# 論文 / 著書情報
# Article / Book Information

| | |
|---|---|
| Title | Semi-Analytic Resampling in Lasso |
| Authors | Tomoyuki Obuchi, Yoshiyuki Kabashima |
| Citation | Journal of Machine Learning Research, Vol. 20, No. 70, pp. 1-33 |
| Pub. date | 2019, 4 |
| URL | http://jmlr.org/papers/v20/18-109.html |

Journal of Machine Learning Research 20 (2019) 1-33 Submitted 2/18; Revised 4/19; Published 4/19

# Semi-Analytic Resampling in Lasso

**Tomoyuki Obuchi** OBUCHI@C.TITECH.AC.JP

**Yoshiyuki Kabashima** KABA@C.TITECH.AC.JP

*Department of Mathematical and Computing Science*
*Tokyo Institute of Technology*
*2-12-1, Ookayama, Meguro-ku, Tokyo, Japan*

**Editor:** Jon McAuliffe

## Abstract

An approximate method for conducting resampling in Lasso, the $\ell_1$ penalized linear regression, in a semi-analytic manner is developed, whereby the average over the resampled datasets is directly computed without repeated numerical sampling, thus enabling an inference free of the statistical fluctuations due to sampling finiteness, as well as a significant reduction of computational time. The proposed method is based on a message passing type algorithm, and its fast convergence is guaranteed by the state evolution analysis, when covariates are provided as zero-mean independently and identically distributed Gaussian random variables. It is employed to implement bootstrapped Lasso (Bolasso) and stability selection, both of which are variable selection methods using resampling in conjunction with Lasso, and resolves their disadvantage regarding computational cost. To examine approximation accuracy and efficiency, numerical experiments were carried out using simulated datasets. Moreover, an application to a real-world dataset, the wine quality dataset, is presented. To process such real-world datasets, an objective criterion for determining the relevance of selected variables is also introduced by the addition of noise variables and resampling. MATLAB codes implementing the proposed method are distributed in (Obuchi, 2018).

**Keywords:** bootstrap method, Lasso, variable selection, message passing algorithm, replica method

## 1. Introduction

Variable selection is an important problem in statistics, signal processing, and machine learning. A desire for useful techniques of variable selection has recently been growing, as cumulated advances in measurement and information technologies have started to steadily produce a large amount of high-dimensional data in science and engineering. A naive method for selecting relevant variables requires solving discrete optimization problems. This involves a serious computational difficulty as the dimensionality of the variables increases, even in the simplest case of linear models (Natarajan, 1995). Hence, certain relaxations or approximations are required for handling such large high-dimensional datasets.

A great deal of progress has been made with regard to relaxation techniques by introducing an $\ell_1$ penalty (Tibshirani, 1996; Meinshausen and Bühlmann, 2004; Banerjee et al., 2006; Friedman et al., 2008). Its model consistency, that is, whether the estimated model

converges to the true model in the large size limit of data, has been extensively studied, particularly in the case of linear models (Knight and Fu, 2000; Zhao and Yu, 2006; Yuan and Lin, 2007; Wainwright, 2009; Meinshausen and Yu, 2009). These studies show that a naive usage of the $\ell_1$ penalty affects model consistency in realistic settings: The resultant estimator cannot completely reject variables not present in the true model if there exist non-trivial correlations between covariates, although the normal consistency in the $\ell_2$ sense is retained. This fact motivated the development of further techniques for recovering model consistency in the variable selection context, particularly in the last decade, including traditional approaches using confidence interval and hypothesis testing (Zou, 2006; Bach, 2008; Meinshausen and Bühlmann, 2010; Javanmard and Montanari, 2014a,b, 2015; Lockhart et al., 2014; G'Sell et al., 2013; Takahashi and Kabashima, 2018). If the distribution of the estimator is accurately figured out and the corresponding P-value is efficiently computed, those traditional approaches are quite efficient and powerful enough to correctly perform variable selection. However in many practical situations, the functional form of the distribution is not clear. In such cases, certain numerical approaches are employed to approximately estimate the distribution. Resampling is one of the representative methods and is focused on in the present paper.

Resampling is a versatile idea applicable to a wide range of problems in statistical modeling and is broadly used in machine learning algorithms; some examples can be found in boosting and bagging (Trevor et al., 2009). In statistics, Efron's bootstrap method is a pioneering example in which resampling is efficiently and systematically used (Efron and Tibshirani, 1994). In the case of the $\ell_1$-penalized linear regression or Lasso (Tibshirani, 1996), through a fine evaluation of each variable's probability to be selected (positive probability) in a fairly general setting, Bach showed that it is possible to perform statistically consistent variable selection by utilizing the bootstrap method (Bach, 2008); the associated algorithm is called *bootstrapped Lasso* (Bolasso). Another algorithm based on a similar idea, called *stability selection* (SS), was also implemented by randomizing the penalty coefficient in Lasso (Meinshausen and Bühlmann, 2010). All these examples demonstrate that resampling is powerful and versatile.

A common disadvantage of such resampling approaches is their computational cost. For example, in the Bolasso case, the $\ell_1$-penalized linear regression should be recursively solved according to the number of resampled datasets, which should be sufficiently large so that the positive probability of variables may be estimated. This multiple computational cost precludes the application of such resampling techniques to large datasets. The aim of this study is to avoid this problem by developing an approximation whereby the resampling is conducted in a semi-analytic manner, and to implement it in Lasso.

Such an approximation was already obtained by Malzahn and Opper (2003), where a general framework of semi-analytic resampling was developed using the replica method from statistical mechanics, and was demonstrated in Gaussian process regression. We follow their idea and pursue how it works in Lasso with certain resampling manners.

The remaining of the paper is organized as follows. In the next section, the general framework for semi-analytic resampling using the replica method is reviewed. Concrete formulas in the Lasso case are also provided. After taking the average with respect to the resampling, there remains an intractable statistical model. Handling this model is another key issue, and certain approximations (expected to be exact under certain conditions) should

be adopted. In this study, Gaussian approximation is used in conjunction with the so-called cavity method, providing a message passing type algorithm. Its dynamical behavior is analyzed by the so-called state evolution, which guarantees that the algorithm convergence is free from the model dimensionality and the dataset size when covariates are provided as zero-mean independently and identically distributed (i.i.d.) Gaussian random variables. They are explained in Section 3. In Section 4, numerical experiments are carried out on both simulated and real-world datasets to examine the accuracy and efficiency of the proposed semi-analytic method. For processing real-world datasets, an objective criterion for determining the relevance of selected variables is also proposed, based on the addition of noise variables and resampling. The last section concludes the paper.

## 2. Formulation for semi-analytic resampling

Let us start from preparing notations and definitions. We denote by $D$ a given dataset, which usually consists of a set of inputs $\boldsymbol{x}_\mu$ and outputs $y_\mu$ as $D = \{(\boldsymbol{x}_\mu, y_\mu)\}_{\mu=1}^M$, and denote our basic statistics by $\hat{\boldsymbol{\beta}}(D) = (\hat{\beta}_1(D), \cdots, \hat{\beta}_N(D))^\top$. They are interpreted as an estimator of true parameters in the true model that is inferred. Based on the Bayesian inference framework, it is assumed that the basic estimator can be expressed as an average over a certain posterior distribution $P(\boldsymbol{\beta}|\boldsymbol{\lambda}, D)$, that is,

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda}, D) = \int d\boldsymbol{\beta}\, \boldsymbol{\beta} P(\boldsymbol{\beta}|\boldsymbol{\lambda}, D) \equiv \langle \boldsymbol{\beta} \rangle \,, \tag{1}$$

where $\langle \cdot \rangle$ represents the average over the posterior distribution which is supposed to be constructed as follows:

$$P(\boldsymbol{\beta}|\boldsymbol{\lambda}, D) = \frac{1}{Z(\boldsymbol{\lambda}, D)} P_0(\boldsymbol{\beta}|\boldsymbol{\lambda}) P(D|\boldsymbol{\beta}). \tag{2}$$

Here, $P_0(\boldsymbol{\beta}|\boldsymbol{\lambda})$ is the prior distribution characterized by the parameters $\boldsymbol{\lambda}$, and $P(D|\boldsymbol{\beta})$ represents the likelihood. The normalization constant or the partition function can be explicitly expressed by

$$Z(\boldsymbol{\lambda}, D) = \int d\boldsymbol{\beta}\, P_0(\boldsymbol{\beta}|\boldsymbol{\lambda}) P(D|\boldsymbol{\beta}). \tag{3}$$

Considering the construction of a subsample by resampling from the full data $D$ of size $M$ with replacement, let an indicator vector $\boldsymbol{c} = (c_1, \cdots, c_M)^\top$ specify any such subsample; each $c_\mu$ is a non-negative integer counting the number of occurrences of the $\mu$-th datapoint of $D$ in the subsample. A subsample identified by $\boldsymbol{c}$ is denoted by $D_{\boldsymbol{c}}$. Specifying a resampling defines the distribution $P(\boldsymbol{c})$ of $\boldsymbol{c}$, and the question is the behavior of the basic estimators and their functions with respect to the average over $P(\boldsymbol{c})$. In some resampling techniques, additional randomization is introduced in the prior distribution (Meinshausen and Bühlmann, 2010). Hence, an average over the parameters $\boldsymbol{\lambda}$ is further considered, and the distribution $P(\boldsymbol{\lambda})$ of $\boldsymbol{\lambda}$ is introduced. For notational simplicity, the average over these distributions is denoted by square brackets with appropriate subscripts as follows:

$$[(\cdots)]_{\boldsymbol{c},\boldsymbol{\lambda}} \equiv \sum_{\boldsymbol{c}} \int d\boldsymbol{\lambda} (\cdots) P(\boldsymbol{c}) P(\boldsymbol{\lambda}).$$

The average of this type is called *configurational average* throughout this paper.

### 2.1. General theory of semi-analytic resampling

The purpose of resampling is to obtain the distribution of the basic estimator

$$P(\beta_i) = \left[\delta\left(\beta_i - \hat{\beta}_i(\boldsymbol{\lambda}, D_c)\right)\right]_{\boldsymbol{c},\boldsymbol{\lambda}}, \tag{4}$$

which is reduced to computing the moments $\left[\hat{\beta}_i^r(\boldsymbol{\lambda}, D_{\boldsymbol{c}})\right]_{\boldsymbol{c},\boldsymbol{\lambda}}$ of arbitrary degree $\forall r \in \mathbb{N}$. We thus show that these moments can be evaluated in the following manner.

