linear models. For example, it is easy to verify that the linear regression model satisfies Condition 4, because $b(\theta) = \theta^2/2$, in which case the function $h(\cdot)$ can be set as a constant function.

Condition 5 (Invertibility condition)**.** *There exist $c_* > 0$ and $q^* = \Theta(\sqrt{n/\log p})$, such that for all $A \subset \{1, \ldots, p\}$ with $|A| = q^* \ge d_* \ge d_0$, and for any $\boldsymbol{\eta}_A \in \{\boldsymbol{\zeta}_A : \|\boldsymbol{\zeta}_A - \boldsymbol{\beta}_A\| \le \varepsilon_0\}$, where $\varepsilon_0 > 0$ is fixed, and if $\boldsymbol{v} \ne \mathbf{0}$ is a q^*-dimensional vector, we have,*

$$\operatorname{pr}\left\{ c_* \le \Big\| \frac{\big(\ddot{b}(X_A \boldsymbol{\eta}_A)\big)^{1/2} X_A \boldsymbol{v}\Big\|^2}{n\|\boldsymbol{v}\|^2} \right\} \to 1, \quad n \to \infty.$$

This condition indicates that in any manifold of dimension less than or equal to q^*, its corresponding restricted maximum likelihood estimator is well-defined and unique. This is a weaker version of the sparse Riesz condition (Zhang and Huang (2008)), in which both the upper and the lower bounds are required. The sparse Riesz condition (or a similar condition) is imposed in the existing literature on the tuning parameter selection consistency using information criteria (Zhang, Li and Tsai (2010)). With the invertibility condition, we can safely terminate the evaluation on the path when the current model size exceeds q^*, without the risk of missing the optimal model.

Condition 6 (Design matrix)**.** *For all $A \subset \{1, \ldots, p\}$, with $|A| = q^*$, where q^* is defined in Condition 5, and for any $\boldsymbol{\eta}_A \in \{\boldsymbol{\zeta}_A : \|\boldsymbol{\zeta}_A - \boldsymbol{\beta}_A\| \le \epsilon_0\}$, where $\epsilon_0 > 0$ is a given constant, the following is satisfied:*

$$\max_{s \in \mathcal{S}} \left\| \frac{1}{n_v} (X_s^A)^\top \ddot{b}(X_s^A \boldsymbol{\eta}_A)(X_s^A) - \frac{1}{n_c} (X_{s^c}^A)^\top \ddot{b}(X_{s^c}^A \boldsymbol{\eta}_A) X_{s^c}^A \right\| = o_p(1),$$

where the norm is the operator norm of the matrices, $s^c = \{1, \ldots, n\} \setminus s$, *and* $\mathcal{S}$ *is the collection of splits.*

This condition bounds the difference between the Fisher information of the validation set and the construction set. This is a reasonably mild condition. The technical details of its corresponding version for linear models are discussed in Section 4.4 of Shao (1993).

Theorem 1. *For penalized generalized linear models with separable sparse-inducing penalties, assume Conditions* $1-6$ *hold, where* $n_c/n \to 0$, $n_c \to \infty$, *and the number of the splits* K *satisfies*

$$K^{-1} n_c^{-2} n^2 \to 0.$$

Then, CV(n_v*) achieves restricted model-selection consistency.*

In Theorem 1, we do not explicitly specify the order of p as a condition. However, the restriction on the dimensionality is implied by Conditions 2 and 3. The ultra-high-dimensional setting where $p = O(\exp(n^a))$, for $0 < a < 1$, is allowed. Theorem 1 can be derived easily from Lemma 2 in the Supplementary Material because CV(n_v), as described in Algorithm 2, reduces a potentially high-dimensional problem to a low-dimensional one. Now, we can use the unpenalized solution in S4 of Algorithm 2 to improve the estimation and prediction performance.

5. Numerical Experiments

In this section, we compare the proposed CV(n_v) with several popular tuning parameter selection methods, including the K-fold CV (K-fold), K-fold CV with one standard error rule (1SE), AIC, BIC, and EBIC. Here, we investigate both the linear regression and the logistic regression with different correlation structures among the covariates.

Before presenting the results of the tuning parameter selection, we examine the behavior of the collections of solutions generated by different splits in the CV procedure.

5.1. Coherent rate

Example 1. (i) Linear regression. For $i = 1, \ldots, n$, let $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}^o + \varepsilon_i$, where $\boldsymbol{x}_i \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}_p, \Sigma)$, with $\mathbf{0}_p$ a length-p vector with all-zero entries and $\Sigma_{j,k} = \rho^{|j-k|}$, $\varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$, $\rho = 0.5$ and $(n, p) = (500, 10{,}000)$. In addition, $\boldsymbol{\beta}^o \in \mathbb{R}^p$ for the first nine coordinates (0.8, 0, 0.7, 0, 0.6, 0, 0.5, 0, 0.4), and is zero elsewhere. (ii)

Logistic regression. For $i = 1, \ldots, n$, y_i satisfies $\text{pr}(y_i = 1) = \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta}^o)/\{1 + \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta}^o)\} = 1 - \text{pr}(y_i = 0)$, where $\boldsymbol{\beta}^o \in \mathbb{R}^p$ for the first nine coordinates (1.6, 0, 1.4, 0, 1.2, 0, 1.0, 0, 0.8), and is zero elsewhere. The remaining part of the simulation setting is the same as that in (i).

Suppose the sequence of tuning parameters for the complete data set is $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_R)$. Here, we employ a variant of the ten-fold CV by repeatedly splitting the complete data set $K = 100$ times into 9/10 fractions as a construction set, and using the remaining 1/10 fraction as the validation set. Denote the collection of validation sets as $\{s_k, k = 1, \ldots, K\}$ and the collection of construction sets as $\{s_{(-k)}, k = 1, \ldots, K\}$. We also denote $s_0 = \{1, \ldots, n\}$ as the complete sample, as a reference. Denote by $\alpha_r^{(k)}$ the model of the r-th location in the collection of solutions constructed by subset $s_{(-k)}$ using its corresponding tuning parameter sequence $\boldsymbol{\lambda}^{(-k)}$, where $r = 1, \ldots, R$, and $k = 0, 1, \ldots, K$. We define the coherent rate as a sequence representing the degree of agreement of the models across different splits for each tuning parameter location, as follows:

$$\text{CR}(r) = \big|\{k = 1, \ldots, K : \alpha_r^{(k)} = \alpha_r^{(0)}\}\big|/K, \quad r = 1, \ldots, R.$$

In the ideal case, where $\text{CR}(r) = 1$, for all $r = 1, \ldots, R$, the CV method for choosing the tuning parameter may serve as a good surrogate for selecting the optimal model. However, this is rarely true in practice, especially after the noise variables are activated in the estimators. Next, we demonstrate the behavior of the coherent rate.

