Supplementary materials for this article are available online.
Please go to www.tandfonline.com/r/JCGS

Modified Cross-Validation for Penalized High-Dimensional Linear Regression Models

Yi YU and Yang FENG

In this article, for Lasso penalized linear regression models in high-dimensional settings, we propose a modified cross-validation (CV) method for selecting the penalty parameter. The methodology is extended to other penalties, such as Elastic Net. We conduct extensive simulation studies and real data analysis to compare the performance of the modified CV method with other methods. It is shown that the popular K-fold CV method includes many noise variables in the selected model, while the modified CV works well in a wide range of coefficient and correlation settings. Supplementary materials containing the computer code are available online.

Key Words: Lasso; Tuning parameter selection.

1. INTRODUCTION

Variable selection is a popular tool for analyzing high-dimensional data. Tibshirani (1996) proposed Lasso, which is the ℓ_1 penalty, or equivalently Chen and Donoho (1994) proposed basis pursuit. Later, elastic net variants (Zou and Hastie 2005) and nonconvex penalties such as smoothly clipped absolute deviation (SCAD; Fan and Li 2001) and minimax concavity penalty (MCP; Zhang 2010) were proposed and widely used over the years. All of these variable-selection procedures proved to have good theoretical properties.

Besides, developing efficient algorithms for calculating the solution path of the coefficient vector as tuning parameter varies is of great importance. A vast literature on calculating the path for penalized linear regression is available. Among these, least angle regression (LARS; Efron et al. 2004), or homotopy (Osborne, Presnell, and Turlach 2000), Local Quadratic Approximation (LQA; Fan and Li 2001), Local Linear Approximation (LLA; Zou and Li 2008), Penalized Linear Unbiased Selection (PLUS; Zhang 2010), and coordinate descent methods (Fu 1998; Friedman, Hastie, and Tibshirani 2007) gained popularity these days.

After getting a path of solutions from the foregoing mentioned methods, users still need to pick one estimator from the path with different penalty levels controlled by the tuning

Yi Yu is Research Associate, Statistical Laboratory, Cambridge University, Cambridge CB30WB, UK (E-mail: *y.yu@statslab.cam.ac.uk*). Yang Feng is Assistant Professor, Department of Statistics, Columbia University, New York, NY 10027 (E-mail: *yangfeng@stat.columbia.edu*).

Journal of Computational and Graphical Statistics, Volume 23, Number 4, Pages 1009–1027
DOI: 10.1080/10618600.2013.849200

parameter. As it turns out, selecting the optimal tuning parameter is both important and difficult. There has been a line of research on using information-type criteria to select the tuning parameter. Tibshirani (1996) used generalized cross-validation (GCV) style statistics, and Efron et al. (2004) used C_p style statistics. Zou, Hastie, and Tibshirani (2007) derived a consistent estimator for the degree of freedom of Lasso, and plugged it into the C_p, AIC, and BIC criteria. But for Lasso estimators in high-dimensional setting, from simulation experience, all these traditional methods tend to overselect, due to the bias introduced by shrinkage. Chen and Chen (2008) proposed extended-Bayesian information criterion (EBIC), by adding an extra term with respect to p to the information criterion. Motivated by generalized information criterion (GIC) proposed by Nishii (1984), Zhang, Li, and Tsai (2010) extended GIC to a more general scenario, which can handle nonconvex penalized likelihood estimation. Wang, Li, and Leng (2009) conjectured that the traditional BIC-type criterion tends to overselect in high-dimensional scenarios and proposed a modified BIC-type criterion.

Another popular family of methods for selecting the tuning parameter is CV, which is a data-driven approach. A majority of theoretical work has been done for CV in the classical linear regression models. For example, leave-one-out cross-validation (CV(1)) is shown to be asymptotically equivalent to Akaike information criterion (AIC), the C_p, the jackknife, and the bootstrap (Stone 1977; Efron 1983, 1986). Shao (1993) gave rigorous proof of the inconsistency of CV(1) for linear regression model, meanwhile he provided the proper sizes of construction and validation set in leave-n_v-out cross-validation (CV(n_v)), under which CV achieves the model selection consistency. Zhang (1993) studied multifold CV and r-fold CV in linear regression models. It turned out both methods tend to select more variables than the truth under certain technical conditions. For several popular packages in R for Lasso, for example, `lars` (Efron et al. 2004), `glmnet` (Friedman, Hastie, and Tibshirani 2010), `glmpath` (Park and Hastie 2007), K-fold CV is still the default option. Researchers have realized that the regular CV in high-dimensional settings tends to be too conservative in the sense that it selects a majority of false positives (FPs). As mentioned in Zhang and Huang (2008), the theoretical justification of CV-based penalty parameter choice is unclear for model selection purposes. CV was also mentioned by Meinshausen (2007), where relaxed Lasso was proposed, which includes least angle regression-ordinary least squares (LARS-OLS; Efron et al. 2004) as a special case. In that article, the author conjectured that by using K-fold CV, the relaxed Lasso estimator is model selection consistent. The tuning parameter selection problem also exists for other types of variable selection methods, for example, the adding-noise approach in Luo, Stefanski, and Boos (2006) and Wu, Boos, and Stefanski (2007).

In this article, we aim to develop a new CV approach for selecting tuning parameter for high-dimensional penalized linear regression problems. It is noteworthy that we are not proposing a new variable selection technique, rather, the goal is to study and improve the variable selection performance for the existing tuning parameter selection methods. The contribution of the article is two-fold. (i) A thorough investigation on several popular CV methods is conducted, and they are shown to be inconsistent via simulation. (ii) A modified CV criterion is provided, which is shown to have better performance in terms of model selection and prediction under high-dimensional settings.

The rest of the article is organized as follows. We introduce the model setup and fix notations in Section 2, and propose the modified CV criterion in Section 3. Extensive

simulation studies, including various simulation settings and comparisons to the existing methods, and real data analysis are conducted in Sections 4 and 5, respectively. A short discussion is presented in Section 6.

2. MODEL SETUP

Given n observation pairs $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, n$, we consider the linear regression model

$$y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \varepsilon_i,$$

where $\boldsymbol{x}_i$'s and $\boldsymbol{\beta}$ are p-dimensional vectors, with $p \gg n$. ε_i's are iid random variables with mean 0 and variance σ^2.

Denote $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$ as the $n \times p$-dimensional design matrix and $\boldsymbol{y} = (y_1, \ldots, y_n)'$ as the response vector. We employ notations $\|\cdot\|$ and $\|\cdot\|_1$ as the ℓ_2 and ℓ_1 norms of a vector, respectively; $\|\cdot\|_0$ as the number of nonzero entries of a vector; and $\boldsymbol{\beta}$ as the true $\boldsymbol{\beta}$, satisfying $\|\boldsymbol{\beta}\|_0 = d_0 < n$. The oracle set $\{j : \beta_j \neq 0\}$ is denoted as $\mathcal{O}$. To analyze this high-dimensional problem, we adopt the popular Lasso estimator (Tibshirani 1996), that is,

$$\hat{\boldsymbol{\beta}}(\lambda) \equiv \arg\min_{\boldsymbol{\beta}} \frac{1}{2n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1.$$

Using any path algorithm, we can get a solution path $\hat{\boldsymbol{\beta}}(\lambda)$ when λ changes. Notice, the *starting point* of this article is that we have a collection of estimators on hand with the goal of choosing the optimal one among them. For the Lasso estimators, it is equivalent to study how to choose the optimal tuning parameter λ. Both λ and $\hat{\boldsymbol{\beta}}$ are functions of n, but to keep the notations simple, we omit the subscript n throughout the article.

Following the notation system developed in Shao (1993), we denote the model corresponding to tuning parameter λ as $\mathcal{M}_\lambda$ and divide the collection of models $\{\mathcal{M}_\lambda, \lambda > 0\}$ into two disjoint categories. Category I includes the models that miss at least one important variable, that is, false negative number is greater than 0 (FN > 0). Category II includes the models with all the important variables, that is, FN $= 0$. The optimal model $\mathcal{M}_{\lambda_*}$ is defined to be the model of the smallest size among Category II models, that is, the most parsimonious candidate model that contains all the important variables. If there are more than one models satisfying these two properties, we define the one with the smallest λ as the optimal one, considering the shrinkage brought in by λ. Here, λ_* is the tuning parameter of the optimal model. Now the goal is to find $\mathcal{M}_{\lambda_*}$ and the corresponding λ_*.