By definition, the moment $\left[\hat{\beta}_i^r(\boldsymbol{\lambda}, D_{\boldsymbol{c}})\right]_{\boldsymbol{c},\boldsymbol{\lambda}}$ with $r \in \mathbb{N}$ can be written as

$$\left[\hat{\beta}_i^r(\boldsymbol{\lambda}, D_{\boldsymbol{c}})\right]_{\boldsymbol{c},\boldsymbol{\lambda}} = \left[\frac{1}{Z^r(\boldsymbol{\lambda}, D_{\boldsymbol{c}})}\int\left\{\prod_{a=1}^{r} d\boldsymbol{\beta}^a\, \beta_i^a P_0(\boldsymbol{\beta}^a|\boldsymbol{\lambda})P(D_{\boldsymbol{c}}|\boldsymbol{\beta}^a)\right\}\right]_{\boldsymbol{c},\boldsymbol{\lambda}} \tag{5}$$

$$= \lim_{n\to 0}\left[Z^{n-r}(\boldsymbol{\lambda}, D_{\boldsymbol{c}})\int\left\{\prod_{a=1}^{r} d\boldsymbol{\beta}^a\, \beta_i^a P_0(\boldsymbol{\beta}^a|\boldsymbol{\lambda})P(D_{\boldsymbol{c}}|\boldsymbol{\beta}^a)\right\}\right]_{\boldsymbol{c},\boldsymbol{\lambda}} \tag{6}$$

$$\doteq \lim_{n\to 0}\left[\int\left\{\prod_{b=1}^{n} d\boldsymbol{\beta}^b\right\}\left\{\prod_{a=1}^{r}\beta_i^a\right\}\left\{\prod_{b=1}^{n} P_0(\boldsymbol{\beta}^b|\boldsymbol{\lambda})P(D_{\boldsymbol{c}}|\boldsymbol{\beta}^b)\right\}\right]_{\boldsymbol{c},\boldsymbol{\lambda}}. \tag{7}$$

The evaluation of (5) is technically difficult owing to the existence of $Z^r(\boldsymbol{\lambda}, D_{\boldsymbol{c}})$ in the denominator: The negative power of the partition function does not allow the analytical evaluation of the average over $\boldsymbol{c}$ and $\boldsymbol{\lambda}$, even if $P(\boldsymbol{c})$ and $P(\boldsymbol{\lambda})$ take reasonable forms. To overcome this difficulty, an auxiliary parameter $n$ is introduced in (6). The exponent $n$ is assumed to be a positive integer larger than $r$ in (7); thus, the power of the partition function can be expanded in the integral form (3). Integral variables $\{\boldsymbol{\beta}^1, \boldsymbol{\beta}^2, \ldots, \boldsymbol{\beta}^n\}$ are termed *replicas* since they are regarded as constituting $n$ copies of the original system. In (7), the configurational average can be analytically computed under appropriate approximations. This yields an expression as a function of $n$ that is analytically continuable from $\mathbb{N}$ to $\mathbb{R}$. The $n \to 0$ limit is taken by employing the analytically continued expression after all computations. The evaluation technique based on these procedures is often called the *replica method* (Mézard et al., 1987; Nishimori, 2001; Dotsenko, 2005). Evidently, this technique is not justified in a strict sense, but it is known that the replica method gives correct results for many problems. Justification of the replica method is known to be difficult (Talagrand, 2003), and here we leave it as a future work and just employ the method for our purpose. In the present problem, it is far from trivial to obtain an analytically continuable expression, and below we concentrate on its derivation.

Accordingly, the configurational average of any power of the basic estimator can be evaluated from the following quantities:

$$\Xi\left(\{\boldsymbol{\beta}^a\}_{a=1}^n \mid D\right) \equiv \int\left\{\prod_{a=1}^{n} d\boldsymbol{\beta}^a\right\}\left[\prod_{a=1}^{n} P_0(\boldsymbol{\beta}^a|\boldsymbol{\lambda})P(D_{\boldsymbol{c}}|\boldsymbol{\beta}^a)\right]_{\boldsymbol{c},\boldsymbol{\lambda}},$$

$$P\left(\{\boldsymbol{\beta}^a\}_{a=1}^n \mid D\right) \equiv \frac{1}{\Xi\left(\{\boldsymbol{\beta}^a\}_{a=1}^n \mid D\right)}\left[\prod_{a=1}^{n} P_0(\boldsymbol{\beta}^a|\boldsymbol{\lambda})P(D_{\boldsymbol{c}}|\boldsymbol{\beta}^a)\right]_{\boldsymbol{c},\boldsymbol{\lambda}},$$

where the former is called replicated partition function and the latter is called replicated Boltzmann distribution. The average over the replicated Boltzmann distribution is denoted by

$$\overline{(\cdots)} \equiv \int \left\{ \prod_{a=1}^{n} d\boldsymbol{\beta}^a \right\} (\cdots)\, P\left(\{\boldsymbol{\beta}^a\}_{a=1}^n \mid D\right). \tag{8}$$

It follows from (5)–(7) that this average converges to the desired total average over the posterior and the configurational variables $\boldsymbol{c}$ and $\boldsymbol{\lambda}$ as $n \to 0$. The next step is to compute this average. The nontrivial technical difficulties are in the integrations with respect to replicas $\{\boldsymbol{\beta}^a\}_{a=1}^n$ and in deriving a functional form that is analytically continuable with respect to $n$. These will be handled after considering the specific case of Lasso.

### 2.2. Specifics to Lasso with independent sampling

The above general theory is now rephrased in the context of Lasso. Given a set of covariates $\boldsymbol{x}_\mu \in \mathbb{R}^N$ and responses $y_\mu \in \mathbb{R}$, $D = \{(\boldsymbol{x}_\mu, y_\mu)\}_{\mu=1}^M$, the usual estimator by Lasso is

$$\hat{\boldsymbol{\beta}}(\lambda, D) = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \sum_{\mu=1}^{M} \left( y_\mu - \sum_i x_{\mu i} \beta_i \right)^2 + \lambda ||\boldsymbol{\beta}||_1 \right\}.$$

In contrast to this, considering Bolasso and SS, given the resampled data $D_{\boldsymbol{c}}$ and the randomized penalty coefficients $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_N)^\top$, the following estimator is introduced:

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda}, D_{\boldsymbol{c}}) = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \sum_{\mu=1}^{M} c_\mu \left( y_\mu - \sum_i x_{\mu i} \beta_i \right)^2 + \sum_{i=1}^{N} \lambda_i |\beta_i| \right\}. \tag{9}$$

For representing this in terms of the posterior average, the following quantities are introduced:

$$\begin{aligned}
\mathcal{H}(\boldsymbol{\beta}|\boldsymbol{\lambda}, D_{\boldsymbol{c}}) &= \frac{1}{2} \sum_{\mu=1}^{M} c_\mu \left( y_\mu - \sum_i x_{\mu i} \beta_i \right)^2 + \sum_{i=1}^{N} \lambda_i |\beta_i|, \\
Z_\gamma(\boldsymbol{\lambda}, D_{\boldsymbol{c}}) &= \int d\boldsymbol{\beta}\, e^{-\gamma \mathcal{H}(\boldsymbol{\beta}|\boldsymbol{\lambda}, D_{\boldsymbol{c}})}, \\
P_\gamma(\boldsymbol{\beta}|\boldsymbol{\lambda}, D_{\boldsymbol{c}}) &= \frac{1}{Z_\gamma} e^{-\gamma \mathcal{H}(\boldsymbol{\beta}|\boldsymbol{\lambda}, D_{\boldsymbol{c}})},
\end{aligned}$$

where these quantities are called (in the order they appear) Hamiltonian, partition function, and Boltzmann distribution, in accordance with physics terminology. As $\gamma \to \infty$, the Boltzmann distribution converges to a pointwise measure at $\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda}, D_{\boldsymbol{c}})$ and thereby becomes the desired posterior distribution, allowing the identification of the Boltzmann distribution with the posterior distribution; thus, the average over the Boltzmann distribution is thereafter denoted by $\langle \cdots \rangle$ introduced in (1)[1]. The prior distribution $P_0(\boldsymbol{\beta}|\boldsymbol{\lambda})$ and the likelihood

1. It is assumed that the order of the integration and the $\gamma \to \infty$ limit may be changed.

$P(D|\boldsymbol{\beta})$ in (2) correspond to the factors $e^{-\gamma\sum_i \lambda_i|\beta_i|}$ and $e^{-\frac{\gamma}{2}\sum_{\mu=1}^{M} c_\mu\left(y_\mu-\sum_i x_{\mu i}\beta_i\right)^2}$ in the Boltzmann distribution, respectively.

Considering Bolasso and SS, we draw the subsample $\boldsymbol{c}$ of a fixed size $m$ in an unbiased manner. Each sample comes out with a probability of $M^{-m}m!/(c_1!\cdots c_M!)$, which has a weak dependency among $\{c_\mu\}_\mu$. However, this dependency is not essential for large $M$ and $m$; thus, it is ignored. Using Stirling's formula $m!\approx(\frac{m}{e})^m$ in conjunction with the relation $m=\sum_{\mu=1}^{M} c_\mu$, $c_\mu$ is approximately regarded as an i.i.d. variable from a Poisson distribution of mean $\tau=m/M$, namely,

$$P(\boldsymbol{c})=\prod_{\mu=1}^{M}\frac{\tau^{c_\mu}}{c_\mu!}e^{-\tau}=\prod_{\mu=1}^{M}P(c_\mu).$$

It is also natural to require that $P(\boldsymbol{\lambda})$ is factorized into a batch of identical distributions as follows:

$$P(\boldsymbol{\lambda})=\prod_{i=1}^{N}P(\lambda_i).$$

At this point, the explicit form of $P(\lambda_i)$ is not specified. These factorized natures allow us to express the replicated Boltzmann distribution as

$$P_\gamma(\{\boldsymbol{\beta}^a\}_{a=1}^n|D)\propto\left[e^{-\gamma\sum_{a=1}^{n}\mathcal{H}(\boldsymbol{\beta}^a|\boldsymbol{\lambda},D_{\boldsymbol{c}})}\right]_{\boldsymbol{c},\boldsymbol{\lambda}}=\prod_{\mu=1}^{M}\Phi_\mu\left(\{\boldsymbol{\beta}_i\}_{i=1}^N\right)\prod_{i=1}^{N}\Psi\left(\boldsymbol{\beta}_i\right), \tag{10}$$

where

$$\Phi_\mu\left(\{\boldsymbol{\beta}_i\}_{i=1}^N\right)=\left[e^{-\frac{1}{2}\gamma c\sum_a(y_\mu-\sum_i x_{\mu i}\beta_i^a)^2}\right]_c,$$
$$\Psi\left(\boldsymbol{\beta}_i\right)=\left[e^{-\gamma\lambda\sum_{a=1}^{n}|\beta_i^a|}\right]_\lambda,$$

and the vector notation $\boldsymbol{\beta}_i=(\beta_i^a)_a$ was introduced for later convenience. $\Phi_\mu$ is hereafter called $\mu$-th potential function.

To proceed further, an approximation should be introduced to make the right-hand side of (10) tractable. Although there are several ways for that, we here make an approximation based on the cavity method from statistical physics. The details are in the next section.