For the setting in Example 1, we calculate the collection of solutions using the R package glmnet for the Lasso, and the use the R package ncvreg for the SCAD (Fan and Li (2001)) and MCP (Zhang (2010)). Figure 3 shows how the coherent rate changes along the path in different scenarios. In addition, we mark the location of the ten-fold CV chosen estimator and the first location where noise variables are selected. It is clear that the coherent rate is much smaller than one at most locations. Note that there exists a small segment where the coherent rate is equal to one for all penalties in the linear regression; furthermore, this segment is longer for the SCAD and MCP than it is for the Lasso. Only on the corresponding segment does the CV average over the same model across different splits. Unfortunately, the ten-fold CV always selects a model with a coherent rate equal to zero (marked by the solid vertical line in Figure 3). In the logistic regression, all penalties lead to a very small coherent rate, even before the noise variables are selected.

From the path-generating procedure, the estimators and tuning parameters

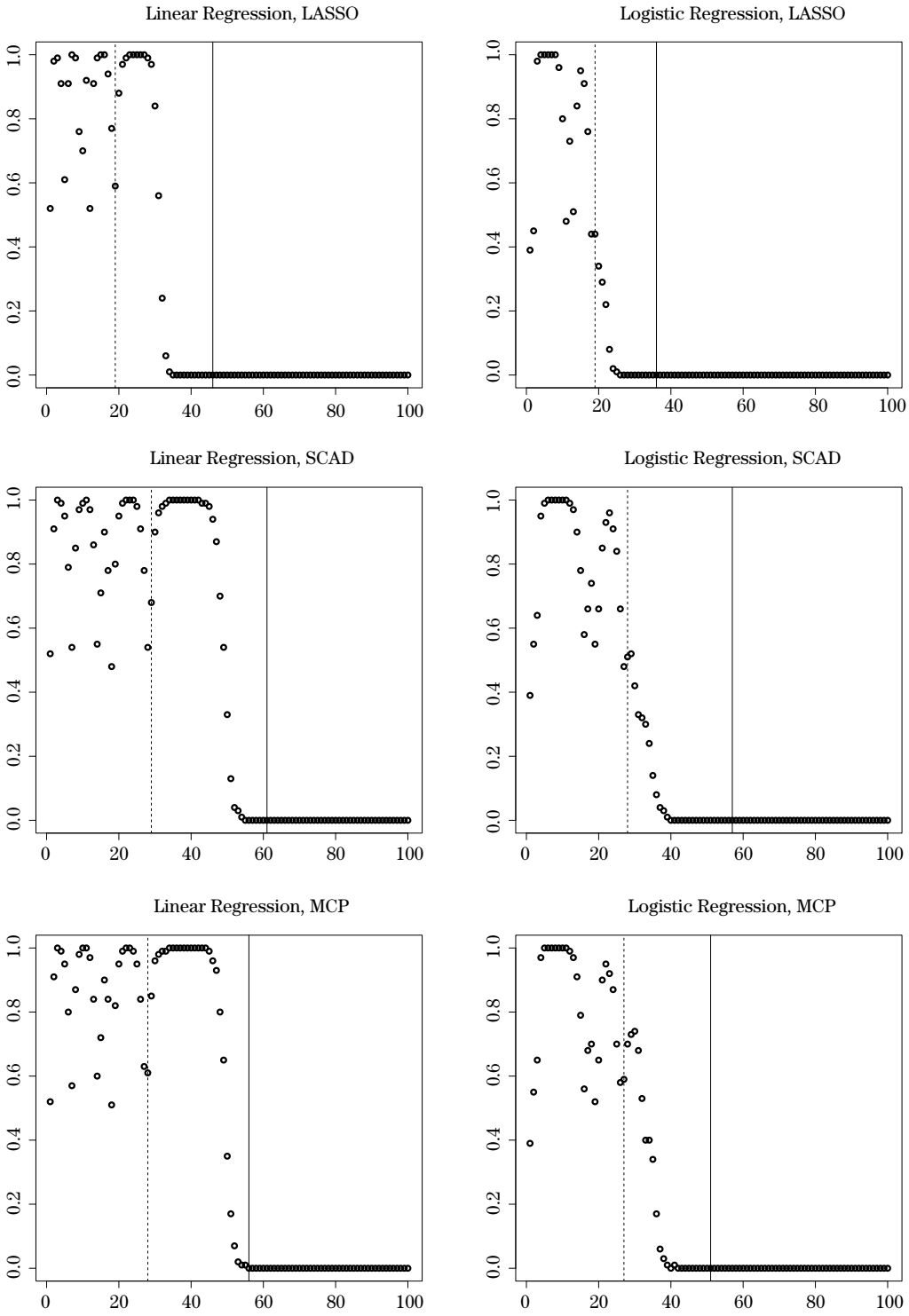

Figure 3. The coherent rate along the path for the Lasso, SCAD, and MCP penalized linear and logistic regression estimators in Example 1. x-axis: the location in the collection of the solutions; y-axis: the coherent rate. The solid line "——" is the selection of the ten-fold CV, and the dotted line "- - -" is where noises start to be selected.

can be regarded as functions of each other, given the data. Therefore, the phenomenon noted above is due to the data-driven property of the tuning parameter selection. When the data change from the complete sample to different subsample splits, the tuning parameter sequence is usually different, which naturally leads to possibly distinct models. In order to hold the models the same, very stringent conditions need to be imposed on the design matrix. However, these are usually not satisfied, even for the simple simulation settings shown here.

5.2. Linear regression

For the linear regression, we use the same settings as those in Example 1 (i), with $\rho = 0$ and $\rho = 0.5$, and repeat the simulation 100 times. Here, the signal-to-noise ratios for the two settings are 1.9 and 4, respectively. For the SCAD and MCP paths, we use the default $\gamma = 3$ in the ncvreg package. In Table 1, for CV(n_v), we set $n_c = \lceil n^{1/2} \rceil = 23$ and $n_v = n - n_c = 477$. We compare our results with those of the ten-fold CV in glmnet and ncvreg. We also include a comparison between the results of the ten-fold CV and those of the 1SE, where λ is chosen as the maximum result with a loss function value less than the minimum loss function value plus its standard error. In addition, we report the performance of popular information criteria, including the AIC, BIC, and EBIC. To compare these methods, we report the FN, FP and prediction error (PE) evaluated on an independent test data set of size n.

In Table 1, for the Lasso penalty, the AIC and ten-fold CV have the largest mean FP, followed by those of the BIC, 1SE, and EBIC. CV(n_v) performs best in terms of the FP, FN and PE in both $\rho = 0$ and $\rho = 0.5$.

The SCAD and MCP show similar performance to Lasso-based methods. The FPs of the ten-fold CV are not as large as those of the Lasso, but CV(n_v) still outperforms the ten-fold CV and AIC in terms of both variable selection and prediction. Note that the difference is not as significant as that in the Lasso case, possibly because of the asymptotic unbiasedness property of the SCAD and MCP (Zhang (2010)). Similarly, when using BIC and EBIC, the SCAD and MCP outperform the Lasso method.