By exploiting the CV approach, the main idea is to repeatedly split the data into construction and validation sets, fit the model for the construction data, and evaluate the performance of the fitted model on the validation set. The prediction performance over different splits is averaged and the average can represent the predictability of the model. One can then choose the best model according to the predictability measure of different models. In Shao (1993), the average is taken over the same models estimated from different splits. However, for the case of CV in Lasso estimators, it is not always possible to perform the same averaging because one will generally get different models from different splits even if the same λ values are used. Instead of averaging the performance for the same model, we measure the performance of the models corresponding to the specific λ value.

For CV, denote s as a subset of $\{1, \ldots, n\}$ containing n_v integers for validation, and $(-s)$ as its complement containing n_c integers, where $n_v + n_c = n$. Considering the subsamples and their associated submodels will appear later, we denote $\mathcal{M}_{(-s),\lambda}$ as the model constructed from sample $(-s)$ given λ. The following notations are used throughout the article. The subscripts are based on the submatrices for the corresponding construction sets and the tuning parameter λ:

$$\boldsymbol{X}_{s,\lambda} = (x_{ij}), i \in s, j \in \mathcal{M}_{(-s),\lambda}; \quad \boldsymbol{X}_{(-s),\lambda} = (x_{ij}), i \in (-s), j \in \mathcal{M}_{(-s),\lambda};$$

$$\boldsymbol{X}_{\cdot,\lambda} = (x_{ij}), i = 1, \ldots, n, j \in \mathcal{M}_{\lambda}; \quad \boldsymbol{X}_{\cdot,\lambda(s)} = (x_{ij}), i = 1, \ldots, n, j \in \mathcal{M}_{(-s),\lambda};$$

$$\boldsymbol{H}_{s,\lambda} = \boldsymbol{X}_{s,\lambda}(\boldsymbol{X}'_{\cdot,\lambda(s)}\boldsymbol{X}_{\cdot,\lambda(s)})^{-1}\boldsymbol{X}'_{s,\lambda};$$

$\boldsymbol{y}_s = (y_i, i \in s)'$; denote $\hat{\boldsymbol{\beta}}_{(-s),\lambda}$ as the Lasso estimator of $\boldsymbol{\beta}$ under $\mathcal{M}_{(-s),\lambda}$.

3. MODIFIED CROSS-VALIDATION

To deal with the overselection issue of the traditional CV in Lasso penalized high-dimensional variable selection, a new CV method is proposed. Instead of developing a new variable selection technique, we would rather say the goal here is to investigate and improve the existing CV methods.

3.1 ALGORITHM

First, we describe a generic cross-validation algorithm for the Lasso estimators in the linear regression.

S1. Compute the Lasso solution path with the whole dataset. A sequence of solutions $\hat{\boldsymbol{\beta}}(\lambda)$ is generated with corresponding penalty level λ's.

S2. Randomly split the whole dataset into construction dataset (size n_c) and validation dataset (size n_v) b times, compute the Lasso solution path for each construction dataset with the λ sequence in S1.

S3. For each split, use the corresponding validation dataset to calculate the values of the criterion function (to be specified) for each path, and average over the paths with the same λ.

S4. Find the $\hat{\lambda}$ with the smallest average criterion value, then fit a linear regression for the model $\mathcal{M}_{\hat{\lambda}}$. This linear regression estimator is the final estimator.

The sequence of solutions $\hat{\boldsymbol{\beta}}(\lambda)$ mentioned in S1 is generated from certain Lasso path algorithm, such as glmnet used in subsequent simulations. Since the goal is to choose the optimal model from a collection of candidate models, the path generation method is not specified in the algorithm description. In S4, considering the bias caused by Lasso procedure, a further linear regression on the selected variable set is conducted after variable selection. The algorithm involves several parameters n_c, n_v, b, and the criterion function used in S3. We are interested in how these parameters and the criterion function affect the final estimator and which are the best ones.

3.2 CRITERION FUNCTION

In this subsection, we study the choice of criterion function used in S3. In the traditional CV, the criterion function to be minimized is

$$\Gamma_0(\lambda) = \frac{1}{n_v}\|\boldsymbol{y}_s - \hat{\boldsymbol{y}}_{(-s),\lambda}\|^2,$$

where $\hat{\boldsymbol{y}}_{(-s),\lambda} = \boldsymbol{X}_{s,\lambda}\hat{\boldsymbol{\beta}}_{(-s),\lambda}$ represents the predicted value on the subset s using the Lasso estimate based on the data $(-s)$ when the penalty level is λ.

Via numerical experience, researchers realized that traditional CV based on Γ_0 tends to select many FPs. This is mainly caused by the bias issue of the Lasso penalty. For convenience, we assume that all the matrix inversions appearing in the article are well defined. This is to say, that for any subset $A \subset \{1, \ldots, p\}$ with small enough size appearing in this article, $\boldsymbol{X}'_{A,A}\boldsymbol{X}_{A,A}$ is of full rank (Zhang 2010). Now, we introduce the new CV criterion *exactly modified cross-validation criterion (EMCC)*, which is defined as

$$\Gamma_1(\lambda) = \frac{1}{n_v}\|\boldsymbol{y}_s - \hat{\boldsymbol{y}}_{(-s),\lambda}\|^2 - \frac{\lambda^2 n_c^2}{n_v}\boldsymbol{M}'_{s,\lambda}\boldsymbol{M}_{s,\lambda}, \tag{1}$$

where $\boldsymbol{M}_{s,\lambda} = \boldsymbol{X}_{s,\lambda}(\boldsymbol{X}'_{(-s),\lambda}\boldsymbol{X}_{(-s),\lambda})^{-1}(\mathrm{sgn}(\hat{\boldsymbol{\beta}}_{(-s),\lambda}))$, and $\mathrm{sgn}(\cdot)$ represents the sign function.

Let $d_{(-s),\lambda}$ be the model size of the current Lasso estimator $\hat{\boldsymbol{\beta}}_{(-s),\lambda}$. If the covariates are standardized and independent, we have $E(\boldsymbol{X}'_{(-s),\lambda}\boldsymbol{X}_{(-s),\lambda}) = n_c\boldsymbol{I}_{d_{(-s),\lambda}}$ and $E(\boldsymbol{X}'_{s,\lambda}\boldsymbol{X}_{s,\lambda}) = n_v\boldsymbol{I}_{d_{(-s),\lambda}}$. Now, if we replace the sample covariance matrices by their population versions, EMCC can be approximately reduced to the following simple form:

$$\Gamma_2(\lambda) = \frac{1}{n_v}\|\boldsymbol{y}_s - \hat{\boldsymbol{y}}_{(-s),\lambda}\|^2 - \lambda^2 d_{(-s),\lambda}, \tag{2}$$

which is called *modified cross-validation criterion (MCC)*. It is clear that MCC can be easily calculated by the knowledge of the current penalty level λ and the current model size $d_{(-s),\lambda}$.

Briefly, the EMCC criterion is designed to remove the systematic bias introduced by the shrinkage. Now, we give the detailed rationale behind it. Define $\hat{\boldsymbol{\beta}}_{\cdot,\lambda}$ as the subvector of $\hat{\boldsymbol{\beta}}(\lambda)$ restricted on $\mathcal{M}_\lambda$; to make notationally consistent, we put $\cdot$ in the subscript to indicate the estimator is derived using the whole dataset. Define $\tilde{\boldsymbol{\beta}}_{\cdot,\lambda}$ as the least-square estimator (LSE) for the model $\mathcal{M}_\lambda$, that is,

$$\tilde{\boldsymbol{\beta}}_{\cdot,\lambda} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{d_{\cdot,\lambda}}}{\mathrm{argmin}}\frac{1}{2n}\|\boldsymbol{y} - \boldsymbol{X}_{\cdot,\lambda}\boldsymbol{\beta}\|^2,$$

where $d_{\cdot,\lambda}$ represents the model size for penalty level λ using all the sample. Correspondingly, $\tilde{\boldsymbol{y}}_{(-s),\lambda} = \boldsymbol{X}_{s,\lambda}\tilde{\boldsymbol{\beta}}_{(-s),\lambda}$. Recall the solution to the Karush–Kuhn–Tucker (KKT) conditions is the unique minimizer of the original optimization problem,

$$\begin{cases} \dfrac{1}{n}\boldsymbol{x}'_j(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) = \lambda\mathrm{sgn}(\hat{\beta}_j), & \hat{\beta}_j \neq 0; \\ \dfrac{1}{n}|\boldsymbol{x}'_j(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})| \leq \lambda, & \hat{\beta}_j = 0. \end{cases} \tag{3}$$

Rearranging the terms in (3), we have the implicit expression, $\hat{\boldsymbol{\beta}}_{\cdot,\lambda} = (\boldsymbol{X}'_{\cdot,\lambda}\boldsymbol{X}_{\cdot,\lambda})^{-1}(\boldsymbol{X}'_{\cdot,\lambda}\boldsymbol{y} - n\lambda\mathrm{sgn}(\hat{\boldsymbol{\beta}}_{\cdot,\lambda}))$.