## 3. Handling the replicated system

We below work with the cavity method, or the belief propagation (BP) in computer science, and use a Gaussian approximation. This treatment is essentially identical to that used in deriving the known approximate message passing (AMP) (Kabashima, 2003; Donoho et al., 2009), and can be justified if each covariate is i.i.d. from Gaussian distributions (Bayati and Montanari, 2011; Barbier et al., 2017). For treating nontrivial correlations between covariates, more sophisticated approximations (Opper and Winther, 2001a,b, 2005; Kabashima and Vehkapera, 2014; Çakmak et al., 2014; Cespedes et al., 2014; Rangan et al., 2016; Ma and Ping, 2017; Takeuchi, 2017) are required. Such extensions are left as future work.

(10) implies that the replicated Boltzmann distribution naturally has a factor graph structure. Hence, by the cavity method, (10) can be "approximately" decomposed into two messages as follows:

$$\widetilde{\phi}_{\mu\to i}(\boldsymbol{\beta}_i) = \frac{1}{Z_{\mu\to i}} \int \prod_{j(\neq i)} d\boldsymbol{\beta}_j \ \Phi_\mu\left(\{\boldsymbol{\beta}_i\}_{i=1}^N\right) \prod_{j(\neq i)} \phi_{j\to\mu}(\boldsymbol{\beta}_j), \tag{11}$$

$$\phi_{i\to\mu}(\boldsymbol{\beta}_i) = \frac{1}{Z_{i\to\mu}} \Psi\left(\boldsymbol{\beta}_i\right) \prod_{\nu(\neq\mu)} \widetilde{\phi}_{\nu\to i}(\boldsymbol{\beta}_i), \tag{12}$$

where $Z_{\mu\to i}, Z_{i\to\mu}$ are normalization factors that are not relevant and will be discarded below. The average of (8) can be computed by employing this set of equations.

### 3.1. Gaussian approximation on cavity method

A crucial observation for assessing (11),(12) is that the residual $R_\mu^a = y_\mu - \sum_j x_{\mu j}\beta_j^a$ appearing in $\Phi_\mu$ has a sum of a large number of random variables; thus, the central limit theorem justifies treating it as a Gaussian variable with the appropriate mean and variance. This consideration leads to the following decomposition in (11)

$$R_\mu^a + x_{\mu i}\beta_i^a \equiv y_\mu - \sum_{j(\neq i)} x_{\mu j}\beta_j^a \approx y_\mu - \sum_{j(\neq i)} x_{\mu j}\overline{\beta}_j^{\setminus\mu} + Z_{\mu i}^a \equiv r_{\mu i} + Z_{\mu i}^a,$$

where $Z_{\mu i}^a$ is a zero-mean Gaussian variable whose covariance is equivalent to that of $\sum_{j(\neq i)} x_{\mu j}\beta_j^a$, and $\overline{(\cdots)}^{\setminus\mu}$ is the average over $\prod_j \phi_{j\to\mu}(\boldsymbol{\beta}_j)$ or the replicated Boltzmann distribution without $\mu$-th potential function. This shows that it is difficult to consider the correlation between $\beta_i$ and $\beta_j$ with different $i, j$ in the present framework. This could be overcome even in the cavity method framework (Opper and Winther, 2001a,b); however, it is beyond the scope of this study.

The so-called replica symmetry is now assumed in the messages. This symmetry implies that any permutation of the replicas $\{\beta_i^1, \cdots, \beta_i^n\}$ yields an identical message, which in turn implies, by De Finetti's theorem (Hewitt and Savage, 1955), that the messages can be expressed as

$$\phi_{i\to\mu}(\boldsymbol{\beta}_i) = \int d\mathcal{F}\rho_{i\to\mu}(\mathcal{F}) \prod_{a=1}^n \mathcal{F}(\beta_i^a),$$

$$\widetilde{\phi}_{\mu\to i}(\boldsymbol{\beta}_i) = \int d\mathcal{G}\rho_{\mu\to i}(\mathcal{G}) \prod_{a=1}^n \mathcal{G}(\beta_i^a),$$

where $d\mathcal{F}\rho_{i\to\mu}(\mathcal{F})$ and $d\mathcal{G}\rho_{i\to\mu}(\mathcal{G})$ are probability measures over the probability distribution functions $\mathcal{F}$ and $\mathcal{G}$, respectively. From this form, two different variances naturally emerge:

$$V_i^{\setminus\mu} \equiv \overline{(\beta_i^a)^2 - \beta_i^a\beta_i^b}^{\setminus\mu} = \int d\boldsymbol{\beta}_i \left((\beta_i^a)^2 - \beta_i^a\beta_i^b\right) \int d\mathcal{F}\rho_{i\to\mu}(\mathcal{F}) \prod_{c=1}^{n} \mathcal{F}(\beta_i^c)$$
$$= \int d\mathcal{F}\rho_{i\to\mu}(\mathcal{F}) \left\{ \int \mathcal{F}(\beta)\beta^2 d\beta - \left( \int \mathcal{F}(\beta)\beta d\beta \right)^2 \right\},$$
$$W_i^{\setminus\mu} \equiv \overline{\beta_i^a\beta_i^b}^{\setminus\mu} - \overline{\beta_i^a}^{\setminus\mu}\overline{\beta_i^b}^{\setminus\mu}$$
$$= \int d\mathcal{F}\rho_{i\to\mu}(\mathcal{F}) \left( \int \mathcal{F}(\beta)\beta d\beta \right)^2 - \left( \int d\mathcal{F}\rho_{i\to\mu}(\mathcal{F}) \int \mathcal{F}(\beta)\beta d\beta \right)^2,$$

where it was assumed that $a \neq b$. Below, the $\mu$-dependence of $V_i^{\setminus\mu}$ and $W_i^{\setminus\mu}$ is ignored, as it is small, and let

$$V_i^{\setminus\mu} \approx V_i = \overline{(\beta_i^a)^2 - \beta_i^a\beta_i^b},$$
$$W_i^{\setminus\mu} \approx W_i = \overline{\beta_i^a\beta_i^b} - \overline{\beta_i^a}\,\overline{\beta_i^b}.$$

It can be implied that $V_i$ describes the average of the variance inside a fixed resampled dataset, whereas $W_i$ represents the inter-sample variance. Using $V_i$ and $W_i$, the covariance of $\boldsymbol{\beta}_i$ is generally written as

$$\mathrm{Cov}^{\setminus\mu}\left(\beta_i^a, \beta_i^b\right) \equiv \overline{\beta_i^a\beta_i^b}^{\setminus\mu} - \overline{\beta_i^a}^{\setminus\mu}\overline{\beta_i^b}^{\setminus\mu} = W_i^{\setminus\mu} + V_i^{\setminus\mu}\delta_{ab} \approx W_i + V_i\delta_{ab}.$$

Using this relation, the covariance of $\{Z_{\mu i}^a\}_a$ can be expressed as

$$\mathrm{Cov}^{\setminus\mu}\left(Z_{\mu i}^a, Z_{\mu i}^b\right) = \sum_{j,k(\neq i)} x_{\mu i}x_{\mu j}\mathrm{Cov}^{\setminus\mu}\left(\beta_j^a, \beta_k^b\right) \approx \sum_j x_{\mu j}^2 \mathrm{Cov}^{\setminus\mu}\left(\beta_j^a, \beta_j^b\right)$$
$$\approx \sum_j x_{\mu j}^2 (W_j + V_j\delta_{ab}) \equiv W_\mu + V_\mu\delta_{ab}.$$

The correlation between $\beta_j$ and $\beta_k$ for $j \neq k$ is neglected in the second sum because it is not taken into account in the present framework, as explained above. Moreover, the addition of the $i$-th term in the same sum does not affect the following discussion because it is sufficiently small in the summation. Based on these observations, $Z_{\mu i}^a$ is decomposed as follows:

$$Z_{\mu i}^a = D_{\mu i} + \Delta_{\mu i}^a,$$

where $D_{\mu i}$ and $\Delta_{\mu i}^a$ are zero-mean Gaussian variables whose covariances are $\mathrm{Cov}^{\setminus\mu}\left(D_{\mu i}, D_{\mu i}\right) = W_\mu$, $\mathrm{Cov}^{\setminus\mu}\left(D_{\mu i}, \Delta_{\mu i}^a\right) = 0$, $\mathrm{Cov}^{\setminus\mu}\left(\Delta_{\mu i}^a, \Delta_{\mu i}^b\right) = V_\mu\delta_{ab}$.

$\widetilde{\phi}_{\mu\to i}$ can now be computed. The integration with respect to $\{\boldsymbol{\beta}_j\}_{j(\neq i)}$ in (11) is replaced by that over $D_i$ and $\Delta_i^a$. The result is

$$\widetilde{\phi}_{\mu\to i}(\boldsymbol{\beta}_i) \approx \int dD_{\mu i}P(D_{\mu i}) \int \prod_a d\Delta_{\mu i}^a P(\Delta_{\mu i}^a) \left[ e^{-\frac{1}{2}\gamma c\sum_a (r_{\mu i} - x_{\mu i}\beta_i^a + D_{\mu i} + \Delta_{\mu i}^a)^2} \right]_c$$
$$= \left[ U(c)^{-\frac{1}{2}} \left(1 + c\gamma V_\mu\right)^{-\frac{n}{2}} e^{-\frac{1}{2}S(c)\sum_a \left(r_{\mu i} - x_{\mu i}\beta_i^a\right)^2 + \frac{1}{2}T(c)\left(\sum_a (r_{\mu i} - x_{\mu i}\beta_i^a)\right)^2} \right]_c,$$

where

$$S(c) = \frac{c\gamma}{1+c\gamma V_\mu},$$
$$T(c) = \frac{c^2\gamma^2 W_\mu}{(1+c\gamma V_\mu)(1+c\gamma(V_\mu+nW_\mu))},$$
$$U(c) = \frac{1+c\gamma(V_\mu+nW_\mu)}{1+c\gamma V_\mu}.$$

$\ln \widetilde{\phi}_{\mu\to i}(\boldsymbol{\beta}_i)$ can be expanded with respect to $x_{\mu i}\beta_i^a$, and the expansion up to the second order leads to an effective Gaussian approximation of the message, yielding

$$\widetilde{\phi}_{\mu\to i}(\boldsymbol{\beta}_i) \approx e^{-\frac{1}{2}\gamma\tilde{A}_{\mu\to i}\sum_a\left(\beta_i^a\right)^2+\frac{1}{2}\gamma^2\tilde{C}_{\mu\to i}\left(\sum_a\beta_i^a\right)^2+\gamma\tilde{B}_{\mu\to i}\sum_a\beta_i^a},$$

where

$$\tilde{A}_{\mu\to i} = \left[\frac{c}{1+c\gamma V_\mu}\right]_c x_{\mu i}^2, \tag{16a}$$
$$\tilde{B}_{\mu\to i} = \left[\frac{c}{1+c\gamma V_\mu}\right]_c r_{\mu i}x_{\mu i}, \tag{16b}$$
$$\tilde{C}_{\mu\to i} = \left\{\left[\left(\frac{c}{1+c\gamma V_\mu}\right)^2\right]_c W_\mu + \left(\left[\left(\frac{c}{1+c\gamma V_\mu}\right)^2\right]_c - \left[\frac{c}{1+c\gamma V_\mu}\right]_c^2\right)(r_{\mu i})^2\right\} x_{\mu i}^2. \tag{16c}$$

It should be noted that the $n\to 0$ limit was already taken in these coefficients.