We also compare the λ-values derived from the universal thresholding (Donoho and Johnstone (1994)) $\lambda_{\text{univ}} = \sigma\sqrt{2(\log p)/n}$, where σ is the error standard deviation, and the λ-values from various methods (see Table 3) under the uncorrelated design ($\rho = 0$). The rationale for the universal thresholding is a theoretical upper bound of the maximum of all of the errors. Hence, it can be regarded as a theoretical lower bound of λ for removing all noise variables. We

Table 1. Comparisons for Example 1(i) with $\rho = 0$ and $\rho = 0.5$. Results are reported in the form of the mean (standard error). For CV(n_v), $n_c = \lceil n^{1/2} \rceil$ and $K = 50$; FP, false positive; FN, false negative; PE, prediction error.

Method		$\rho = 0$			$\rho = 0.5$	
Lasso	FP	FN	PE	FP	FN	PE
CV(n_v)	0.01(0.01)	0.00(0.00)	1.01(0.01)	0.07(0.03)	0.04(0.02)	1.02(0.01)
ten-fold	48.39(3.99)	0.00(0.00)	1.12(0.01)	30.72(3.04)	0.00(0.00)	1.09(0.01)
1SE	3.31(0.81)	0.00(0.00)	1.19(0.01)	1.51(0.37)	0.00(0.00)	1.16(0.01)
AIC	497.32(1.23)	0.00(0.00)	1.38(0.01)	471.54(1.36)	0.00(0.00)	1.37(0.01)
BIC	2.04(0.21)	0.00(0.00)	1.16(0.01)	1.75(0.15)	0.00(0.00)	1.12(0.01)
EBIC	0.58(0.08)	0.00(0.00)	1.18(0.01)	0.90(0.09)	0.00(0.00)	1.13(0.01)
SCAD	FP	FN	PE	FP	FN	PE
CV(n_v)	0.02(0.01)	0.00(0.00)	1.01(0.01)	0.05(0.02)	0.00(0.00)	1.01(0.01)
ten-fold	24.50(2.80)	0.00(0.00)	1.03(0.01)	21.74(2.37)	0.00(0.00)	1.03(0.01)
1SE	0.48(0.12)	0.00(0.00)	1.08(0.01)	0.21(0.05)	0.00(0.00)	1.08(0.01)
AIC	42.19(2.60)	0.00(0.00)	1.03(0.01)	27.02(1.89)	0.04(0.02)	1.07(0.02)
BIC	0.94(0.11)	0.00(0.00)	1.04(0.01)	0.77(0.11)	0.04(0.02)	1.08(0.02)
EBIC	0.30(0.05)	0.00(0.00)	1.05(0.01)	0.22(0.05)	0.04(0.02)	1.09(0.02)
MCP	FP	FN	PE	FP	FN	PE
CV(n_v)	0.04(0.02)	0.00(0.00)	1.01(0.01)	0.06(0.02)	0.01(0.01)	1.01(0.01)
ten-fold	4.87(0.66)	0.00(0.00)	1.02(0.01)	5.25(0.66)	0.00(0.00)	1.02(0.01)
1SE	0.01(0.01)	0.00(0.00)	1.07(0.01)	0.02(0.02)	0.01(0.01)	1.07(0.01)
AIC	87.14(0.45)	0.00(0.00)	1.18(0.01)	80.23(0.75)	0.00(0.00)	1.16(0.01)
BIC	1.20(0.90)	0.00(0.00)	1.02(0.01)	1.43(0.94)	0.00(0.00)	1.02(0.01)
EBIC	0.05(0.02)	0.00(0.00)	1.02(0.01)	0.06(0.02)	0.00(0.00)	1.02(0.01)

observe from the table that only CV(n_v) yields a λ-value larger than λ_{univ}. On the other hand, note that the lowest signal level in this example is 0.4, which can serve as an upper bound of λ in order to retain all important variables. This analysis explains the good performance of CV(n_v).

Next, we consider an additional simulation setting, described in the following example.

Example 2. Linear regression. For $i = 1, \ldots, n$, let $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}^o + \varepsilon_i$, where $\boldsymbol{x}_i \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}_p, \Sigma)$, with $\mathbf{0}_p$ a length-p vector with all-zero entries and $\Sigma_{j,k} = \rho^{|j-k|}$, $\varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$, $\rho = 0$ or 0.5, and $(n,p) = (500, 10{,}000)$. In addition, $\boldsymbol{\beta}^o \in \mathbb{R}^p$ for the first seven coordinates (1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4), and is zero elsewhere.

Note that this is a more challenging scenario than that in Example 1(i) because there are more signal variables, and the correlations between the signal variables are stronger when $\rho = 0.5$. The results for Example 2 are shown in Table 2. For $\rho = 0$ with the Lasso penalty, CV(n_v) performs significantly better

Table 2. Comparisons for Example 2 with $\rho = 0$ and $\rho = 0.5$. Results are reported in the form of the mean (standard error). For CV(n_v), $n_c = \lceil n^{1/2} \rceil$ and $K = 50$; FP, false positive; FN, false negative; PE, prediction error.

Method	$\rho = 0$			$\rho = 0.5$		
Lasso	FP	FN	PE	FP	FN	PE
CV(n_v)	0.02(0.01)	0.01(0.01)	1.01(0.01)	0.03(0.02)	0.07(0.03)	1.02(0.01)
ten-fold	73.56(5.13)	0.00(0.00)	1.17(0.01)	32.47(3.33)	0.00(0.00)	1.09(0.01)
1SE	7.44(0.83)	0.00(0.00)	1.23(0.01)	0.99(0.36)	0.00(0.00)	1.15(0.01)
AIC	484.59(1.39)	0.00(0.00)	1.41(0.01)	402.84(1.19)	0.00(0.00)	1.31(0.01)
BIC	3.41(0.26)	0.00(0.00)	1.24(0.01)	1.10(0.14)	0.00(0.00)	1.11(0.01)
EBIC	0.76(0.10)	0.00(0.00)	1.28(0.01)	0.26(0.05)	0.00(0.00)	1.12(0.01)
SCAD	FP	FN	PE	FP	FN	PE
CV(n_v)	0.01(0.01)	0.01(0.01)	1.01(0.01)	0.04(0.02)	0.07(0.03)	1.02(0.01)
ten-fold	19.80(2.35)	0.00(0.00)	1.02(0.01)	22.99(1.78)	0.00(0.00)	1.03(0.01)
1SE	0.20(0.06)	0.00(0.00)	1.08(0.01)	1.13(0.23)	0.02(0.01)	1.07(0.01)
AIC	214.58(1.31)	0.00(0.00)	1.49(0.02)	34.73(2.48)	0.00(0.00)	1.03(0.01)
BIC	0.82(0.11)	0.00(0.00)	1.04(0.01)	1.05(0.16)	0.01(0.01)	1.05(0.01)
EBIC	0.19(0.04)	0.00(0.00)	1.04(0.01)	0.30(0.06)	0.02(0.01)	1.06(0.01)
MCP	FP	FN	PE	FP	FN	PE
CV(n_v)	0.01(0.01)	0.01(0.01)	1.01(0.01)	0.04(0.02)	0.07(0.03)	1.02(0.01)
ten-fold	6.46(1.01)	0.00(0.00)	1.02(0.01)	7.16(0.76)	0.00(0.00)	1.03(0.01)
1SE	0.01(0.01)	0.00(0.00)	1.07(0.01)	0.05(0.03)	0.06(0.02)	1.07(0.01)
AIC	102.30(0.48)	0.00(0.00)	1.81(0.02)	46.67(1.04)	0.00(0.00)	1.06(0.01)
BIC	100.72(1.13)	0.00(0.00)	1.80(0.02)	0.66(0.19)	0.01(0.01)	1.03(0.01)
EBIC	0.06(0.03)	0.00(0.00)	1.02(0.01)	0.07(0.03)	0.02(0.01)	1.03(0.01)