The Lasso prediction error (PE) based on the construction dataset $(-s)$ and validation dataset s can be analyzed via inserting the prediction based on corresponding LSE, that is, $\tilde{\boldsymbol{y}}_{(-s),\lambda}$. We have,

$$\begin{aligned}\|\boldsymbol{y}_s - \hat{\boldsymbol{y}}_{(-s),\lambda}\|^2 &= \|\boldsymbol{y}_s - \tilde{\boldsymbol{y}}_{(-s),\lambda}\|^2 + \|\hat{\boldsymbol{y}}_{(-s),\lambda} - \tilde{\boldsymbol{y}}_{(-s),\lambda}\|^2 + (\boldsymbol{y}_s - \tilde{\boldsymbol{y}}_{(-s),\lambda})'(\hat{\boldsymbol{y}}_{(-s),\lambda} - \tilde{\boldsymbol{y}}_{(-s),\lambda}) \\ &\equiv \|\boldsymbol{y}_s - \tilde{\boldsymbol{y}}_{(-s),\lambda}\|^2 + (I) + (II).\end{aligned}$$

The expectation of (II) equals 0, and (I) can be simplified as $\lambda^2 n_c^2/n_v \boldsymbol{M}'_{s,\lambda}\boldsymbol{M}_{s,\lambda}$ via straightforward matrix operations. So to get rid of the systematic bias, EMCC is derived from subtracting (I) on both sides.

We call the CV methods using (1) and (2) *exactly modified CV* and *modified CV*, respectively. It is expected that there is a tradeoff between the accuracy and computational efficiency. These two criteria will be compared with the traditional CV and other methods in Section 4.

3.3 Data Splitting Strategy

In this subsection, we study the choices of n_c, n_v, and b. CV(n_v) with $n_v/n \to 1$, and $n_c \to \infty$ as $n \to \infty$ works well in model selection for fixed dimensional linear regression model (Shao 1993). For notational simplicity, if without extra explanation, by CV(n_v) we mean CV(n_v) with $n_v/n \to 1$, and $n_c \to \infty$ as $n \to \infty$. The simplification is also applied to other CV methods to be introduced later.

Here, to improve computational efficiency in high-dimensional settings, instead of carrying out the calculation for all different splits when $n_v > 1$ (which is of the order $\binom{p}{n_v}$), we apply Monte Carlo method to split the dataset, by randomly drawing (with or without replacement) a collection of b subsets of $\{1, \ldots, n\}$ with size n_v and selecting the model with minimum average criterion function value over all splits. The Monte Carlo CV was also considered in Picard and Cook (1984) and Shao (1993).

We would like to point out that the EMCC calculation is equivalent to the LSE of the model sequence on the solution path, which is closely related to the LARS-OLS (Efron et al. 2004). As suggested by an anonymous referee, a brief theoretical comparison is conducted below, with more emphasis on the computational issues in sequel.

In the Lasso penalized estimation problem, the essential goal is model selection. However, this is usually substituted by tuning parameter selection, with the rationale that the tuning parameters and models have one-to-one relationship given the data and algorithm. In reality, this is easily violated. For instance, in the CV procedure, different splits lead to different model sequences even under the same tuning parameter sequence. To examine the disagreement, suppose the sequence of tuning parameters of the whole dataset is $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)$. Denote $\alpha_\ell^{(j)}$ as the active set of the ℓth estimate on the path constructed by the jth subsample s_j using the same tuning parameter sequence $\boldsymbol{\lambda}$, where $\ell = 1, \ldots, N$; $j = 0, 1, \ldots, b$, and s_0 represents the whole dataset. Feng and Yu (2013) defined the *coherent rate (CR)* as a sequence representing the degree of agreement of the active sets across different splits for each tuning parameter location,

$$\mathrm{CR}(\ell) = \frac{\#\{j : \alpha_\ell^{(j)} = \alpha_\ell^{(0)}\}}{b}, \quad \text{where } \ell = 1, \ldots, N.$$

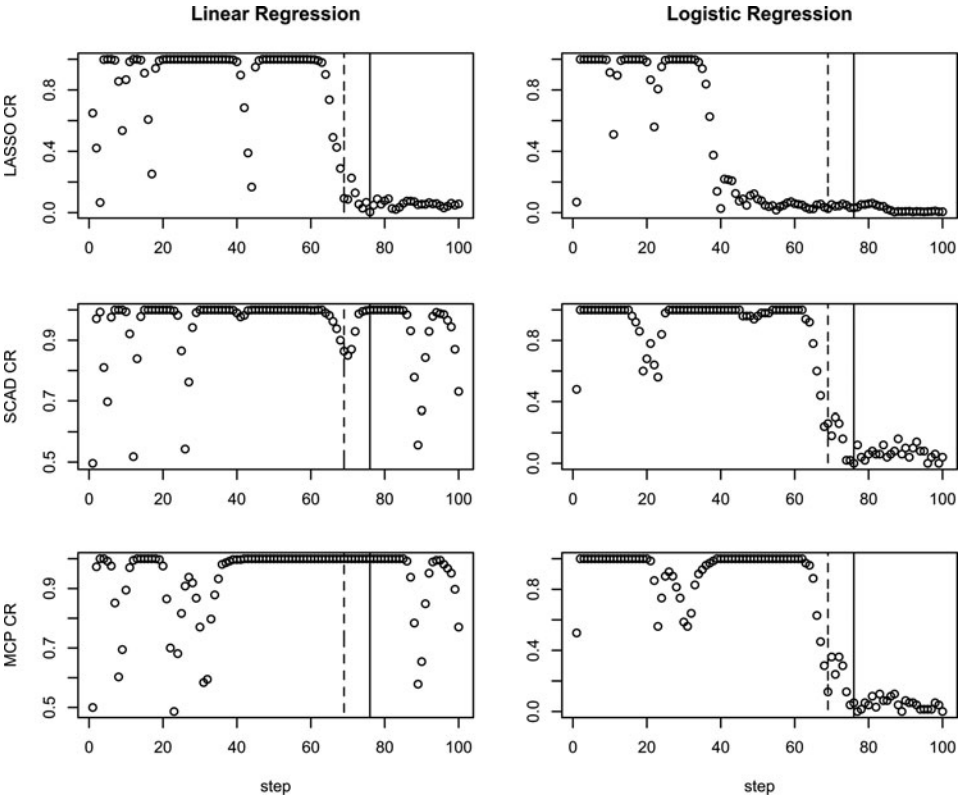

Figure 1. Coherent rate along the path for Lasso penalized linear regression estimators. The solid line "——" is where 10-fold CV chooses, and the dashed line "– –" is where noise variables start to get involved. The x-axis represents the step indices along the path, and the y-axis is the coherent rate.

In the ideal case when CR$(\cdot)$ equals 1 for all ℓ's, the conventional CV method for choosing the tuning parameter serves as an excellent surrogate for selecting the optimal model. However, this is rarely true in practice, especially when the noise variables are activated in the estimators. We demonstrate the behavior of the CR as follows.