Owing to the Gaussian approximation of $\widetilde{\phi}_{\mu\to i}(\boldsymbol{\beta}_i)$, the marginal distribution of the replicated Boltzmann distribution can be simply written as

$$\begin{aligned} P_i(\boldsymbol{\beta}_i|D) &\equiv \int\prod_{j(\neq i)} d\boldsymbol{\beta}_j P(\{\boldsymbol{\beta}_i\}_i|D) \propto \Psi(\boldsymbol{\beta}_i)\prod_\mu \widetilde{\phi}_{\mu\to i}(\boldsymbol{\beta}_i) \\ &\approx \left[\int Dz\; e^{\gamma\left(-\frac{1}{2}A_i\sum_a\left(\beta_i^a\right)^2+(B_i+\sqrt{C_i}z)\sum_a\beta_i^a-\lambda\sum_a|\beta_i^a|\right)}\right]_\lambda, \end{aligned} \tag{17}$$

where $Dz = dze^{-\frac{1}{2}z^2}/\sqrt{2\pi}$, and the following identity was used:

$$e^{\frac{1}{2}Cx^2} = \int Dz\; e^{\sqrt{C}zx}.$$

Moreover, $A_i = \sum_\mu \tilde{A}_{\mu\to i}$, $B_i = \sum_\mu \tilde{B}_{\mu\to i}$, and $C_i = \sum_\mu \tilde{C}_{\mu\to i}$. All replicas are now factorized, and the average can be taken for each replica independently, allowing passing to the $n\to 0$ limit by analytic continuation from $\mathbb{N}$ to $\mathbb{R}$ with respect to $n$. For example, the mean of $\beta_i^a$ is computed as

$$\begin{aligned} \overline{\beta_i^a} &= \left[\int Dz\left(\int d\beta\, e^{\gamma\left(-\frac{1}{2}A_i\beta^2+(B_i+\sqrt{C_i}z)\beta_i-\lambda|\beta|\right)}\right)^n\right]_\lambda^{-1} \\ &\times\left[\int Dz\int d\beta\,\beta e^{\gamma\left(-\frac{1}{2}A_i\beta^2+(B_i+\sqrt{C_i}z)\beta_i-\lambda|\beta|\right)}\left(\int d\beta\, e^{\gamma\left(-\frac{1}{2}A_i\beta^2+(B_i+\sqrt{C_i}z)\beta_i-\lambda|\beta|\right)}\right)^{n-1}\right]_\lambda \\ &\xrightarrow{n\to 0}\left[\int Dz\frac{\int d\beta\,\beta e^{\gamma\left(-\frac{1}{2}A_i\beta^2+(B_i+\sqrt{C_i}z)\beta_i-\lambda|\beta|\right)}}{\int d\beta\, e^{\gamma\left(-\frac{1}{2}A_i\beta^2+(B_i+\sqrt{C_i}z)\beta_i-\lambda|\beta|\right)}}\right]_\lambda \xrightarrow{\gamma\to\infty}\left[\int DzS_\lambda(B_i+\sqrt{C_i}z;A_i)\right]_\lambda, \end{aligned}$$

where $S_\lambda$ is the so-called soft thresholding function

$$S_\lambda(x;A) = \frac{1}{A}(x - \lambda \mathrm{sgn}\,(x))\theta(|x|-\lambda),$$

and $\theta(x)$ is the step function that is equal to 1 if $x > 0$ and 0 otherwise. Thus, the mean of the basic estimator takes a reasonable form.

In the above average, it is assumed that the coefficients $A_i, B_i, C_i$ remain finite as $\gamma \to \infty$. This is the case if the following holds:

$$\chi_\mu \equiv \gamma V_\mu = \sum_i x_{\mu i}^2 \gamma V_i \to O(1),\ (\gamma \to \infty).$$

This scaling is consistent, which can be shown by

$$\begin{aligned}
&\chi_i \equiv \gamma V_i = \gamma \left\{ \overline{(\beta_i^a)^2 - \beta_i^a \beta_i^b} \right\} \\
&\xrightarrow{n\to 0} \left[ \int Dz\, \gamma \left\{ \frac{\int d\beta\, \beta^2 e^{\gamma\left(-\frac{1}{2}A_i\beta^2 + (B_i + \sqrt{C_i}z)\beta_i - \lambda|\beta|\right)}}{\int d\beta\, e^{\gamma\left(-\frac{1}{2}A_i\beta^2 + (B_i + \sqrt{C_i}z)\beta_i - \lambda|\beta|\right)}} - \left( \frac{\int d\beta\, \beta e^{\gamma\left(-\frac{1}{2}A_i\beta + (B_i + \sqrt{C_i}z)\beta_i - \lambda|\beta|\right)}}{\int d\beta\, e^{\gamma\left(-\frac{1}{2}A_i\beta^2 + (B_i + \sqrt{C_i}z)\beta_i - \lambda|\beta|\right)}} \right)^2 \right\} \right]_\lambda \\
&\xrightarrow{\gamma\to\infty} \frac{1}{A_i} \left[ \int Dz \theta\left( |B_i + \sqrt{C_i}z| - \lambda \right) \right]_\lambda = \frac{\partial \overline{\beta_i}}{\partial B_i}.
\end{aligned}$$

In contrast to $V_i$, the inter-sample fluctuation $W_i$ takes a finite value even as $\gamma \to \infty$. Its explicit form is

$$\begin{aligned}
&W_i = \overline{\beta_i^a \beta_i^b} - \overline{\beta_i^a}\,\overline{\beta_i^b} \\
&\xrightarrow{n\to 0,\ \gamma\to\infty} \left[ \int Dz S_\lambda^2 \left( B_i + \sqrt{C_i}z; A_i \right) \right]_\lambda - \left( \left[ \int Dz S_\lambda \left( B_i + \sqrt{C_i}z; A_i \right) \right]_\lambda \right)^2.
\end{aligned}$$

It follows that any moment of the basic estimator can be computed from (17), once the coefficients $\{A_i, B_i, C_i\}_{i=1}^N$ are correctly estimated. Hence, the next task is to derive a set of self-consistent equations for the coefficients and to construct an algorithm for solving it.

### 3.2. Self-consistent equations and a message passing algorithm

Using (12), we obtain

$$\begin{aligned}
&\phi_{i\to\mu}(\boldsymbol{\beta}_i) \propto \Psi\left(\boldsymbol{\beta}_i\right) \prod_{\nu(\neq\mu)} \widetilde{\phi}_{\nu\to i}(\boldsymbol{\beta}_i) \\
&\approx \left[ \int Dz\ e^{\gamma\left(-\frac{1}{2}A_{i\to\mu}\sum_a \left(\beta_i^a\right)^2 + (B_{i\to\mu} + \sqrt{C_{i\to\mu}}z)\sum_a \beta_i^a - \lambda \sum_a |\beta_i^a|\right)} \right]_\lambda,
\end{aligned}$$

where $A_{i\to\mu} = \sum_{\nu(\neq\mu)} \tilde{A}_{\nu\to i}$, $B_{i\to\mu} = \sum_{\nu(\neq\mu)} \tilde{B}_{\nu\to i}$, and $C_{i\to\mu} = \sum_{\nu(\neq\mu)} \tilde{C}_{\nu\to i}$. Inserting this into (11), we can derive a set of self-consistent equations determining all the cavity coefficients $\{A_{i\to\mu}, B_{i\to\mu}, C_{i\to\mu}, \tilde{A}_{\mu\to i}, \tilde{B}_{\mu\to i}, \tilde{C}_{\mu\to i}\}_{i,\mu}$. This procedure suggests an iterative

algorithm, conventionally called BP algorithm, which is schematically described as follows:

$$\{\tilde{A}_{\mu\to i}, \tilde{B}_{\mu\to i}, \tilde{C}_{\mu\to i}\}^{(t)} \leftarrow \{\overline{\beta_i}^{\setminus\mu}, V_i, W_i\}^{(t)}, \tag{18a}$$

$$\{A_{i\to\mu}, B_{i\to\mu}, C_{i\to\mu}\}^{(t+1)} \leftarrow \{\tilde{A}_{\mu\to i}, \tilde{B}_{\mu\to i}, \tilde{C}_{\mu\to i}\}^{(t)}, \tag{18b}$$

$$\{\overline{\beta_i}^{\setminus\mu}, V_i, W_i\}^{(t+1)} \leftarrow \{A_{i\to\mu}, B_{i\to\mu}, C_{i\to\mu}\}^{(t+1)}, \tag{18c}$$

where $t = 0, 1, \cdots$, denotes the algorithm time step. If the BP algorithm converges, the full coefficients $\{A_i, B_i, C_i\}_{i=1}^N$ are given from the converged values of $\{\tilde{A}_{\mu\to i}, \tilde{B}_{\mu\to i}, \tilde{C}_{\mu\to i}\}_{i,\mu}$. However, this algorithm is not particularly efficient because its computational cost is $O(NM^2)$.