Table 3. Comparison of λ-values derived from various methods for Example 2 (i) with $\rho = 0$. Results are presented in the form of the mean (standard error).

Universal	CV(n_v)	ten-fold	1SE	AIC	BIC	EBIC
0.19	0.20(0.02)	0.12(0.02)	0.18(0.02)	0.01(0.00)	0.17(0.01)	0.18(0.01)

than all competing methods in terms of both the FP and the PE. We observe similar results for the other settings.

5.3. Logistic regression

For the logistic regression, we use the setting in Example 1 (ii) with $\rho = 0$ and $\rho = 0.5$, and repeat the simulation 100 times. In Table 4, for CV(n_v), we set $n_c = \lceil n^{3/4} \rceil = 106$, following the results in Section 3.2. In contrast to the linear case, instead of reporting the PE, we report the classification error (CE), which is defined as the average classification error evaluated on an independent test data set of size n. The remaining settings and packages used are the same

Table 4. Comparison of logistic regression with $\rho = 0$ and $\rho = 0.5$. Results are reported in the form of the mean (standard error). For CV(n_v), $n_c = \lceil n^{3/4} \rceil$ and $K = 50$; FN, false negative; FP, false positive; CE, classification error.

Method		$\rho = 0$			$\rho = 0.5$	
Lasso	FP	FN	CE(%)	FP	FN	CE(%)
CV(n_v)	1.63(0.14)	0.01(0.01)	19.34(0.20)	0.92(0.10)	0.10(0.03)	16.06(0.20)
ten-fold	95.86(4.60)	0.00(0.00)	20.81(0.23)	87.76(4.28)	0.01(0.01)	17.22(0.20)
1SE	21.48(2.05)	0.00(0.00)	19.44(0.20)	15.60(1.67)	0.03(0.02)	16.19(0.18)
AIC	21.63(1.34)	0.00(0.00)	19.50(0.20)	20.01(1.56)	0.02(0.01)	16.28(0.19)
BIC	1.88(0.14)	0.08(0.03)	19.49(0.19)	1.75(0.16)	0.05(0.02)	16.14(0.18)
EBIC	0.46(0.07)	0.16(0.04)	19.72(0.20)	0.60(0.08)	0.11(0.03)	16.25(0.18)
SCAD	FP	FN	CE(%)	FP	FN	CE(%)
CV(n_v)	1.84(0.13)	0.02(0.01)	19.48(0.22)	1.17(0.12)	0.08(0.03)	16.21(0.19)
ten-fold	55.05(2.06)	0.00(0.00)	19.22(0.20)	52.40(1.93)	0.02(0.01)	16.72(0.19)
1SE	10.88(0.76)	0.00(0.00)	19.34(0.19)	8.29(0.70)	0.03(0.02)	16.40(0.18)
AIC	31.24(1.25)	0.00(0.00)	19.20(0.20)	23.84(1.34)	0.06(0.02)	16.48(0.19)
BIC	3.23(0.26)	0.03(0.02)	19.61(0.19)	2.17(0.20)	0.08(0.03)	16.41(0.19)
EBIC	0.92(0.10)	0.11(0.03)	19.90(0.19)	0.77(0.08)	0.10(0.03)	16.54(0.19)
MCP	FP	FN	CE(%)	FP	FN	CE(%)
CV(n_v)	2.08(0.12)	0.02(0.01)	19.76(0.20)	1.36(0.10)	0.06(0.02)	16.60(0.19)
ten-fold	13.10(0.79)	0.00(0.00)	18.84(0.21)	13.31(0.87)	0.03(0.02)	16.23(0.19)
1SE	0.91(0.14)	0.06(0.02)	19.13(0.19)	1.09(0.22)	0.10(0.03)	16.25(0.19)
AIC	19.38(1.08)	0.00(0.00)	18.92(0.20)	33.39(1.02)	0.03(0.02)	16.86(0.20)
BIC	2.51(0.23)	0.03(0.02)	18.88(0.20)	2.16(0.27)	0.08(0.03)	16.06(0.18)
EBIC	0.49(0.07)	0.07(0.03)	18.97(0.19)	0.32(0.06)	0.13(0.03)	16.16(0.18)

as those in the linear regression case.

In Table 4, CV(n_v) significantly outperforms the ten-fold CV and AIC in terms of FP. The difference is more significant than that in the linear regression case, which uses the SCAD or MCP. For the 1SE with the Lasso, the logistic regression leads to significantly more FPs than the linear regression. Furthermore, the EBIC tends to work better than the AIC and BIC do, showing similar performance to the CV(n_v) when the SCAD and MCP are applied. When evaluated using the CE, CV(n_v) still performs best in most scenarios.

6. Data Analysis

We now illustrate two applications of the proposed method using data sets on eye disease gene expressions (Scheetz et al. (2006)) and leukemia (Golub et al. (1999)).

In the eye disease gene expression data set, in order to harvest tissue from the eyes for subsequent microarray analyses, 120 12-week-old male rats were selected.

Table 5. Model size and prediction error for the eye disease gene expression data sets. Results are reported in the form of the mean (standard error).