We set $(n, p) = (500, 1000)$ and $\boldsymbol{\beta} \in \mathbb{R}^p$ with the first five coordinates (2.0, 1.6, 1.2, 0.8, 0.4) and 0 elsewhere. For $i = 1, \ldots, n$, we generate the response y_i as $y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \varepsilon_i$, where $\boldsymbol{x}_i \overset{\text{iid}}{\sim} N(\mathbf{0}_p, \boldsymbol{\Sigma})$ with $\mathbf{0}_p$ the length-p vector with 0 entries and $\varepsilon_i \overset{\text{iid}}{\sim} N(0, 1)$. $\Sigma_{j,k} = \rho^{|j-k|}$ with $\rho = 0.5$. The random splits are carried out 100 times with $n_c = 0.9n$.

It is obvious from Figure 1 that the CR is much smaller than 1 in a majority of locations, especially after the noise variables are selected. If one wants to hold the estimators (or at least the models) the same, very stringent conditions need to be imposed on the design matrix, which are usually not satisfied even for the simple simulation setting we have shown.

Considering this, we would like to compare the proposed methods with LARS-OLS regarding data splitting strategy under an ideal scenario—the induced model sequence under the same tuning parameter sequence is the same for different splits as that for the whole dataset. For this ideal setting, Shao (1993) proved that under mild conditions, CV based on

LSEs are consistent when $n_c/n \to 0$; otherwise, the CV procedure fails to be consistent. Noteworthy, this holds for our proposed algorithms, while $n_c/n = (K-1)/K > 0$ in the LARS-OLS algorithm.

3.4 EXTENSIONS

Before concluding this section, it is worthwhile to highlight the main idea of (E)MCC for Lasso penalized linear regression. Motivated by the overselection phenomenon in CV procedure caused by shrinkage, (E)MCC is developed via removing the shrinkage. Based on the linear regression on the subset of covariates, leave-n_v-out data splitting strategy is used. Back to the comparison to LARS-OLS, due to the simplicity of the Lasso penalty, we have the approximated version MCC that does not require any matrix operations when the solution path is available. By calculating the LSE for the model sequence of a given solution path, the EMCC idea can be easily extended to other popular penalties, such as SCAD (Fan and Li 2001), elastic net (Zou and Hastie 2005), MCP (Zhang 2010), among others. The algorithm can be extended to a general penalty by replacing all the solution path calculations in S1 and S2 by the ones using a general penalty, and use the following new criteria function in S3

$$\Gamma_3(\lambda) = \frac{1}{n_v}\|\boldsymbol{y}_s - \tilde{\boldsymbol{y}}_{(-s),\lambda}\|^2. \tag{4}$$

A simulation example for elastic net is available in Section 4.

4. SIMULATION

In this section, we study the performance of EMCC/MCC-based CV(n_v). In Example 1, we introduce the basic setup of the simulation with various correlation settings, including an extension to elastic net. In Example 2, we decrease the signal strength and compare different methods for the case where the position of the signals are randomly assigned. In Examples 3 and 4, performances of different n_c's and b's are reported, respectively.

Example 1 (Different Correlation Settings). We set $(n, p) = (300, 1000)$ and $\boldsymbol{\beta} \in \mathbb{R}^p$ with the first eight coordinates $(4, 3, 2, 0, 0, -4, 3, -2)$ and 0 elsewhere. For $i = 1, \ldots, n$, we generate the response y_i as follows:

$$y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \varepsilon_i,$$

where $\boldsymbol{x}_i \overset{\text{iid}}{\sim} N(\mathbf{0}_p, \boldsymbol{\Sigma})$ with $\mathbf{0}_p$ the length-p vector with 0 entries and $\varepsilon_i \overset{\text{iid}}{\sim} N(0, 1)$. The following three different correlation settings are considered.

(a) (Independent) $\Sigma_{j,k} = \mathbb{1}\{j = k\}$.

(b) (Exponential decay) $\Sigma_{j,k} = \rho^{|j-k|}$, with $\rho = 0.2, 0.5$, and 0.7.

(c) (Equal correlation) $\Sigma_{j,k} = \rho + (1-\rho)1\{j = k\}$, with $\rho = 0.2, 0.5$, and 0.7.

We repeat the simulation for 100 times.

Table 1. Comparison of the performance for different methods, for the setting of Example 1(a). Results are reported in the form of mean (standard deviation). For (e)m-MCCV(n_v), $n_c = \lceil n^{3/4} \rceil$ and $b = 50$. For m-K-fold and K-fold, $K = 10$

	Independent		
Methods	FN	FP	PE
m-MCCV(n_v)	0.00(0.00)	0.01(0.10)	0.93(0.02)
em-MCCV(n_v)	0.00(0.00)	0.00(0.00)	0.93(0.01)
m-K-fold	0.00(0.00)	38.22(16.19)	1.34(0.18)
em-K-fold	0.00(0.00)	0.38(1.45)	0.94(0.05)
K-fold	0.00(0.00)	34.99(22.06)	1.11(0.06)
AIC	0.00(0.00)	35.32(18.90)	1.11(0.06)
BIC	0.00(0.00)	4.47(3.10)	1.17(0.08)
EBIC	0.00(0.00)	1.37(1.40)	1.22(0.09)
LARS-OLS	0.00(0.00)	0.26(0.80)	0.94(0.04)
Relaxed Lasso	0.00(0.00)	0.34(1.44)	0.94(0.04)
AdaLasso	0.00(0.00)	0.00(0.00)	0.94(0.02)
Elastic Net			
em-MCCV (n_v)	0.00(0.00)	0.87(1.45)	0.95(0.05)
K-fold	0.00(0.00)	54.78(25.23)	1.21(0.08)

The results for Example 1 are reported in Table 1 for case (a) and Figure 2 for cases (b) and (c). The detailed results for cases (b) and (c) are available in Table A.1 in the Appendix. We use the `glmnet` package to generate the Lasso solution paths for the whole dataset and every subsample. For modified Monte Carlo CV(n_v) (m-MCCV(n_v)) and exactly modified Monte Carlo CV(n_v) (em-MCCV(n_v)), we set $n_c = \lceil n^{3/4} \rceil = 73$, $n_v = n - n_c = 227$, and $b = 50$, which gives robust results, with reasonable computation cost. The model selection performances of m-MCCV(n_v), em-MCCV(n_v), modified K-fold CV (m-K-fold, $K = 10$), exactly modified K-fold CV (em-K-fold, $K = 10$), and K-fold CV ($K = 10$) in `glmnet` package are presented, along with those of AIC, BIC, EBIC, LARS-OLS, relaxed Lasso, adaptive Lasso (Zou 2006), and Elastic Net. Both LARS-OLS and relaxed Lasso are computed by R package `relaxo`, adaptive Lasso solutions are obtained by `parcor`, with 10-fold CV as the default tuning parameter selection method. In Figure 2, Tables 1 and A.1, results for em-MCCV(n_v) applied to Elastic Net estimators are also included, whose paths are generated by R package `glmnet` with the default parameters, and $n_c = \lceil n^{2/3} \rceil$. To compare the performance of different methods, we report FN, FP, and PE. Here, PE is defined as the average squared PE calculated on an independent test dataset of size n.

For the independent design case, the two CV(n_v) methods have no FN and almost no FP, which indicates that they nearly achieve the model selection consistency. On the other hand, all the other cross-validation methods have a significantly large number of FPs. It is interesting to note that em-K-fold has similar behavior as LARS-OLS, as expected. But still, they have larger FP than that of em-MCCV(n_v). And the PEs of the m-MCCV and em-MCCV are smaller than those of the other methods.