A more efficient algorithm is derived by approximately rewriting the cavity coefficients and the cavity mean $\overline{\boldsymbol{\beta}}^{\setminus\mu}$ using the full coefficients $\{A_i, B_i, C_i\}_i$. To this end, $\phi_{i\to\mu}(\boldsymbol{\beta}_i)$ is connected with $P_i(\boldsymbol{\beta}_i|D)$ in a perturbative manner. Comparing the coefficients, it follows that the difference between $A_i$ and $A_{i\to\mu}$ is negligibly small, as it is proportional to $x_{\mu i}^2 = O(1/N)$. The same is true between $C_i$ and $C_{i\to\mu}$. Hence, the relevant difference is only $\Delta B_i^{(t)} = B_i^{(t)} - B_{i\to\mu}^{(t)}$ and is expressed as

$$\begin{aligned}\Delta B_i^{(t)} &= \left[\frac{c}{1+c\chi_\mu^{(t-1)}}\right]_c r_{\mu i}^{(t-1)} x_{\mu i} = a_\mu^{(t-1)} x_{\mu i} + \left[\frac{c}{1+c\chi_\mu^{(t-1)}}\right]_c \left(\overline{\beta_i}^{\setminus\mu}\right)^{(t-1)} x_{\mu i}^2 \\ &\approx a_\mu^{(t-1)} x_{\mu i},\end{aligned}$$

where

$$a_\mu^{(t)} \equiv \left[\frac{c}{1+c\chi_\mu^{(t)}}\right]_c \left(y_\mu - \sum_j x_{\mu j}\left(\overline{\beta_j}^{\setminus\mu}\right)^{(t)}\right) = \left[\frac{c}{1+c\chi_\mu^{(t)}}\right]_c \left(r_{\mu i}^{(t)} - x_{\mu i}\left(\overline{\beta_i}^{\setminus\mu}\right)^{(t)}\right). \tag{19}$$

Accordingly, the difference between $\overline{\beta_i}^{\setminus\mu}$ and $\overline{\beta_i}$ is computed as

$$\left(\overline{\beta_i}^{\setminus\mu}\right)^{(t)} \approx \overline{\beta_i}^{(t)} - \frac{\partial\overline{\beta_i}^{(t)}}{\partial B_i^{(t)}}\Delta B_i^{(t)} = \overline{\beta_i}^{(t)} - \chi_i^{(t)} a_\mu^{(t-1)} x_{\mu i}, \tag{20}$$

Inserting (20) into (19) yields

$$a_\mu^{(t)} = \left[\frac{c}{1+c\chi_\mu^{(t)}}\right]_c \left(y_\mu - \sum_j x_{\mu j}\overline{\beta_j}^{(t)} + \chi_\mu^{(t)} a_\mu^{(t-1)}\right),$$

where the last term is interpreted as the Onsager reaction term in physics. Rewriting $r_{\mu i}$ using $a_\mu$ in the right-hand sides of (16) and collecting several factors, a simplified message

passing algorithm corresponding to (18) is obtained as follows:

$$\chi_\mu^{(t)} = \sum_i x_{\mu i}^2 \chi_i^{(t)}, \tag{21a}$$

$$W_\mu^{(t)} = \sum_i x_{\mu i}^2 W_i^{(t)}, \tag{21b}$$

$$\left(f_{1\mu}^{(t)}, f_{2\mu}^{(t)}\right) = \left(\left[\frac{c}{1+c\chi_\mu^{(t)}}\right]_c, \left[\left(\frac{c}{1+c\chi_\mu^{(t)}}\right)^2\right]_c\right), \tag{21c}$$

$$a_\mu^{(t)} = f_{1\mu}^{(t)}\left(y_\mu - \sum_j x_{\mu j}\overline{\beta_j}^{(t)} + \chi_\mu^{(t)} a_\mu^{(t-1)}\right), \tag{21d}$$

$$A_i^{(t+1)} = \sum_\mu x_{\mu i}^2 f_{1\mu}^{(t)}, \tag{21e}$$

$$B_i^{(t+1)} = \sum_\mu x_{\mu i} a_\mu^{(t)} + \left(\sum_\mu x_{\mu i}^2 f_{1\mu}^{(t)}\right)\overline{\beta_i}^{(t)}, \tag{21f}$$

$$C_i^{(t+1)} = \sum_\mu x_{\mu i}^2 \left\{ f_{2\mu}^{(t)} W_\mu^{(t)} + \left(f_{2\mu}^{(t)} - \left(f_{1\mu}^{(t)}\right)^2\right)\left(\frac{a_\mu^{(t)}}{f_{1\mu}^{(t)}}\right)^2 \right\}, \tag{21g}$$

$$\overline{\beta_i}^{(t+1)} = \left[\int Dz S_\lambda\left(B_i^{(t+1)} + \sqrt{C_i^{(t+1)}}z; A_i^{(t+1)}\right)\right]_\lambda, \tag{21h}$$

$$\chi_i^{(t+1)} = \frac{1}{A_i^{(t+1)}}\left[\int Dz\ \theta\left(\left|B_i^{(t+1)} + \sqrt{C_i^{(t+1)}}z\right| - \lambda\right)\right]_\lambda, \tag{21i}$$

$$W_i^{(t+1)} = \left[\int Dz S_\lambda^2\left(B_i^{(t+1)} + \sqrt{C_i^{(t+1)}}z; A_i^{(t+1)}\right)\right]_\lambda - \left(\overline{\beta_i}^{(t+1)}\right)^2. \tag{21j}$$

We call the algorithm (21) AMPR (Approximate Message Passing with Resampling) because it can be regarded as an extension of the AMP in the usual Lasso to the resampling case. The computational cost is $O(NM)$ per iteration and is significantly reduced compared with the BP algorithm. This yields the main result of this study.

There is an ambiguity in the initial condition for AMPR. Here we assume that we are given an initial estimate $\{\overline{\beta_i}^{(0)}, \chi_i{}^{(0)}, W_i{}^{(0)}\}_i$ and conduct the iteration based on (21) until convergence. Still, there is an ambiguity in computing $a_\mu^{(0)}$, due to the presence of $a_\mu^{(-1)}$ in (21d). To resolve this, we assume $a_\mu^{(-1)} = 0$, yielding

$$a_\mu^{(0)} = \left[\frac{c}{1+c\chi_\mu^{(0)}}\right]_c\left(y_\mu - \sum_j x_{\mu j}\overline{\beta_j}^{(0)}\right). \tag{22}$$

These completely determine the initial condition.

As explained at the end of Section 3.1, the convergent solution of AMPR, $\{A_i^*, B_i^*, C_i^*\}_i$, enables the computation of any moment of the basic estimator as follows:

$$[\langle\beta_i\rangle^r]_{\boldsymbol{c},\boldsymbol{\lambda}} = \lim_{n\to 0}\overline{\prod_{a=1}^r \beta_i^a} = \left[\int Dz S_\lambda^r\left(B_i^* + \sqrt{C_i^*}z; A_i^*\right)\right]_\lambda,$$

which indicates that the marginal distribution of the basic estimator is obtained as

$$P(\beta_i) = \left[ \int Dz\ \delta\left(\beta_i - S_\lambda\left(B_i^* + \sqrt{C_i^*}z; A_i^*\right)\right)\right]_\lambda .$$

This yields the positive probability, which is important for variable selection techniques, as follows:

$$\Pi_i \equiv \mathrm{Prob}\left(|\hat{\beta}_i| \neq 0\right) = \left[ \int Dz\ \theta\left(\left|B_i^* + \sqrt{C_i^*}z\right| - \lambda\right)\right]_\lambda . \tag{23}$$

Applications using these relations are provided in Section 4.

### 3.3. State evolution for AMPR

A benefit of the AMP type algorithms is that it is possible to track the macroscopic dynamical behavior of the algorithm. This can be done by using the so-called state evolution (SE) equations. Here we derive the SE equations associated with AMPR. The derivation relies on the i.i.d. assumption of the covariates, and hence we assume each $x_{\mu i}$ is i.i.d. from the zero-mean Gaussian as

$$x_{\mu i} \sim \mathcal{N}(0, \sigma_x^2). \tag{24}$$

The variance value is arbitrary in general, but for notational simplicity it is chosen as $\sigma_x^2 = 1/N$ in this section. Furthermore, we assume the data is generated from the following linear process:

$$\boldsymbol{y} = X\boldsymbol{\beta}_0 + \boldsymbol{\xi},$$

where $\boldsymbol{\xi}$ is the noise vector whose component is i.i.d. from $\mathcal{N}(0, \sigma_\xi^2)$ and $\boldsymbol{\beta}_0$ is the true parameters whose component is also i.i.d. from a certain distribution $P_{\beta_0}(\cdot)$.

Under the assumption (24) with $\sigma_x^2 = 1/N$, the intra- and inter-sample variances can be simplified as

$$\chi_\mu = \sum_i x_{\mu i}^2 \chi_i \approx \sum_i \mathrm{E}\left[x_{\mu i}^2 \chi_i\right] \approx \frac{1}{N}\sum_i \chi_i \equiv \tilde{\chi}, \tag{25}$$

$$W_\mu = \sum_i x_{\mu i}^2 W_i \approx \sum_i \mathrm{E}\left[x_{\mu i}^2 W_i\right] \approx \frac{1}{N}\sum_i W_i \equiv \tilde{W}, \tag{26}$$

where we have neglected the correlations between $x_{\mu i}$ and the variances[2]. Accordingly, many quantities appearing in (21) become independent of the subscripts $\mu$ and $i$. Terms retaining the dependence are only the linear terms with respect to $x_{\mu i}$ such as $\overline{\beta}_i, B_i$ and $a_\mu$. To derive the SE equations, we need to handle those terms.

2. Remembering the discussion in Section 3.2, these variances are actually the ones computed in the absence of the $\mu$-th potential function, $\chi_i^{\setminus\mu}$ and $W_i^{\setminus\mu}$, and hence this neglect can be justified. The same discussion can be applied to the following.

We start from the following form of $B_i^{(t+1)}$:

$$B_i^{(t+1)} = \sum_\nu \left[\frac{c}{1+c\chi_\nu}\right]_c r_{\nu i}^{(t)} x_{\nu i} \approx f_1^{(t)} \sum_\nu x_{\nu i} \left( y_\nu - \sum_{j(\neq i)} x_{\nu j} \left(\bar{\beta}_j^{\setminus \nu}\right)^{(t)} \right). \tag{27}$$

The righthand side is the sum of a large number of random variables, and hence we can treat it as a Gaussian variable with appropriate mean and variance. The mean is

$$\begin{aligned}
&\mathrm{E}\left[ f_1^t \sum_\nu x_{\nu i} \left( y_\nu - \sum_{j(\neq i)} x_{\nu j} \left(\bar{\beta}_j^{\setminus \nu}\right)^{(t)} \right) \right] \\
&= f_1^t \sum_\nu \left( \mathrm{E}\left[x_{\nu i}^2\right] \beta_{0i} + \sum_{j(\neq i)} \mathrm{E}\left[x_{\nu i} x_{\nu j}\right] \left( \beta_{0j} - \left(\bar{\beta}_j^{\setminus \mu}\right)^{(t)} \right) + \mathrm{E}\left[x_{\nu i}\xi_\nu\right] \right) \\
&= f_1^t \sum_\nu \frac{1}{N} \beta_{0i} = \alpha f_1^t \beta_{0i},
\end{aligned} \tag{28}$$

where $\alpha = M/N$ is the ratio of the dataset size to the dimensionality. In the same way, the variance becomes

$$\mathrm{V}\left[ f_1^t \sum_\nu x_{\nu i} \left( y_\nu - \sum_{j(\neq i)} x_{\nu i} \left(\bar{\beta}_j^{\setminus \nu}\right)^{(t)} \right) \right] \approx \alpha \left(f_1^t\right)^2 \left( \mathrm{MSE}^{(t)} + \sigma_\xi^2 \right) \equiv v_0^{(t+1)}, \tag{29}$$

where $\mathrm{MSE}^{(t)}$ denotes the mean-squared error (MSE) between the true and averaged parameters:

$$\mathrm{MSE}^{(t)} \equiv \frac{1}{N} \sum_{i=1}^N \left( \beta_{0i} - \overline{\beta}_i^{(t)} \right)^2 \approx \frac{1}{N} \sum_{i=1}^N \left( \beta_{0i} - \left(\overline{\beta}_i^{\setminus \mu}\right)^{(t)} \right)^2. \tag{30}$$