	Lasso		SCAD		MCP	
Method	Size	PE	Size	PE	Size	PE
CV(n_v)	2.46(0.08)	0.01(0.00)	2.23(0.07)	0.01(0.00)	2.36(0.07)	0.01(0.00)
ten-fold	61.18(1.68)	0.01(0.00)	33.54(0.59)	0.01(0.00)	11.12(0.30)	0.01(0.00)
1SE	31.03(1.16)	0.01(0.00)	24.84(0.71)	0.01(0.00)	5.39(0.31)	0.01(0.00)
AIC	103.02(0.48)	0.01(0.00)	0.37(0.05)	0.02(0.00)	5.38(0.25)	0.01(0.00)
BIC	99.99(0.71)	0.01(0.00)	0.17(0.04)	0.02(0.00)	4.65(0.25)	0.01(0.00)
EBIC	1.03(0.24)	0.02(0.00)	0.02(0.01)	0.02(0.00)	1.90(0.13)	0.01(0.00)

Table 6. Model size and test classification error for the leukemia data sets. Results are reported in the form of the mean (standard error).

	Lasso		SCAD		MCP	
Method	Size	CE(%)	Size	CE(%)	Size	CE(%)
CV(n_v)	8.93(0.58)	8.07(0.75)	10.14(0.57)	7.80(0.79)	5.62(0.14)	8.33(0.76)
ten-fold	21.52(0.41)	5.85(0.74)	17.04(0.31)	7.45(0.75)	5.07(0.14)	9.84(0.94)
1SE	12.51(0.46)	8.87(0.93)	11.70(0.43)	9.84(0.92)	2.85(0.16)	14.54(1.25)
AIC	16.17(0.39)	6.65(0.76)	1.00(0.05)	30.59(1.37)	4.14(0.17)	10.02(0.98)
BIC	4.32(0.30)	17.20(1.14)	0.91(0.06)	30.59(1.37)	3.56(0.15)	10.37(1.01)
EBIC	0.48(0.06)	31.29(1.33)	0.37(0.05)	31.56(1.32)	1.46(0.08)	14.72(1.03)

The microarrays used to analyze the RNA from the eyes of these animals contain more than 31,042 different probe sets (Affymetric GeneChip Rat Genome 230 2.0 Array). The intensity values were normalized using the robust multichip averaging method (Irizarry et al., 2003) to obtain summary expression values for each probe set. The gene expression levels were analyzed on a logarithmic scale.

Following Huang et al. (2010) and Fan et al. (2011), we are interested in identifying the genes related to the TRIM32 gene, which was recently found to cause Bardet–Biedl syndrome (Chiang et al., 2006), a genetically heterogeneous disease of multiple organ systems, including the retina. Although more than 30,000 probe sets are represented on the Rat Genome 230 2.0 Array, we focused on the 18,975 probes expressed in the eye tissue.

This leukemia data set was previously analyzed in Golub et al. (1999). There are $p = 7,129$ genes, and $n = 72$ samples from two classes: 47 in class ALL (acute lymphocytic leukemia), and 25 in class AML (acute myelogenous leukemia).

We modeled these two problems using a linear and a logistic regression, respectively. In the eye gene expression data set, we randomly chose 100 out of 120 observations from the sample without replacement, which we used as

training data. We used the remaining sub-sample of size 20 as the test data. In the leukemia data set, we randomly chose 60 out of 72 observations from the sample, without replacement, as the training data, and used the remaining observations as the test data.

We repeated this procedure 100 times. The results are reported in Tables 5 and 6 in the form of the mean (standard error). For each split, we used glmnet and ncvreg to compute the Lasso and SCAD/MCP collections of solutions, respectively. Then, we compared our proposed CV(n_v) with the ten-fold CV, which is the default tuning parameter selection method in glmnet and ncvreg. In addition, we investigated the performance of the 1SE, AIC, BIC, and EBIC.

For the eye disease gene expression data sets, Table 5 shows that CV(n_v) performs well compared with the ten-fold CV and the information-type criteria. In terms of model size, the EBIC leads to the smallest model, on average, when using the Lasso penalty. However, it probably missed some important predictors because the prediction error is larger than those of the other methods. Of the models that give the best prediction error, CV(n_v) always selects the sparsest model. Similar behaviors are evident for the SCAD and MCP, although the differences in performance are not as pronounced.

For the leukemia data set, we can see from Table 6 that both the BIC and the EBIC select very small models with a large test classification error for all three penalties. The CV(n_v) tends to provide a reasonably good balance between the complexity of the model and the test classification error. Although the ten-fold CV has a smaller test classification error for the Lasso and the SCAD, it selects many more variables, on average.

7. Discussion

In this study, we applied CV methods to the tuning parameter selection problem in high-dimensional penalized generalized linear models. For the K-fold CV, we showed that the mis-alignment for different splits is one possible reason of over-selection. We advocate using the CV(n_v) with a proper choice of n_v for the path algorithms, which was shown to be restricted model selection consistent in high-dimensional settings.

Future research should examine the theoretical implications of the low coherent rate of the CV, as demonstrated in the numerical results, on the model selection performance. In addition, the proposed algorithm is a general framework, which can be extended to using other methods (e.g., forward regression)

to generate the collection of solutions. It would also be interesting to extend the methodology and the associated theory to other models, including additive models and the Cox proportional hazards models, among others. In addition, we are interested in selecting the concavity parameter γ in folded-concave penalties using CV.

An implementation of the CV(n_v) method for high-dimensional variable selection is available at `https://github.com/statcodes/rccv`.

Supplementary Material

The online supplementary material includes all technical details and additional simulation results.

Acknowledgements

The authors would like to thank the co-Editor Professor Hsin-Cheng Huang, the AE, and three referees for their insightful comments, which have greatly improved the scope and quality of the paper. This research was partially supported by NSF CAREER grant DMS-1554804.

References

Breheny, P. and Huang, J. (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics* **5**, 232–253.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for High-Dimensional Data*. Springer-Verlag New York Inc.

Chen, J. and Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika* **95**, 759–71.

Chen, S. and Donoho, D. (1994). Basis pursuit. In *1994 Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers*, IEEE **1**, 41–4.

Chiang, A. P., Beck, J. S., Yen, H.-J., Tayeh, M. K., Scheetz, T. E., Swiderski, R. E., Nishimura, D. Y., Braun, T. A., Kim, K.-Y. A., Huang, J., Elbedour, K., Carmi, R., Slusarski, D. C., Casavant, T. L., Stone, E. M. and Sheffield, V. C. (2006). Homozygosity mapping with SNP arrays identifies TRIM32, an E3 ubiquitin ligase, as a Bardet–Biedl syndrome gene (BBS11). *Proceedings of the National Academy of Sciences* **103**, 6287–92.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81**, 425–455.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *Biometrika* **78**, 316–31.

Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *Journal of the American statistical Association* **81**, 461–70.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *The Annals of statistics* **32**, 407–499.

Fan, J., Feng, Y. and Song, R. (2011). Nonparametric independence screening in sparse ultra-high-dimensional additive models. *Journal of the American Statistical Association* **106**, 544–57.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96**, 1348–60.

Fan, J. and Lv, J. (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica* **20**, 101–148.

Fan, Y. and Tang, C. Y. (2013). Tuning parameter selection in high-dimensional penalized likelihood. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **75**, 531–552.