The results for AIC, BIC, and EBIC are also reported. Notice that EBIC has the best performance among the three information criterion-based methods, although it is still worse than the em-MCCV. One advantage of the information criterion-based methods is

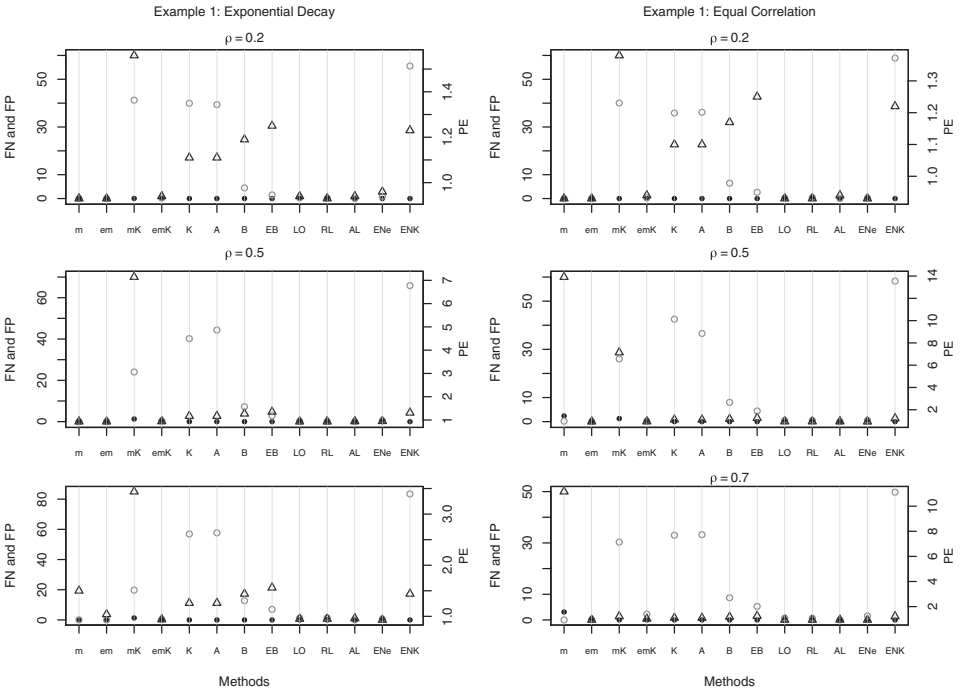

Figure 2. Comparisons of different methods and scenarios in Example 1(b) and 1(c), which are presented in the left and right columns, respectively. Along the x-axis, from left to right, the methods are m-MCCV(n_v), em-MCCV(n_v), m-K-fold, em-K-fold, K-fold, AIC, BIC, EBIC, LARS-OLS, relaxed Lasso, elastic net with em-MCCV(n_v) and elastic net with K-fold. Means of FN, FP, and PE over 100 repetitions are labeled by symbols •, △, and ○, respectively.

that they are more efficient to compute without the need of cross-validation. As suggested by one referee, we also include the comparison with adaptive Lasso, which turns out to have a comparable performance with the EMCC-based Lasso. We would like to point out that the main goal here is to find a better tuning parameter selection method for the Lasso estimator, while adaptive Lasso has different solutions from Lasso. We refer the interested readers to Zou (2006) for a detailed treatment of adaptive Lasso with the advantages over the ordinary Lasso.

For the exponential decay cases, m-MCCV and em-MCCV still perform very well with the smallest FP and FN among all the different methods, especially compared with more than 40 FPs on average in K-fold CV ($K = 10$). Due to the correlation in design matrices, compared to uncorrelated cases, the m-K-fold method misses some important variables while the em-K-fold can pick them up.

To make the case more extreme, and to see the difference between two modified criteria, we report the results for the equal correlation cases, which are not common in real applications. From the results, we can see as an approximation, the m-MCCV method is too aggressive and selects too few variables (it misses some important variables) when there is strong correlation among variables. In the presence of strong correlation, we recommend using the em-MCCV, which has superior performance.

Table 2. Computation cost comparison for the tuning parameter selection. Here, K is the number of folds, b is the number of splits in (e)m-MCCV, and L is the number of different ϕ's considered in the relaxed Lasso

(e)m-K-fold	LARS-OLS	(e)m-MCCV	Relaxed Lasso
$O(Knp\min\{n, p\})$	$O(Knp\min\{n, p\})$	$O(bnp\min\{n, p\})$	$O(KLnp\min\{n, p\})$

Now, we study the computation cost of different methods. It is well known that the cost for calculating the solution path of Lasso is $O(np\min\{n, p\})$ (Efron et al. 2004; Meinshausen 2007). The information-type methods, including AIC, BIC, and EBIC, have the best performance, since they only need one-time calculation for each model on the solution path, which leads to the computation cost $O(np\min\{n, p\})$. In Table 2, we show the computing cost comparison for other methods for the purpose of choosing the tuning parameter. We see that the (e)m-K-fold and LARS-OLS have computation cost of the same order. Since relaxed Lasso involves cross-validation on a two-dimensional parameter grid, the computation cost is $O(KLnp\min\{n, p\})$, where L is the number of different ϕ's, representing different level of penalty on the specific variable. Depending on the values of b and L, we expect (e)m-MCCV and relaxed Lasso have similar computation cost.

As mentioned in Section 3, the idea of removing the systematic bias can be easily extended to other popular penalties. As expected, em-MCCV(n_v) leads to much smaller FPs than those of K-fold CV for elastic net, and it also leads to a smaller PE.

Example 2 (Random Position with Small Signals). We use exactly the same setting as Example 1 except to reduce the signal strength by setting the $\boldsymbol{\beta}$ to have the nonzero coordinates $(1.2, 0.8, 0.4)$. In this example, we use independent design and exponential decay design with $\rho = 0.5$. In the exponential decay design case, aside from the case where the signals lie in the first three coordinates, we also consider the case when the signals are randomly positioned.

The results of Example 2 are reported in Figure A.1, which is left in the Appendix. The purpose of this example is to investigate the performances of different methods when the signals are of small strength, also to show the results when the signals are randomly positioned. In general, we have similar conclusions as Example 1. It is interesting to notice that when the signals are randomly positioned, the methods have similar behavior as independent case, which is because exponential decay correlation implies that the signal variables are approximately independent since their positions can be very different.

Example 3 (Different n_c). In this example, we would like to study the influence of different splitting rates for the construction and validation dataset (i.e., different values of n_c). We report the results for independent and exponential decay settings in Example 1 with $\rho = 0.5$.

Figure 3 summarizes the results of m-MCCV(n_v) and em-MCCV(n_v) with n_c varying from $\lceil n^{10/16} \rceil$ to $\lceil n^{15/16} \rceil$. For both methods, $n_c = \lceil n^{12/16} \rceil$ leads to the best performance,

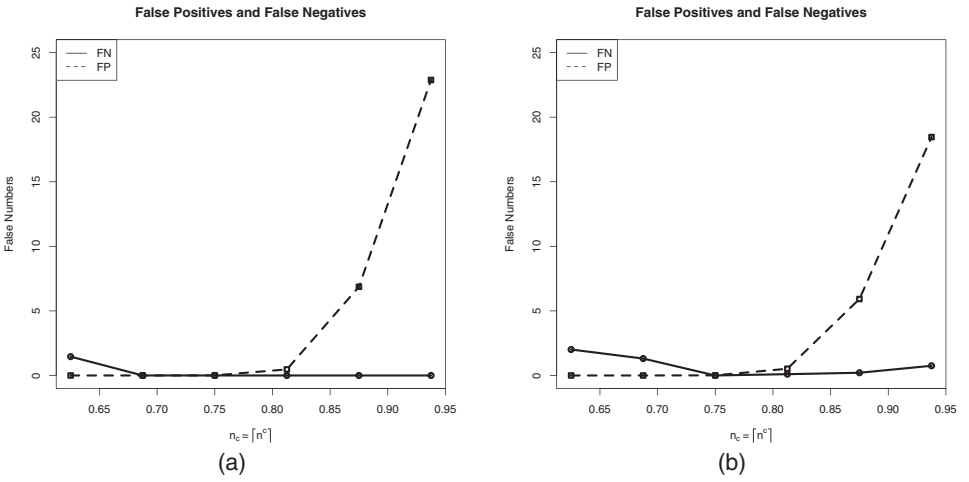

Figure 3. FN and FP of m-MCCV(n_v) for independent and exponential decay designs.

in terms of FN, FP, and PE. But if n_c is smaller than $\lceil n^{12/16} \rceil$ (e.g., when $n_c = \lceil n^{10/16} \rceil$), m-MCCV misses some important variables. The possible reason is that when n_c is too small in this fixed sample simulation example, the penalty imposed on selecting one more variable exceeds the loss of missing one important variable. For $c = 14/16$ and $c = 15/16$ cases, the approximation results in substantial number of FN, which the em-MCCV(n_v) can remedy. If computation cost is an issue, m-MCCV(n_v) is preferred as long as n_c is properly chosen and the correlation is not too extreme.