Hence, we may write

$$B_i^{(t+1)} = \alpha f_1^t \beta_{0i} + \sqrt{v_0^{(t+1)}} u_i, \tag{31}$$

where $u_i \sim \mathcal{N}(0,1)$. Besides, in the computation of $C_i^{(t+1)}$, we have

$$\sum_\nu x_{\nu i}^2 f_{2\nu}^{(t)} W_\nu^{(t)} \approx \alpha f_2^{(t)} \tilde{W}^{(t)}, \tag{32}$$

$$\sum_\nu x_{\nu i}^2 \left( r_{\nu i}^{(t)} \right)^2 \approx \alpha \left( \mathrm{MSE}^{(t)} + \sigma_\xi^2 \right). \tag{33}$$

This yields

$$C_i^{(t+1)} \approx C^{(t+1)} = \alpha f_2^{(t)} \tilde{W}^{(t)} + \alpha \left( f_2^{(t)} - \left( f_1^{(t)} \right)^2 \right) \left( \mathrm{MSE}^{(t)} + \sigma_\xi^2 \right). \tag{34}$$

In the same level of approximation, we get $A_i^{(t+1)} \approx A^{(t+1)} = \alpha f_1^{(t)}$.

To derive a closed set of equations, we have to compute $\tilde{\chi}^{(t+1)}, \tilde{W}^{(t+1)}$ and $\mathrm{MSE}^{(t+1)}$ from $\{A^{(t+1)}, \{B_i^{(t+1)}\}_i, C^{(t+1)}\}$. As an example, we show the derivation of $\tilde{\chi}^{(t+1)}$ from (21i) in the following:

$$\begin{aligned}\tilde{\chi}^{(t+1)} &= \frac{1}{N}\sum_{i=1}^{N}\chi_i^{(t+1)} \approx \frac{1}{N}\sum_{i=1}^{N}\frac{1}{A^{(t+1)}}\left[\int Dz\ \theta\left(\left|B_i^{(t+1)}+\sqrt{C^{(t+1)}}z\right|-\lambda\right)\right]_\lambda \\ &\approx \frac{1}{A^{(t+1)}}\int d\beta P_{\beta_0}(\beta)\int Du\left[\int Dz\ \theta\left(\left|\alpha f_1^{(t)}\beta+\sqrt{v_0^{(t+1)}}u+\sqrt{C^{(t+1)}}z\right|-\lambda\right)\right]_\lambda, \quad (35)\end{aligned}$$

where we have applied the law of large numbers. The other quantities $\tilde{W}, \mathrm{MSE}$ are computed in the same manner. Overall, we reach the following set of equations:

$$\left(f_1^{(t)}, f_2^{(t)}\right) = \left(\left[\frac{c}{1+c\tilde{\chi}^{(t)}}\right]_c, \left[\left(\frac{c}{1+c\tilde{\chi}^{(t)}}\right)^2\right]_c\right), \quad (36a)$$

$$A^{(t+1)} = \alpha f_1^{(t)}, \quad (36b)$$

$$C^{(t+1)} = \alpha f_2^{(t)}\tilde{W}^{(t)} + \alpha\left(f_2^{(t)} - \left(f_1^{(t)}\right)^2\right)\left(\mathrm{MSE}^{(t)}+\sigma_\xi^2\right), \quad (36c)$$

$$v_0^{(t+1)} = \alpha\left(f_1^{(t)}\right)^2\left(\mathrm{MSE}^{(t)}+\sigma_\xi^2\right), \quad (36d)$$

$$\begin{aligned}\tilde{\chi}^{(t+1)} &= \frac{1}{A^{(t+1)}}\int d\beta P_{\beta_0}(\beta)\int Du \\ &\quad\times\left[\int Dz\ \theta\left(\left|A^{(t+1)}\beta+\sqrt{v_0^{(t+1)}}u+\sqrt{C^{(t+1)}}z\right|-\lambda\right)\right]_\lambda, \quad (36e)\end{aligned}$$

$$\begin{aligned}\tilde{W}^{(t+1)} &= \int d\beta P_{\beta_0}(\beta)\int Du\left\{\left[\int DzS_\lambda^2\left(A^{(t+1)}\beta+\sqrt{v_0^{(t+1)}}u+\sqrt{C^{(t+1)}}z; A^{(t+1)}\right)\right]_\lambda\right. \\ &\quad\left.-\left[\int DzS_\lambda\left(A^{(t+1)}\beta+\sqrt{v_0^{(t+1)}}u+\sqrt{C^{(t+1)}}z; A^{(t+1)}\right)\right]_\lambda^2\right\}, \quad (36f)\end{aligned}$$

$$\begin{aligned}\mathrm{MSE}^{(t+1)} &= \int d\beta P_{\beta_0}(\beta)\int Du \\ &\quad\times\left\{\beta-\left[\int DzS_\lambda\left(A^{(t+1)}\beta+\sqrt{v_0^{(t+1)}}u+\sqrt{C^{(t+1)}}z; A^{(t+1)}\right)\right]_\lambda\right\}^2. \quad (36g)\end{aligned}$$

Given an initial condition $\{\tilde{\chi}^{(0)}, \tilde{W}^{(0)}, \mathrm{MSE}^{(0)}\}$, we can track the dynamical evolution of those quantities according to (36). This is the SE equations for AMPR.

A direct consequence of the SE equations is the convergence property of AMPR: Its convergence depends on neither the dataset size $M$ nor the model dimensionality $N$. Hence, we can assume the iteration steps required for convergence is $O(1)$ and the total computational cost of AMPR is thus guaranteed to be $O(NM)$. This reinforces the superiority of the present approach.

We, however, warn that the discussion based on the SE equations strongly relies on the i.i.d. assumption among covariates, which is not necessarily satisfied in real-world datasets. For covariates with non-trivial correlations and heterogeneity, it is known that

AMP type algorithms tend to show slow convergence, or even not to converge in particular cases (Caltagirone et al., 2014). A common prescription to overcome this difficulty is to introduce a damping factor in the update of the messages (Rangan et al., 2014), which is also employed in our implementation (Obuchi, 2018). In Sections 4.1.5 and 4.2, we see how this prescription works for datasets with nontrivial covariates in numerical simulations.

## 4. Numerical experiments

In this section, the accuracy and the computational time of the proposed semi-analytic method based on AMPR is examined by a comparison with direct numerical resampling. We also check how nontrivial correlations among covariates affect the performance of AMPR. Both simulated and real-world datasets (from UCI machine learning repository, Lichman, 2013) are used.

For all experiments involving numerical resampling, *Glmnet* (Friedman et al., 2010), implemented as an *MEX* subroutine in MATLAB®, was employed for solving (9), given a sample $\{\boldsymbol{\lambda}, D_{\boldsymbol{c}}\}$. Moreover, the proposed AMPR algorithm was implemented as raw code in MATLAB. This is not the most optimized approach because AMPR uses a number of *for* and *while* loops which are slow in MATLAB; hence, the comparison of computational time is not necessarily fair. However, even in this comparison, there is a significant difference in the computational time between the proposed semi-analytic method and the numerical resampling approach. For reference, it should be noted that all experiments below were conducted in a single thread on a single CPU of Intel(R) Xeon(R) E5-2630 v3 2.4GHz.

For actual computations, the distribution $P(\boldsymbol{\lambda})$ should be specified. In SS, the following distribution is used (Meinshausen and Bühlmann, 2010):

$$P(\boldsymbol{\lambda}) = \prod_{i=1}^{N} \left\{ p_w \delta\left(\lambda_i - \lambda/w\right) + (1-p_w)\delta\left(\lambda_i - \lambda\right) \right\},$$

with $0 < w \leq 1$ and $0 < p_w < 1$. The case of the non-random penalty coefficient, in which Bolasso is included (Bach, 2008), is recovered at $w = 1$, irrespective of the value of $p_w$. This distribution is adopted below.

### 4.1. Simulated dataset

Here, simulated datasets are treated. The data is supposed to be generated from the following linear model:

$$\boldsymbol{y} = X\boldsymbol{\beta}_0 + \boldsymbol{\xi},$$

where each component of the design matrix $X = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$ is i.i.d. from $\mathcal{N}\left(0, N^{-1}\right)$, and $\boldsymbol{\xi}$ is the noise vector, whose component is i.i.d. from $\mathcal{N}(0, \sigma_\xi^2)$. The ratio of the dataset size $M$ to the model dimensionality $N$ is denoted as $\alpha \equiv M/N$ hereafter. These settings are identical to the ones assumed in Section 3.3. The true signal $\boldsymbol{\beta}_0 \in \mathbb{R}^N$ is assumed to be $K_0(= N\rho_0)$-sparse vector, and the non-zero components are i.i.d. from $\mathcal{N}(0, 1/\rho_0)$, setting the power of the signal unity. The index set of non-zero components is denoted by $S_0 = \{i | |\beta_{0i}| \neq 0\}$ and is called true support. Any estimator of the true support is simply called support and is hereafter denoted by $S$.

For simplicity, the number of resampling times is fixed at $N_{\mathrm{res}} = 1000$ in the experiments with numerical resampling.

#### 4.1.1. Accuracy of the semi-analytic method

Let us first check the consistency between the results of our semi-analytic method and of the direct numerical resampling.