Friedman, J. et al. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1–22.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. and Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**, 531–37.

Huang, J., Horowitz, J. L. and Wei, F. (2010). Variable selection in nonparametric additive models. *The Annals of Statistics* **38**, 2282–313.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U. and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* **4**, 249–64.

Luo, S. and Chen, Z. (2014). Sequential Lasso cum ebic for feature selection with ultra-high dimensional feature space. *Journal of the American Statistical Association* **109**, 1229–1240.

Meinshausen, N. (2007). Relaxed Lasso. *Computational Statistics and Data Analysis* **52**, 374–93.

Park, M. Y. and Hastie, T. (2007). An l_1 regularization-path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69**, 659–677.

Scheetz, T. E., Kim, K.-Y. A., Swiderski, R. E., Philp, A. R., Braun, T. A., Knudtson, K. L., Dorrance, A. M., DiBona, G. F., Huang, J., Casavant, T. L., Sheffield, V. C. and Stone, E. M. (2006). Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences* **103**, 14429–34.

Shao, J. (1993). Linear model selection by cross-validation. *Journal of the American statistical Association* **88**, 486–94.

Shao, J. (1996). Bootstrap model selection. *Journal of the American statistical Association* **91**, 655–65.

Städler, N., Bühlmann, P. and Van De Geer, S. (2010). l1-penalization for mixture regression models. *Test* **19**, 209–256.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **39**, 44–7.

Sun, T. and Zhang, C.-H. (2012). Scaled sparse linear regression. *Biometrika* **99**, 879–898.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal*

Statistical Society: Series B (Statistical Methodology) **9**, 267–88.

Wang, H., Li, B. and Leng, C. (2009). Shrinkage tuning parameter selection with a diverging number of parameters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **71**, 671–683.

Wang, H., Li, R. and Tsai, C.-L. (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika* **94**, 553–68.

Yu, Y. and Feng, Y. (2014a). Apple: Approximate path for penalized likelihood estimators. *Statistics and Computing* **24**, 803–819.

Yu, Y. and Feng, Y. (2014b). Modified cross-validation for penalized high-dimensional linear regression models. *Journal of Computational and Graphical Statistics* **23**, 1009–1027.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics* **38**, 894–942.

Zhang, C.-H. and Huang, J. (2008). The sparsity and bias of the Lasso selection in high-dimensional linear regression. *The Annals of Statistics* **36**, 1567–94.

Zhang, Y., Li, R. and Tsai, C.-L. (2010). Regularization parameter selections via generalized information criterion. *Journal of the American statistical Association* **105**, 312–23.

Department of Statistics, Columbia University, New York, NY 10027, USA.

E-mail: yang.feng@columbia.edu

School of Mathematics, University of Bristol, Bristol BS8 1TH, UK.

E-mail: y.yu@bristol.ac.uk

(Received November 2015; accepted January 2018)

Supplementary materials for "The restricted consistency property of leave-n_v-out cross-validation for high-dimensional variable selection"

Yang Feng and Yi Yu

Columbia University and University of Bristol

The supplementary material includes all the technical details and additional simulation results.

A.1 Additional lemmas and proofs

The following Lemma is adapted from Lalley (2013). It helps us to develop the asymptotic theory where N, the size of the candidate models, is allowed to diverge with the sample size.

Lemma 1 (Gaussian concentration)**.** *Let γ be the standard Gaussian probability measure on $\mathbb{R}^n$ (that is, the distribution of a $\mathcal{N}(0, I_n)$ random vector), and let $F : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz in each variable separately relative to the Euclidean metric, with Lipschitz constant c. Then for every $t > 0$,*

$$\gamma\{|F - E_\gamma(F)| \geq t\} \leq 2\exp\left(-\frac{t^2}{c^2\pi^2}\right).$$

Lemma 2. *With $p < n$, let $\tilde{\beta}$ be the MLEs of a generalized linear model. Assume the penalty function $p(\cdot)$ is separable, and assume Conditions 1 - 6 hold. Furthermore, assume $n_c \to \infty$ and $n_c/n \to 0$ as $n \to \infty$, and the size of the splits K satisfies*

$$K^{-1}n_c^{-2}n^2 \to 0.$$

Then, CV(n_v) with K times subsampling is restricted model selection consistent.

Proof of Lemma 2. Due to the properties of generalized linear models with canonical parameter, we have

$$E(y_i \mid \boldsymbol{x}_i) = \dot{b}(\boldsymbol{x}_i^\top \boldsymbol{\beta}), \quad \sigma_i^2 = a(\phi)\ddot{b}(\boldsymbol{x}_i^\top \boldsymbol{\beta}), \quad i = 1, \cdots, n,$$

and define $\sigma^2 = (1/n)\sum_{i=1}^n \sigma_i^2$. The target is to select the model that minimizes the loss

$$\tilde{\Gamma}_\alpha = \frac{1}{Kn_v}\sum_{s\in\mathcal{S}}\Big\{-\boldsymbol{y}_s^\top(X_s^\alpha \tilde{\boldsymbol{\beta}}_{s^c,\alpha}) + \mathbf{1}^\top b(X_s^\alpha \tilde{\boldsymbol{\beta}}_{s^c,\alpha})\Big\}, \tag{1}$$

where $\mathcal{S}$ represents the collection of validation sets in different splits and $\mathbf{1}$ is an all-one vector.

Denote $E_{\mathcal{S}}$ and $\text{var}_{\mathcal{S}}$ as the expectation and variance with respect to the random selection of $\mathcal{S}$. By using the equality

$$E_{\mathcal{S}}\left(\frac{1}{r}\sum_{s\in\mathcal{S}}a_s\right)=\binom{n}{n_v}^{-1}\sum_{s\in \text{ all } s}E(a_s),$$

rewriting (1), and denoting $\ell_s(\boldsymbol{\beta})=\boldsymbol{y}_s^\top(X_s\boldsymbol{\beta})-\mathbf{1}^\top b(X_s\boldsymbol{\beta})$ and $\ell_n(\tilde{\boldsymbol{\beta}}_\alpha)=\boldsymbol{y}^\top(X_\alpha\tilde{\boldsymbol{\beta}}_\alpha)-\mathbf{1}^\top b(X_\alpha\tilde{\boldsymbol{\beta}}_\alpha)$, we have