It is also interesting to see the trend in FP. When n_c increases, the FP also increases. This is consistent with our intuition, that if the validation dataset is small, detection of different models becomes more difficult. It is clear to see, along with the increasing of n_c rate, that is, the ratio of constructing sample size to the whole sample size, there will be more FPs and less FNs. This also gives us a rough guideline how to choose the proper n_c rate. From the figures we can see, if $c = 3/4$, both FPs and FNs attain a relatively low value. Larger or smaller c value will cause one side uneven.

Example 4 (Different Number of Splits b). For the setting used in Example 1, we vary b from 10 to 300, which is equivalent to the order $n^{3/8}$ to n. In Table 3, we show that results for modified-reversed-K-fold (m-r-K-fold) strategy, that is, instead of using one fold to validate and $K - 1$ folds to construct, we use $K - 1$ folds to validate and one fold to construct.

The simulation results for different b are almost the same in our experiments, which indicates that FN, FP, and PE are not sensitive to the choice of b in both correlation settings. The detailed results can be found in Table A.2. In real applications, a conservative way is to set b slightly larger if enough computational resource is available. In all the other simulations and the following real data analysis, we use $b = 50$, which exceeds $n^{1/2}$ and produces stable results.

Table 3. Reversed K-fold splitting strategy, with $K = 10$. Results are reported in the form of mean(standard deviation). (e)m-r-K-fold CV is short for (exactly) modified reversed K-fold cross-validation

	Independent		
Methods	FN	FP	PE
m-r-K-fold CV	0.00(0.00)	1.04(0.06)	1.33(0.38)
em-r-K-fold CV	0.00(0.00)	0.10(0.10)	1.06(0.02)
K-fold CV	0.00(0.00)	34.99(22.06)	1.11(0.06)
	Exponential decay ($\rho = 0.5$)		
m-r-K-fold CV	0.00(0.00)	0.98(0.04)	0.17(0.38)
em-r-K-fold CV	0.00(0.00)	0.06(0.03)	0.95(0.05)
K-fold CV	0.00(0.00)	40.21(18.18)	1.17(0.07)
	Equal correlation ($\rho = 0.5$)		
m-r-K-fold CV	0.00(0.00)	1.02(0.05)	0.98(0.02)
em-r-K-fold CV	0.21(0.40)	4.27(5.37)	0.99(0.05)
K-fold CV	0.00(0.00)	42.52(22.59)	1.11(0.06)

In Table 3, it is surprising to see, by using a small number of splits (e.g., $K = 10$), the modified-reversed-K-fold strategy can achieve very good results. This strategy guarantees each sample appears in the construction/validation set for the same number of times. It is worth to point out that although the results are very good when using the modified-reversed-K-fold, this splitting strategy does not belong to the block incomplete design or BICV (Shao 1993), since it does not balance the frequency of the pairs. In addition, the modified-reversed-K-fold takes less time to compute and has the same order of computation cost as the regular K-fold CV.

5. DATA ANALYSIS

We now illustrate one application of the proposed m-MCCV(n_v) method via the dataset reported by Scheetz et al. (2006) and analyzed by Huang, Horowitz, and Wei (2010) and Fan, Feng, and Song (2011). In this dataset, for harvesting of tissue from the eyes and subsequent microarray analysis, 120 12-week-old male rats were selected. 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. Gene expression levels were analyzed on a logarithmic scale.

Following Huang, Horowitz, and Wei (2010) and Fan, Feng, and Song (2011), we are interested in finding the genes that are related to the TRIM32 gene, which was recently found to cause Bardet–Biedl syndrome (Chiang et al. 2006) and is a genetically heterogenous 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, many of these are not expressed in the eye tissue. We only focus on the 18,975 probes that are expressed in the eye tissue.

Table 4. Comparison results of m-MCCV(n_v) and 10-fold-CV for real dataset, model size, and PE are presented in the form of mean(standard deviation), 100 repetitions are conducted

m-MCCV(n_v)		10-fold CV	
Size	PE	Size	PE
17.90(2.99)	0.01(0.01)	60.30(17.07)	0.01(0.00)

We use R package `glmnet` to compute the Lasso solution paths, and compare our proposed modified CV criterion with the 10-fold CV. The results are presented in Table 4, with $n_c = \lceil n^{3/4} \rceil$ and $b = 50$. We can achieve the same PE with only 17 variables on average, compared with 60 in 10-fold CV case. This shows that the m-MCCV(n_v) can generate more a parsimonious model while keeping the same prediction power, which could be potentially helpful in guiding the biologists to focus on the fewer selected genes.

In Figure 4, we show the histograms of the proportion of the gene being selected in 100 splits. In the stability selection theory developed in Meinshausen and Bühlmann (2010), the selection proportion of a certain variable can represent the degree of "stability" for the Lasso estimator. As a result, the variables with larger values of proportion are more likely to be "important." We reproduce the histogram in the right two subfigures of Figure 4 for genes with selected proportion larger than 0.4. It is worth noting that the histogram of proposed m-MCCV has a big gap between 0.5 and 0.7, while no similar pattern is observed for that of K-fold CV. This particular gap may serve as a natural threshold as whether the gene is important.

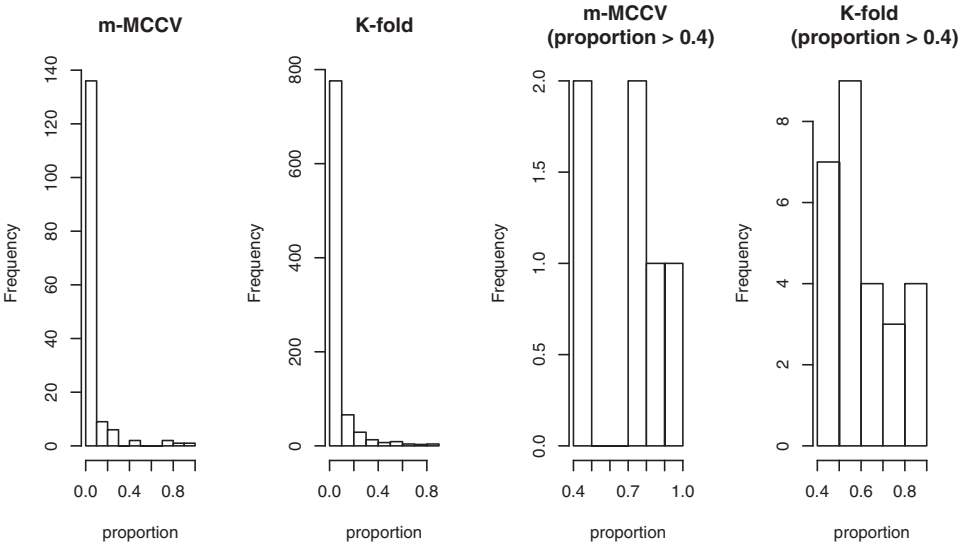

Figure 4. Histograms of gene appearance proportions using m-MCCV(n_v) and K-fold CV. The two left figures are the histograms of all the genes appeared in 100 repetitions and two right ones are genes with proportion greater than 0.4 only.

In Table A.3, we list the symbols of the genes selected by each method along with the selection proportion up to the median number of selected variables in 100 splits. Note that "—" represents there is no known symbol for the corresponding gene. In general, genes with symbols have been shown to carry certain biological functions. It is observed that 70.6% of the genes selected by m-MCCV(n_v) have gene symbols, compared with 59.3% for 10-fold CV. In addition, if we adopt 0.6 as a selection cut-point (this corresponds to the gap mentioned in (b)), all the four genes m-MCCV(n_v) selected have gene symbols, compared with only 50% for the 10-fold CV.

6. DISCUSSION

In this article, we systematically investigated the behavior of different types of CV, with different criterion functions, applied to the tuning parameter selection problem in Lasso penalized linear regression models. By removing the bias caused by the Lasso penalty, we proposed a new CV method em-MCCV with an approximated version m-MCCV. Both methods work well in simulations and real applications.

Some interesting future work includes the theoretical investigation of the inconsistency of the traditional K-fold CV. Also, we conjecture that the newly proposed em-MCCV(n_v) is model selection consistent under certain technical conditions. Other work includes extensions to generalized linear models, Cox models, and semiparametric models.