Figure 1 shows the plots of the experimental values of the quantities $\{\overline{\beta_i}, W_i, \Pi_i\}_i$ against their semi-analytic counterparts for the non-random penalty case $w = 1$ with the bootstrap resampling $\tau = 1$, which is the situation considered in Bolasso. The same plots in the SS situation, $w = 0.5(< 1)$, $p_w = 0.5$, and $\tau = 0.5$, are shown in Figure 2. Other detailed parameters are provided in the captions. These results show that the proposed semi-analytic

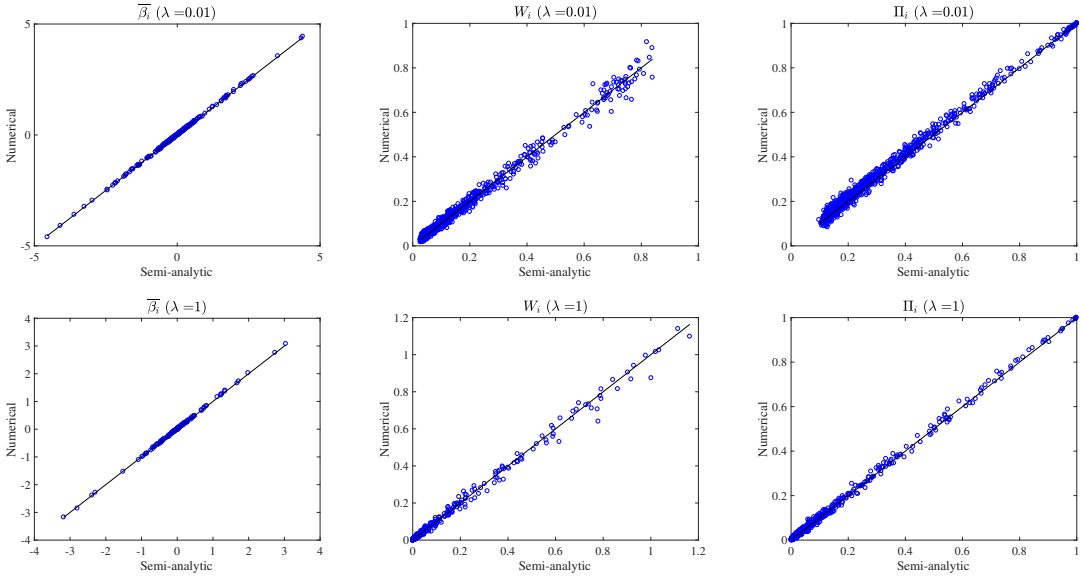


Figure 1: Experimental values of $\overline{\beta_i}$ (left), $W_i$ (middle), and $\Pi_i$ (right) are plotted against those computed by the semi-analytic method in the non-random penalty $w = 1$ and $\tau = 1$ case. The upper panels are for $\lambda = 0.01$ and the lower are for $\lambda = 1$. The other parameters are set to be $(N, \alpha, \rho_0, \sigma_\xi^2) = (1000, 0.5, 0.2, 0.01)$.

method reproduces the numerical results fairly accurately. As far as it was examined, results of similar accuracy were obtained for a very wide range of parameters. These validate the proposed semi-analytic method.

In the above experiments using Glmnet, we set a threshold $\epsilon$ to judge the algorithm convergence as $\epsilon = 10^{-10}$, which is rather tighter than the default value. This is necessary for examining consistency with the proposed semi-analytic method. For example, for $\lambda = 0.01$ (the upper panels in Figures 1 and 2), a systematic deviation from the proposed semi-analytic method ($\overline{\beta}_i$ tends to be underestimated) emerges at the default value $\epsilon = 10^{-7}$.

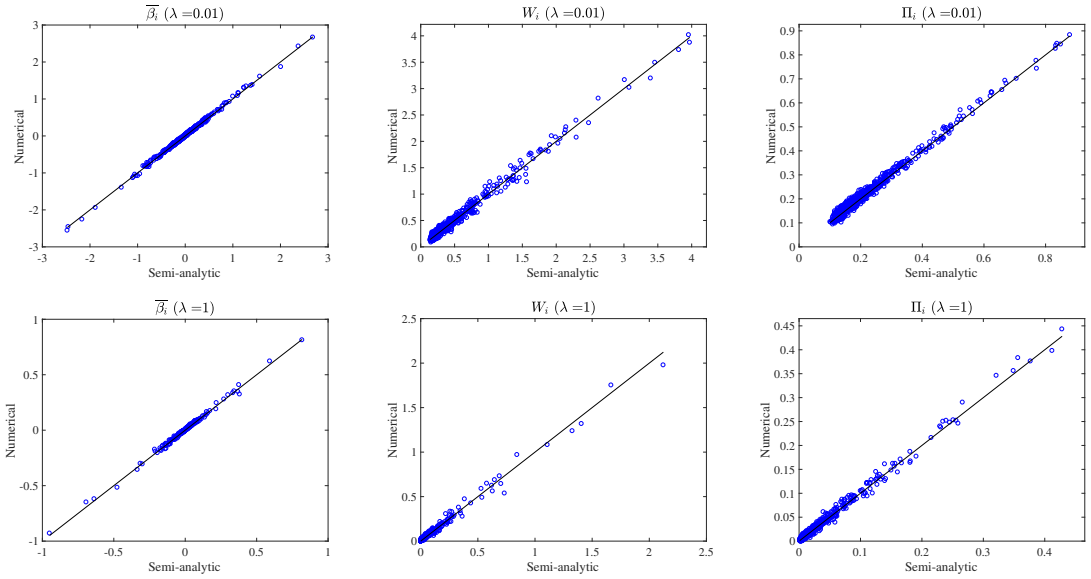


Figure 2: Same plots as in Figure 1 in the random penalty case with $w = 0.5$, $p_w = 0.5$, and $\tau = 0.5$. The parameters of each panel are identical to the corresponding parameters in Figure 1. In comparison to Figure 1, $|\overline{\beta}_i|$ and $\Pi_i$ tend to be smaller, whereas $W_i$ tends to be larger. This is probably due to the additional stochastic variation coming from the randomization on the penalty coefficient and the difference between $\tau = 1$ and $1/2$.

This implies that a rather tight threshold is required for microscopic quantities such as $\overline{\beta_i}$ and $W_i$. Unless explicitly mentioned, this value $\epsilon = 10^{-10}$ is used below.

#### 4.1.2. Comparison with state evolution

To examine the convergence properties of AMPR, we next see the dynamical behavior of the macroscopic quantities $(\tilde{\chi}^{(t)}, \tilde{W}^{(t)}, \mathrm{MSE}^{(t)})$ as the algorithm step $t$ proceeds, in comparison with the SE equations (36). Figure 3 shows the plots of them against $t$ for different parameters. In all the cases, these macroscopic quantities rapidly take stable values and

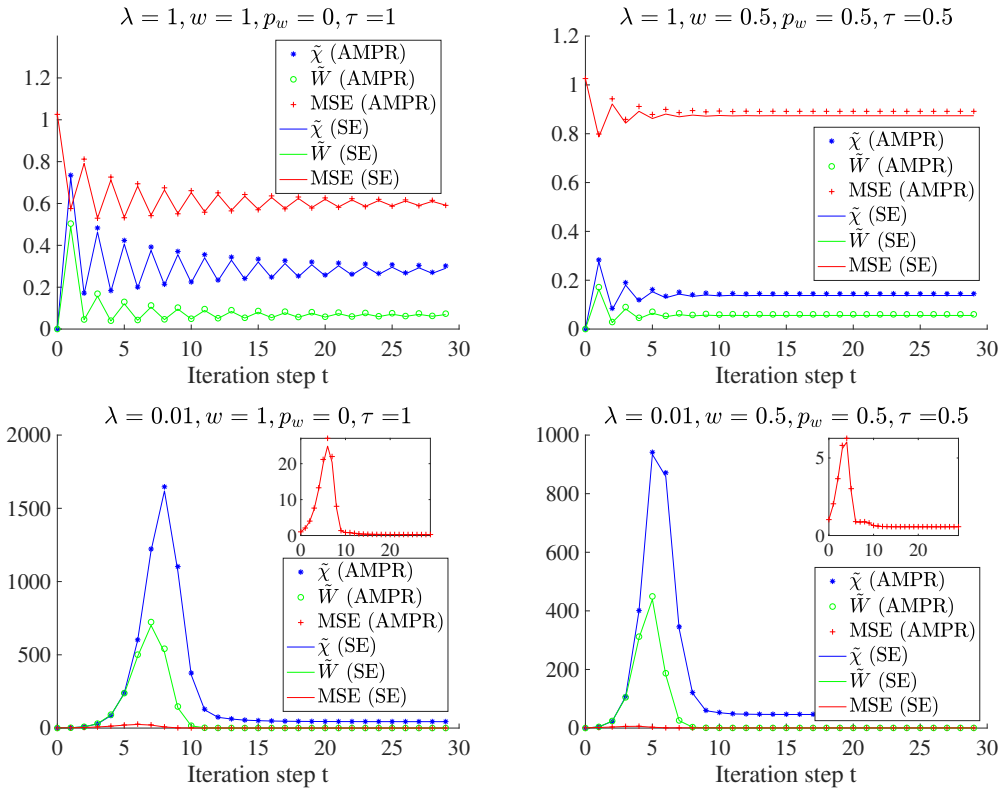


Figure 3: Dynamical behavior of macroscopic parameters $(\tilde{\chi}^{(t)}, \tilde{W}^{(t)}, \mathrm{MSE}^{(t)})$ initialized at $(\tilde{\chi}^{(0)}, \tilde{W}^{(0)}) = (0, 0)$ and $\mathrm{MSE}^{(0)} \approx 1$. The result computed by SE is denoted by lines while the one by AMPR is represented by markers, and the agreement is fairly good. (Left) The non-randomized penalty case $(w = 1, p_w = 0, \tau = 1)$. (Right) The randomized penalty case $(w = p_w = \tau = 1/2)$. The upper panels are for $\lambda = 1$, while the lower ones are of $\lambda = 0.01$ for which insets are given to show the MSE values in visible scales. The AMPR result is obtained at $N = 20000$. The other parameters are fixed at $(\alpha, \rho_0, \sigma_\xi^2) = (0.5, 0.2, 0.01)$.

the agreement between the AMPR and SE results is excellent. This demonstrates the fast convergence of AMPR, which does not depend on both $N$ and $M$. To see a good agreement

with the SE result by just one sample of $\{\boldsymbol{\beta}_0, X, \boldsymbol{\xi}\}$, the model dimensionality $N$ is chosen as a rather large value of $N = 20000$ in the AMPR experiment. Although in Figure 3 the initial condition is fixed at $\boldsymbol{\chi}^{(0)} = \boldsymbol{W}^{(0)} = \overline{\boldsymbol{\beta}}^{(0)} = \mathbf{0}$ corresponding to $(\tilde{\chi}^{(0)}, \tilde{W}^{(0)}) = (0, 0)$ and $\mathrm{MSE}^{(0)} \approx 1$, we also examined several other initial conditions and confirmed the good agreement between the AMPR and SE results in all the cases.

#### 4.1.3. Application to Bolasso

Bolasso is a variable selection method utilizing the positive probability (23) evaluated by the bootstrap resampling $\tau = 1$ with no penalty randomization $w = 1$. Its soft version, abbreviated as Bolasso-S (Bach, 2008), selects variables with $\Pi_i \geq 0.9$ as active variables and other variables are rejected from the support. We here adopt this manner and see the performance of AMPR used for implementing this.