$$\begin{aligned}
E_{\mathcal{S}}(\tilde{\Gamma}_\alpha)=&E_{\mathcal{S}}\left(-\frac{1}{Kn_v}\sum_{s\in\mathcal{S}}\ell_s(\boldsymbol{\beta}^o)\right)+E_{\mathcal{S}}\left(\frac{1}{Kn_v}\sum_{s\in\mathcal{S}}\big\{\ell_s(\boldsymbol{\beta}^o)-\big(\boldsymbol{y}_s^\top(X_s^\alpha\tilde{\boldsymbol{\beta}}_\alpha)-\mathbf{1}^\top b(X_s^\alpha\tilde{\boldsymbol{\beta}}_\alpha)\big)\big\}\right)\\
&+E_{\mathcal{S}}\left(\frac{1}{Kn_v}\sum_{s\in\mathcal{S}}\big\{\big(\boldsymbol{y}_s^\top(X_s^\alpha\tilde{\boldsymbol{\beta}}_\alpha)-\mathbf{1}^\top b(X_s^\alpha\tilde{\boldsymbol{\beta}}_\alpha)\big)-\big(\boldsymbol{y}_s^\top(X_s^\alpha\tilde{\boldsymbol{\beta}}_{s^c,\alpha})-\mathbf{1}^\top b(X_s^\alpha\tilde{\boldsymbol{\beta}}_{s^c,\alpha})\big)\big\}\right)\\
=&E\bigg(-\frac{1}{n}\ell_n(\boldsymbol{\beta}^o)+\frac{1}{n}\big(\ell_n(\boldsymbol{\beta}^o)-\ell_n(\tilde{\boldsymbol{\beta}}_\alpha)\big)\\
&+\binom{n}{n_v}^{-1}\sum_{s\in \text{ all } s}\frac{1}{n_v}\big\{\boldsymbol{y}_s^\top(X_s^\alpha\tilde{\boldsymbol{\beta}}_\alpha-X_s^\alpha\tilde{\boldsymbol{\beta}}_{s^c,\alpha})-\mathbf{1}^\top\big(b(X_s^\alpha\tilde{\boldsymbol{\beta}}_\alpha)-b(X_s^\alpha\tilde{\boldsymbol{\beta}}_{s^c,\alpha})\big)\big\}\bigg)\\
=&-\frac{1}{n}E(\ell_n(\boldsymbol{\beta}^o))+E(A_{\alpha1})+\binom{n}{n_v}^{-1}\sum_{s\in \text{ all } s}E(A_{\alpha2,s}).
\end{aligned}$$

For different α, $E(\ell_n(\boldsymbol{\beta}^o))$ stays the same, so we only need to focus on $A_{\alpha1}$ and $A_{\alpha2,s}$.

From Wilks' theorem, it is known that, if $\alpha\in\mathcal{A}\setminus\mathcal{A}_c$, as $n\to\infty$, we have $A_{\alpha1}\xrightarrow{\mathcal{D}}(1/2)\chi^2(k_\alpha)$, where $k_\alpha=d_0-d_{\alpha0}$, $d_{\alpha0}=|\{j:\beta_j\in\alpha\cap\alpha_0\}|$, i.e., k_α is the number of false negatives. This means $E(A_{\alpha1})=k_\alpha$; otherwise, $E(A_{\alpha1})=O(1/n)$.

For any s,

$$\begin{aligned}
&\mathbf{1}^\top\big(b(X_s^\alpha\tilde{\boldsymbol{\beta}}_\alpha)-b(X_s^\alpha\tilde{\boldsymbol{\beta}}_{s^c,\alpha})\big)=\big(\dot{b}(X_s^\alpha\tilde{\boldsymbol{\beta}}_\alpha)\big)^\top X_s^\alpha(\tilde{\boldsymbol{\beta}}_\alpha-\tilde{\boldsymbol{\beta}}_{s^c,\alpha})\\
&-\frac{1}{2}(\tilde{\boldsymbol{\beta}}_\alpha-\tilde{\boldsymbol{\beta}}_{s^c,\alpha})^\top(X_s^\alpha)^\top\ddot{b}(X_s^\alpha\tilde{\boldsymbol{\beta}}_\alpha)X_{s,\alpha}(\tilde{\boldsymbol{\beta}}_\alpha-\tilde{\boldsymbol{\beta}}_{s^c,\alpha})+o(1).
\end{aligned}$$

Define $u_{s^c}(\boldsymbol{\gamma})=(1/n_c)(X_{s^c}^\alpha)^\top\big(\boldsymbol{y}_{s^c}-\dot{b}(X_{s^c}^\alpha\boldsymbol{\gamma})\big)$, then $\tilde{\boldsymbol{\beta}}_{s^c,\alpha}$ is the solution to $u_{s^c}(\boldsymbol{\gamma})=0$. By Taylor expansion, we get

$$\tilde{\boldsymbol{\beta}}_\alpha-\tilde{\boldsymbol{\beta}}_{s^c,\alpha}=\big(\dot{u}_{s^c}(\tilde{\boldsymbol{\beta}}_\alpha)\big)^{-1}u_{s^c}(\tilde{\boldsymbol{\beta}}_\alpha)(1+o(1)),$$

where $\dot{u}_{s^c}(\tilde{\boldsymbol{\beta}}_\alpha)=-(1/n_c)(X_{s^c}^\alpha)^\top\ddot{b}(X_{s^c}^\alpha\tilde{\boldsymbol{\beta}}_\alpha)X_{s^c}^\alpha$.

Define $D_{s,\alpha} = \ddot{b}^{1/2}(X_s^\alpha \tilde{\boldsymbol{\beta}}_\alpha) X_{s^c}^\alpha$, then

$$\begin{aligned}
A_{\alpha 2,s} &= \frac{1}{n_v}\big(\boldsymbol{y}_s - \dot{b}(X_s^\alpha \tilde{\boldsymbol{\beta}}_\alpha)\big)^\top X_s^\alpha (\tilde{\boldsymbol{\beta}}_\alpha - \tilde{\boldsymbol{\beta}}_{s^c,\alpha}) \\
&\quad + \frac{1}{2n_v}(\tilde{\boldsymbol{\beta}}_\alpha - \tilde{\boldsymbol{\beta}}_{s^c,\alpha})^\top (X_s^\alpha)^\top \ddot{b}(X_s^\alpha \tilde{\boldsymbol{\beta}}_\alpha) X_s^\alpha (\tilde{\boldsymbol{\beta}}_\alpha - \tilde{\boldsymbol{\beta}}_{s^c,\alpha}) + o(1/n_v) \\
&= \frac{1}{n_v}\big(\boldsymbol{y}_s - \dot{b}(X_s^\alpha \tilde{\boldsymbol{\beta}}_\alpha)\big)^\top X_s^\alpha \big(\dot{u}_{s^c}(\tilde{\boldsymbol{\beta}}_\alpha)\big)^{-1} u_{s^c}(\tilde{\boldsymbol{\beta}}_\alpha) + o(1/n_v) \\
&\quad + \frac{1}{2n_v}\big(\boldsymbol{y}_{s^c} - \dot{b}(X_{s^c}^\alpha \tilde{\boldsymbol{\beta}}_\alpha)\big)^\top \big(\ddot{b}(X_s^\alpha \tilde{\boldsymbol{\beta}}_\alpha)^{-1/2}\big) D_{s,\alpha}\big(D_{s,\alpha}^\top D_{s,\alpha}\big)^{-1} \\
&\quad \times \big((X_s^\alpha)^\top \ddot{b}(X_s^\alpha \tilde{\boldsymbol{\beta}}_\alpha) X_s^\alpha\big)\big((X_{s^c}^\alpha)^\top \ddot{b}(X_s^\alpha \tilde{\boldsymbol{\beta}}_\alpha) X_{s^c}^\alpha\big)^{-1} \\
&\quad \times D_{s,\alpha}^\top \big(\ddot{b}(X_s^\alpha \tilde{\boldsymbol{\beta}}_\alpha)^{-1/2}\big)\big(\boldsymbol{y}_{s^c} - \dot{b}(X_{s^c}^\alpha \tilde{\boldsymbol{\beta}}_\alpha)\big)(1 + o(1)) \\
&= B_\alpha + C_\alpha.
\end{aligned}$$