APPENDIX: TABLES AND FIGURES

We include the detailed tables and figures of Sections 4 and 5 in the Appendix.

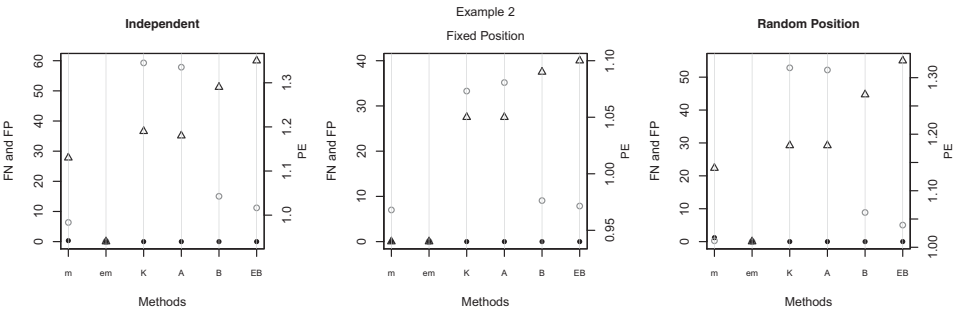

Figure A.1. Comparisons of different methods and scenarios in Example 2. Along the x-axis, the methods are m-MCCV(n_v), em-MCCV(n_v), K-fold, AIC, BIC, and EBIC, sequentially. Means of FN, FP, and PE over 100 repetitions are labeled by symbols •, △, and ○, respectively.

Table A.1. Comparison of the performance for different methods, for exponential decay cases with different ρ in Example 1(b) and (c). Results are reported in the form of mean (standard deviation). For (e)m-MCCV(n_v), $n_c = \lceil n^{3/4} \rceil$ and $b = 50$. For m-K-fold and K-fold, $K = 10$

Methods	FN	FP	PE	FN	FP	PE	FN	FP	PE
Exponential decay		$\rho = 0.2$			$\rho = 0.5$			$\rho = 0.7$	
m-MCCV(n_v)	0.00(0.00)	0.00(0.00)	0.93(0.01)	0.00(0.00)	0.01(0.10)	0.93(0.02)	0.26(0.09)	0.01(0.10)	1.50(0.17)
em-MCCV(n_v)	0.00(0.00)	0.00(0.00)	0.93(0.01)	0.00(0.00)	0.03(0.17)	0.93(0.02)	0.08(0.09)	0.00(0.00)	1.04(0.32)
m-K-fold	0.03(0.30)	41.24(10.34)	1.56(1.57)	1.28(1.73)	24.02(24.10)	7.14(8.25)	1.32(0.10)	19.73(10.20)	3.44(5.21)
em-K-fold	0.00(0.00)	0.23(0.93)	0.94(0.04)	0.00(0.00)	0.42(1.43)	0.94(0.04)	0.00(0.00)	0.25(0.70)	0.94(0.03)
K-fold	0.00(0.00)	39.93(29.03)	1.11(0.06)	0.00(0.00)	40.21(18.18)	1.17(0.07)	0.00(0.00)	56.93(5.73)	1.26(0.09)
AIC	0.00(0.00)	39.39(18.01)	1.11(0.06)	0.00(0.00)	44.36(20.54)	1.17(0.07)	0.00(0.00)	57.65(17.81)	1.26(0.09)
BIC	0.00(0.00)	4.48(3.42)	1.19(0.08)	0.00(0.00)	7.25(4.49)	1.27(0.10)	0.00(0.00)	12.76(5.65)	1.44(0.12)
EBIC	0.00(0.00)	1.52(1.53)	1.25(0.10)	0.00(0.00)	2.90(2.23)	1.35(0.11)	0.00(0.00)	7.02(3.33)	1.56(0.16)
LARS-OLS	0.00(0.00)	0.29(1.12)	0.94(0.04)	0.00(0.00)	0.23(0.71)	0.93(0.03)	0.00(0.00)	1.19(1.13)	0.95(0.03)
Relaxed Lasso	0.00(0.00)	0.16(0.72)	0.93(0.03)	0.00(0.00)	0.15(0.61)	0.93(0.03)	0.00(0.00)	1.41(1.27)	0.95(0.04)
AdaLasso	0.00(0.00)	0.00(0.00)	0.94(0.02)	0.00(0.00)	0.00(0.00)	0.94(0.02)	0.00(0.00)	0.00(0.00)	0.96(0.03)
Elastic Net									
em-MCCV(n_v)	0.00(0.00)	0.95(1.53)	0.96(0.06)	0.00(0.00)	0.53(1.03)	0.95(0.04)	0.00(0.00)	0.54(0.82)	0.93(0.02)
K-fold	0.00(0.00)	55.56(22.67)	1.23(0.08)	0.00(0.00)	65.87(21.73)	1.31(0.09)	0.00(0.00)	83.45(16.91)	1.44(0.13)
Equal correlation									
m-MCCV(n_v)	0.00(0.00)	0.00(0.00)	0.93(0.02)	2.36(1.22)	0.20(0.78)	13.91(6.54)	3.09(0.47)	0.00(0.00)	11.16(0.67)
em-MCCV(n_v)	0.00(0.00)	0.00(0.00)	0.93(0.02)	0.00(0.00)	0.06(0.34)	0.93(0.02)	0.00(0.00)	0.22(0.52)	0.93(0.02)
m-K-fold	0.00(0.00)	40.04(8.19)	1.38(0.11)	1.28(1.74)	26.03(21.32)	7.14(8.25)	0.00(0.00)	30.31(10.53)	1.22(0.20)
em-K-fold	0.00(0.00)	0.29(1.00)	0.94(0.04)	0.00(0.00)	0.30(0.79)	0.94(0.03)	0.00(0.00)	2.23(0.07)	1.03(0.02)
K-fold	0.00(0.00)	35.86(18.53)	1.10(0.06)	0.00(0.00)	42.52(22.59)	1.11(0.06)	0.00(0.00)	32.99(14.74)	1.10(0.06)
AIC	0.00(0.00)	36.18(17.37)	1.10(0.06)	0.00(0.00)	36.58(16.38)	1.10(0.06)	0.00(0.00)	33.21(12.06)	1.10(0.06)
BIC	0.00(0.00)	6.43(3.82)	1.17(0.08)	0.00(0.00)	8.01(3.95)	1.17(0.07)	0.00(0.00)	8.63(4.27)	1.18(0.07)
EBIC	0.00(0.00)	2.63(2.05)	1.25(0.11)	0.00(0.00)	4.44(3.47)	1.25(0.11)	0.00(0.00)	5.28(3.61)	1.25(0.10)
LARS-OLS	0.00(0.00)	0.21(0.76)	0.93(0.03)	0.00(0.00)	0.51(1.51)	0.94(0.05)	0.00(0.00)	0.73(1.25)	0.94(0.04)
Relaxed Lasso	0.00(0.00)	0.43(1.96)	0.93(0.04)	0.00(0.00)	0.51(1.37)	0.94(0.03)	0.00(0.00)	0.63(1.14)	0.94(0.03)
AdaLasso	0.00(0.00)	0.02(0.20)	0.94(0.02)	0.00(0.00)	0.01(0.10)	0.94(0.02)	0.00(0.00)	0.01(0.10)	0.95(0.03)
Elastic Net									
em-MCCV(n_v)	0.00(0.00)	0.20(0.60)	0.93(0.03)	0.00(0.00)	0.54(0.91)	0.93(0.02)	0.00(0.00)	1.56(1.93)	0.94(0.03)
K-fold	0.00(0.00)	58.90(23.36)	1.22(0.08)	0.00(0.00)	58.35(21.19)	1.22(0.08)	0.00(0.00)	49.76(11.63)	1.22(0.08)

Table A.2. Comparison of m-MCCV(n_v) with different b, using the settings in Example 1, with $n_c = \lceil n^{3/4} \rceil$. Results are reported in the form of mean(standard deviation)

Methods	b	FN	FP	PE
		Independent		
m-MCCV(n_v)	10	0.00(0.00)	0.01(0.10)	0.92(0.02)
	50	0.00(0.00)	0.01(0.10)	0.92(0.02)
	150	0.00(0.00)	0.01(0.10)	0.92(0.02)
	300	0.00(0.00)	0.01(0.10)	0.92(0.02)
K-fold CV		0.00(0.00)	34.99(22.06)	1.33(0.16)
		Exponential decay ($\rho = 0.5$)		
m-MCCV(n_v)	10	0.00(0.00)	0.01(0.10)	0.93(0.02)
	50	0.00(0.00)	0.01(0.10)	0.93(0.02)
	150	0.00(0.00)	0.01(0.10)	0.93(0.02)
	300	0.00(0.00)	0.01(0.10)	0.93(0.02)
K-fold CV		0.00(0.00)	40.21(18.18)	1.17(0.07)

Table A.3. Proportions of gene being selected in 100 splits for m-MCCV(n_v) and K-fold CV

m-MCCV(n_v)				10-fold CV			
Gene symbol	Prop.	Gene symbol	Prop.	Gene symbol	Prop.	Gene symbol	Prop.
TRIM41	0.95	TRIM41	0.87	CTDSPL	0.50	FRAS1	0.30
CCBL1	0.89	TNFSF13	0.87	RASL12	0.48	RGD1307201	0.30
ES1	0.80	—	0.87	GJB2	0.47	—	0.29
TRAK2	0.71	—	0.87	HERC3	0.46	RGD1566403	0.29
—	0.42	—	0.75	ASMT	0.46	WSB2	0.29
—	0.41	—	0.74	—	0.45	RGD1308031	0.28
LOC678910	0.29	—	0.71	ADRB2	0.42	—	0.28
RGD1305680	0.29	ACAT1	0.70	—	0.40	—	0.28
—	0.27	—	0.68	—	0.39	LOC296637	0.28
HDAC11	0.26	ZFP367	0.68	—	0.38	CPNE9	0.28
—	0.25	ANO10	0.62	—	0.37	—	0.27
TNFSF13	0.24	FAM118B	0.60	—	0.36	HINT1	0.27
—	0.19	RGD1561792	0.58	—	0.35	RGD1309888	0.27
HEATR6	0.16	PURB	0.57	—	0.35	—	0.26
ACAT1	0.13	—	0.56	—	0.35	BGLAP	0.25
CABP1	0.13	ACLY	0.55	—	0.34	GFAP	0.24
MARVELD1	0.13	HDAC11	0.55	PRR12	0.34	STK11	0.24
		—	0.55	YTHDF3	0.33	CYP4A3	0.24
		JAK2	0.53	KLRD1	0.32	ES1	0.24
		ATP6V1A	0.52	—	0.32		

SUPPLEMENTARY MATERIALS

R Code: The supplemental files for this article include R programs that can be used to replicate the simulation study and the real data analysis. The real data are available upon request. Please read file README contained in the zip file for more details. (MCC.zip)

ACKNOWLEDGMENTS

The authors thank the editor, the associate editor, and two anonymous referees for their constructive comments that have greatly improve the scope of the article. The financial support from NSF grant DMS-1308566 is greatly acknowledged.

[Received September 2012. Revised September 2013.]

REFERENCES

Chen, J., and Chen, Z. (2008), "Extended Bayesian Information Criteria for Model Selection With Large Model Spaces," *Biometrika*, 95, 759–771. [1010]

Chen, S., and Donoho, D. (1994), "Basis Pursuit," in *Signals, Systems and Computers, 1994. 1994 Conference Record of the Twenty-Eighth Asilomar Conference on*, Vol. 1, pp. 41–44. [1009]

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, 6287–6292. [1021]

Efron, B. (1983), "Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation, and the Repeated Learning-Testing Methods," *Biometrika*, 78, 316–331. [1010]

——— (1986), "How Biased is the Apparent Error Rate of a Prediction Rule?" *Journal of the American Statistical Association*, 81, 461–470. [1010]

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression" (with discussion), *The Annals of Statistics*, 32, 407–499. [1009,1010,1014,1019]

Fan, J., Feng, Y., and Song, R. (2011), "Nonparametric Independence Screening Nonparametric Independence Screening in Sparse Ultra-High Dimensional Additive Models," *Journal of the American Statistical Association*, 106, 544–557. [1021]

Fan, J., and Li, R. (2001), "Variable Selection Via Nonconcave Penalized Likelihood and its Oracle Properties," *Journal of the American Statistical Association*, 96, 1348–1360. [1009,1016]

Feng, Y., and Yu, Y. (2013), "Consistent Cross-Validation for Tuning Parameter Selection in High-Dimensional Variable Selection," Arxiv Preprint 1308.5390v1. [1014]

Friedman, J., Hastie, T., and Tibshirani, R. (2007), "Pathwise Coordinate Optimization," *The Annals of Statistics*, 1, 302–332. [1009]

——— (2010), "Regularization Paths for Generalized Linear Models Via Coordinate Descent," *Journal of Statistical Software*, 33, 1–22. [1010]

Fu, W. (1998), "Penalized Regressions: The Bridge Versus the Lasso," *Journal of Computational and Graphical Statistics*, 7, 397–416. [1009]

Huang, J., Horowitz, J. L., and Wei, F. (2010), "Variable Selection in Nonparametric Additive Models," *The Annals of Statistics*, 38, 2282–2313. [1021]

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, 249–264. [1021]

Luo, X., Stefanski, L. A., and Boos, D. D. (2006), "Tuning Variable Selection Procedures by Adding Noise," *Technometrics*, 48, 165–175. [1010]

Meinshausen, N. (2007), "Relaxed Lasso," *Computational Statistics and Data Analysis*, 52, 374–393. [1010,1019]

Meinshausen, N., and Bühlmann, P. (2010), "Stability Selection," *Journal of the Royal Statistical Society,* Series B, 72, 417–473. [1022]

Nishii, R. (1984), "Asymptotic Properties of Criteria for Selection of Variables Asymptotic Properties of Criteria for Selection of Variables in Multiple Regression," *The Annals of Statistics*, 12, 758–765. [1010]

Osborne, M., Presnell, B., and Turlach, B. (2000), "A New Approach to Variable Selection in Least Squares Problems," *IMA Journal of Numerical Analysis*, 20, 389–404. [1009]

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, 69, 659–677. [1010]

Picard, R. R., and Cook, R. D. (1984), "Cross-Validation of Regression Models," *Journal of the American Statistical Association*, 79, 575–583. [1014]

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, 14,429–14,434. [1021]

Shao, J. (1993), "Linear Model Selection by Cross-Validation," *Journal of the American Statistical Association*, 88, 486–494. [1010,1011,1014,1015,1021]

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, 39, 44–47. [1010]

Tibshirani, R. (1996), "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society,* Series B, 9, 1135–1151. [1009,1011]

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, 71, 671–683. [1010]

Wu, Y., Boos, D. D., and Stefanski, L. A. (2007), "Controlling Variable Selection by the Addition of Pseudo Variables," *Journal of the American Statistical Association*, 102, 235–243. [1010]

Zhang, C. H. (2010), "Nearly Unbiased Variable Selection Under Minimax Concave Penalty," *The Annals of Statistics*, 38, 894–942. [1009,1013,1016]

Zhang, C. H., and Huang, J. (2008), "The Sparsity and Bias of the Lasso Selection in High-Dimensional Regression," *The Annals of Statistics*, 36, 1567–1594. [1010]

Zhang, P. (1993), "Model Selection Via Multifold Cross Validation," *The Annals of Statistics*, 21, 299–313. [1010]

Zhang, Y., Li, R., and Tsai, C. L. (2010), "Regularization Parameter Selections Via Generalized Information Criterion," *Journal of American Statistical Association*, 105, 312–323. [1010]

Zou, H. (2006), "The Adaptive Lasso and its Oracle Properties," *Journal of the American Statistical Association*, 101, 1418–1429. [1017,1018]

Zou, H., and Hastie, T. (2005), "Regularization and Variable Selection Via the Elastic Net," *Journal of the Royal Statistical Society,* Series B, 67, 301–320. [1009,1016]

Zou, H., Hastie, T., and Tibshirani, R. (2007), "On the "Degrees of Freedom" of the Lasso," *The Annals of Statistics*, 35, 2173–2192. [1010]

Zou, H., and Li, R. (2008), "One-Step Sparse Estimates in Nonconcave Penalized Likelihood Models," *The Annals of Statistics*, 38, 1509–1533. [1009]