By plugging in the expansion form of $\dot{u}_{s^c}(\cdot)$ and $u_{s^c}(\cdot)$,

$$B_\alpha = -\frac{1}{n_v}\big(\boldsymbol{y}_s - \dot{b}(X_s^\alpha \tilde{\boldsymbol{\beta}}_\alpha)\big)^\top X_s^\alpha \big((X_{s^c}^\alpha)^\top \ddot{b}(X_{s^c}^\alpha \tilde{\boldsymbol{\beta}}_\alpha) X_{s^c}^\alpha\big)^{-1} (X_{s^c}^\alpha)^\top \big(\boldsymbol{y}_{s^c} - \dot{b}(X_{s^c}^\alpha \tilde{\boldsymbol{\beta}}_\alpha)\big)(1 + o(1)).$$

From Conditions 5 and 6, straight calculations lead to

$$E(B_\alpha) = 0, \quad \operatorname{var}(B_\alpha) = d_\alpha a(\phi)(n_c n_v)^{-1/2}(1 + o(1)).$$

For C_α we have,

$$\begin{aligned}
C_\alpha &= \frac{1}{2n_c}\big(\boldsymbol{y}_{s^c} - \dot{b}(X_{s^c}^\alpha \tilde{\boldsymbol{\beta}}_\alpha)\big)^\top \big(\ddot{b}(X_s^\alpha \tilde{\boldsymbol{\beta}}_\alpha)^{-1/2}\big) D_{s,\alpha}\big(D_{s,\alpha}^\top D_{s,\alpha}\big)^{-1} D_{s,\alpha}^\top \\
&\quad \times \big(\ddot{b}(X_s^\alpha \tilde{\boldsymbol{\beta}}_\alpha)^{-1/2}\big)\big(\boldsymbol{y}_{s^c} - \dot{b}(X_{s^c}^\alpha \tilde{\boldsymbol{\beta}}_\alpha)\big)(1 + o(1)).
\end{aligned}$$

Thus, after taking expectation we have,

$$E(A_{\alpha 2,s}) = d_\alpha a(\phi)/n_c + o(1/n_c).$$

If $\alpha \in \mathcal{A} \setminus \mathcal{A}_c$,

$$\tilde{\Gamma}_{\alpha_*} - \tilde{\Gamma}_\alpha = \frac{1}{n}\big(\ell_n(\tilde{\boldsymbol{\beta}}_{\alpha_*}) - \ell_n(\tilde{\boldsymbol{\beta}}_\alpha)\big) + O(1/n_c).$$

From Lemma 1 and Condition 3, by exploiting Gaussian concentration, $\forall \varepsilon > 0$, we have

$$R \cdot \operatorname{pr}\left\{ n_c \left| \max_{\alpha \in \mathcal{A} \setminus \mathcal{A}_c} \left| \frac{1}{n}\big(\ell_n(\tilde{\boldsymbol{\beta}}_{\alpha_*}) - \ell_n(\tilde{\boldsymbol{\beta}}_\alpha)\big) \right| - E\left(\max_{\alpha \in \mathcal{A} \setminus \mathcal{A}_c} \left| \frac{1}{n}\big(\ell_n(\tilde{\boldsymbol{\beta}}_{\alpha_*}) - \ell_n(\tilde{\boldsymbol{\beta}}_\alpha)\big) \right| \right) \right| > \varepsilon \right\} \to 0.$$

The parallel result for $\alpha \in \mathcal{A}_c$ but $\alpha \neq \alpha_*$ holds similarly. Therefore, as $n \to \infty$, $\operatorname{pr}\{\hat{\alpha} \in \alpha_*\} \to 1$.

□

A.2 Additional numerical results

We conducted an additional simulation for the setting in Example 1(i) when $\rho = -0.5$ with the results summarized in Table 1. In this case, CV(n_v) works very well compared with other methods and we skip the detailed discussion since the message is very similar to the cases of $\rho = 0$ and $\rho = 0.5$.

Table 1: Comparisons in linear regression with $\rho = -0.5$. Results are reported in the form of mean (standard error). FP, false positive; FN, false negative; PE, prediction error.

Method	$\rho = -0.5$		
Lasso	FP	FN	PE
CV(n_v)	0.03(0.02)	0.02(0.01)	1.01(0.01)
K-fold	30.53(2.84)	0.00(0.00)	1.09(0.01)
1SE	1.54(0.21)	0.00(0.00)	1.15(0.01)
AIC	469.97(1.39)	0.00(0.00)	1.38(0.01)
BIC	2.18(0.17)	0.00(0.00)	1.12(0.01)
EBIC	0.91(0.10)	0.00(0.00)	1.13(0.01)
SCAD	FP	FN	PE
CV(n_v)	0.06(0.03)	0.01(0.01)	1.01(0.01)
K-fold	24.48(2.70)	0.00(0.00)	1.03(0.01)
1SE	0.30(0.09)	0.00(0.00)	1.08(0.01)
AIC	25.20(2.02)	0.05(0.03)	1.09(0.03)
BIC	0.70(0.09)	0.05(0.03)	1.10(0.03)
EBIC	0.16(0.04)	0.05(0.03)	1.11(0.03)
MCP	FP	FN	PE
CV(n_v)	0.02(0.01)	0.00(0.00)	1.01(0.01)
K-fold	4.76(0.82)	0.00(0.00)	1.02(0.01)
1SE	0.04(0.04)	0.00(0.00)	1.07(0.01)
AIC	77.29(0.96)	0.00(0.00)	1.15(0.01)
BIC	0.52(0.11)	0.00(0.00)	1.02(0.01)
EBIC	0.06(0.03)	0.00(0.00)	1.02(0.01)

Bibliography

Lalley, S. P. (2013). Concentration inequalities. URL `http://www.stat.uchicago.edu/$\sim$lalley/Courses/386/Concentration.pdf`.

Department of Statistics, Columbia University

E-mail: yang.feng@columbia.edu

School of Mathematics, University of Bristol

E-mail: y.yu@bristol.ac.uk