The left panel of Figure 4 shows the plot of the true positive ratio (TP) against the false positive ratio (FP) as the values of $\alpha$ change. Bolasso requires scaling the regularization parameter as $\lambda \propto \sqrt{\alpha}$, and here, it is set to $\lambda = (1/2)\sqrt{\alpha}$. The other parameters are fixed at $(N, \rho_0, \sigma_\xi^2) = (1000, 0.2, 0.01)$. As was theoretically shown (Bach, 2008), TP converges to unity, whereas FP tends to zero in this setup. This demonstrates that the

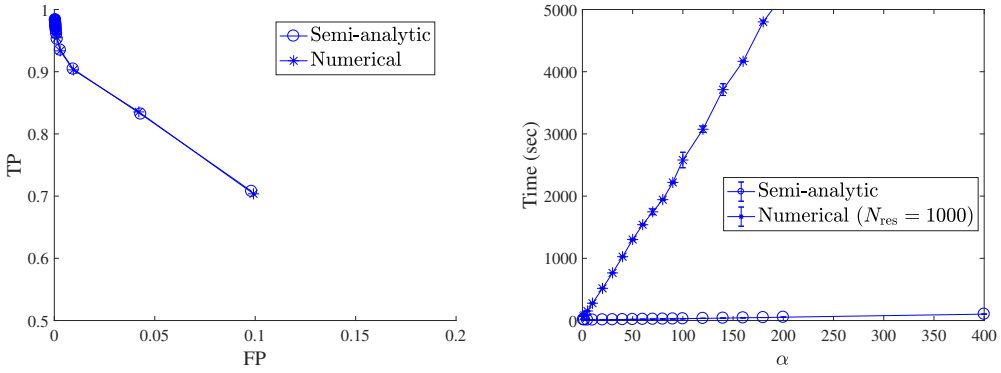


Figure 4: Bolasso experiment at $(N, \rho_0, \sigma_\xi^2) = (1000, 0.2, 0.01)$. The semi-analytic result (circle) is plotted with the numerical resampling result (asterisk). (Left) TP is plotted against FP as $\alpha$ increases: TP approaches unity, whereas FP tends to zero. The semi-analytic result completely overlaps that of the numerical resampling. Error bars are omitted for clarity. (Right) Computational time plotted against $\alpha$. The observed large difference is attributed to the numerical resampling cost.

model consistency in the variable selection context is recovered. The contribution of this study is a significant reduction of the computational time: In the right panel, the actual computational time is compared between the semi-analytic method using AMPR and the direct numerical resampling, by plotting it against $\alpha$. The computational costs of AMPR and Glmnet are both scaled as $O(NM)$ and thus should be scaled linearly with respect to $\alpha$ for fixed $N$. Figure 4 clearly shows this linearity. The significant difference in compu-

tational time is fully attributed to the numerical resampling cost. This demonstrates the efficiency of AMPR. It should be noted that the average over 10 different samples of $D$ is taken in Figure 4 to obtain smooth curves and error bars.

#### 4.1.4. Application to stability selection

SS is another variable selection method utilizing positive probability. The difference from Bolasso is the presence of the penalty coefficient randomization ($w < 1$) and that $\tau$ is set to 0.5. These introduce a further stochastic variation in the method, and consequently they tend to show a clearer discrimination in the positive probabilities between the variables in and outside the true support. AMPR can easily implement this, and here, it is compared with numerical resampling.

Figure 5 shows the plots of the positive probability values against $\lambda$, the so-called stability path (Meinshausen and Bühlmann, 2010), and the computational time for obtaining the stability path against the covariate dimensionality $N$. The other parameters are fixed at $(\alpha, \rho_0, \sigma_\xi^2, w, p_w) = (2, 0.2, 0.01, 0.5, 0.5)$. Drawing all stability paths $\{\Pi_i\}_i$ leads to an

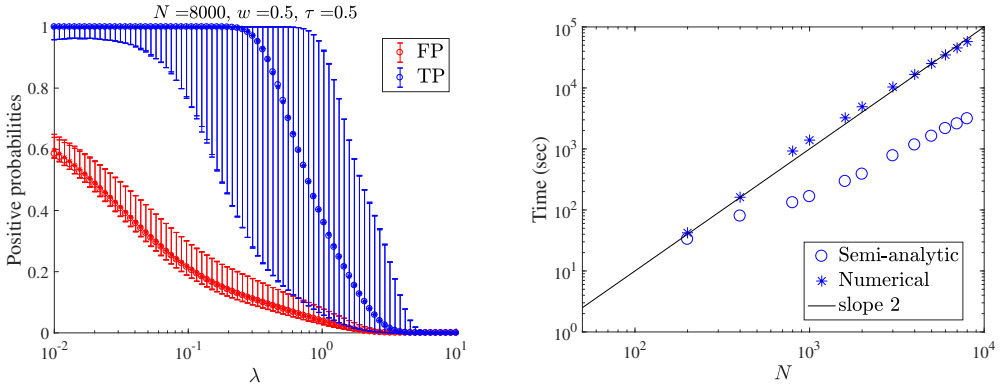


Figure 5: SS experiment at $(\alpha, \rho_0, \sigma_\xi^2, w, p_w) = (2, 0.2, 0.01, 0.5, 0.5)$. The semi-analytic result (circle) is plotted with the numerical resampling result (asterisk). (Left) Plots of the medians (points) and $q$-percentiles (bars) of $\{\Pi_i\}_{i\in S_0}$ (TP) and $\{\Pi_i\}_{i\notin S_0}$ (FP) with $q = 16$ and 84 against $\lambda$ for $N = 8000$. The semi-analytic result well overlaps that of the numerical resampling. There is a clear gap between TP and FP, suggesting that an accurate variable selection is possible. (Right) Computational time plotted against $N$ on double-logarithmic scale. The dominant reason for the difference is again the numerical resampling cost.

unclear plot; thus, only the median and $q$-percentiles of $\{\Pi_i\}_{i\in S_0}$ and those of $\{\Pi_i\}_{i\notin S_0}$ are plotted, which are denoted by TP and FP in the left panel of Figure 5, respectively. The medians are represented by points, and the percentiles of $q = 16$ and 84, approximately corresponding to one-sigma points in normal distributions, are represented by bars. There is a clear gap between TP and FP, suggesting that an accurate variable selection is possible by setting an appropriate threshold of probability value, as discussed by Meinshausen and

Bühlmann (2010). The right panel shows a plot of the computational time by AMPR and by the numerical resampling. Both are supposed to be scaled as $O(NM = N^2)$. Although such a scaling is not observed in the AMPR result, it is expected to appear for larger $N$. Again, a significant difference in computational time is present between the semi-analytic and the numerical resampling methods, demonstrating the effectiveness of the proposed method.

#### 4.1.5. Correlated covariates

In the derivation of AMPR and the associated SE equations, the weakness of correlations between covariates is assumed, but this assumption does not necessarily hold in realistic situations. Hence, it is important to check the accuracy of results obtained by AMPR for datasets with correlated covariates. Here we numerically examine this point.

To introduce correlations into the simulated dataset described above in a systematic way, we generate our covariates $\{\boldsymbol{x}_i\}_{i=1}^N$ in the following manner: As a common component we first generate a vector $\boldsymbol{x}^{\rm com} \in \mathbb{R}^M$ each component of which is i.i.d. from $\mathcal{N}(0, 1/N)$; choose a number $0 \leq r^{\rm com} < 1$ controlling the ratio of the common component and generate a binary vector $\boldsymbol{m}_i$ each component of which independently takes 1 with probability $r^{\rm com}$; take another vector $\tilde{\boldsymbol{x}}_i \in \mathbb{R}^M$ each component of which is i.i.d. from $\mathcal{N}(0, 1/N)$, and generate a covariate vector $\boldsymbol{x}_i$ as a linear combination between $\boldsymbol{x}^{\rm com}$ and $\tilde{\boldsymbol{x}}_i$ with using $\boldsymbol{m}_i$ as a mask. These operations can be summarized in the following equation:

$$\boldsymbol{x}_i = \boldsymbol{x}^{\rm com} \circ \boldsymbol{m}_i(r^{\rm com}) + \tilde{\boldsymbol{x}}_i \circ (1 - \boldsymbol{m}_i(r^{\rm com})), \tag{37}$$

where $\circ$ denotes Hadamard (component-wise) product. The covariates' correlations monotonically increase as $r^{\rm com}$ grows. To quantify this, we compute an overlap between the covariates, $\mathrm{overlap}_{ij} = \boldsymbol{x}_i^\top \boldsymbol{x}_j / \left( \sqrt{\boldsymbol{x}_i^\top \boldsymbol{x}_i} \sqrt{\boldsymbol{x}_j^\top \boldsymbol{x}_j} \right)$, and plot its mean value over $i$ and $j$ $(\neq i)$ in Figure 6. Keeping in mind this quantitative information, we check the accuracy of AMPR below.

To directly see the accuracy of AMPR for correlated covariates, we plot the experimental values of the quantities $\{\overline{\beta_i}, W_i, \Pi_i\}_i$ against those of AMPR in Figure 7 for two different values of the common component ratio, $r^{\rm com} = 0.4$ and $r^{\rm com} = 0.8$. For $r^{\rm com} = 0.8$, the AMPR result shows a clear deviation from the experimental one, while for $r^{\rm com} = 0.4$ it still exhibits a good agreement with the experiment. Although Figure 7 is the specific result to the non-random penalty case $(w = 1, \tau = 1)$ with parameters $(N, \alpha, \rho_0, \sigma_\xi^2, \lambda) = (1000, 0.5, 0.2, 0.01, 1)$, we have tested several different cases and observed similar tendency. To capture more global and quantitative information, we introduce a normalized MSE of $\overline{\beta}$ between the experimental $\overline{\beta}^{\rm exp}$ and AMPR $\overline{\beta}^{\rm AMPR}$ values as

$$\frac{\sum_{i=1}^N \left( \overline{\beta}_i^{\rm exp} - \overline{\beta}_i^{\rm AMPR} \right)^2}{\sum_{i=1}^N \left( \overline{\beta}_i^{\rm AMPR} \right)^2}. \tag{38}$$

The counterparts for $W$ and $\Pi$ are defined in the same way. Plots of the normalized MSEs for these quantities against $r^{\rm com}$ are given in Figure 8. Here the results for both the randomized $(w = p_w = \tau = 0.5)$ and non-randomized $(w = \tau = 1, p_w = 0)$ penalty

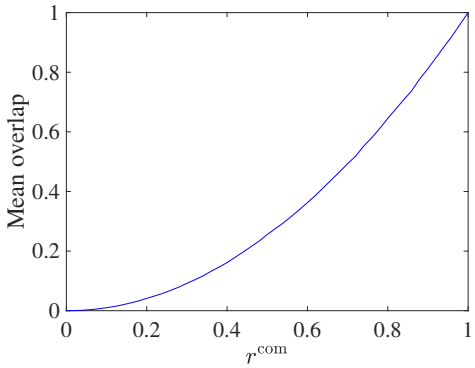


Figure 6: Plot of mean overlap of covariates against the common component ratio $r^{\mathrm{com}}$. This is a result of a single numerical simulation at $(N, \alpha) = (1000, 0.5)$, but the invariability of the result against changing the parameters has been also checked.

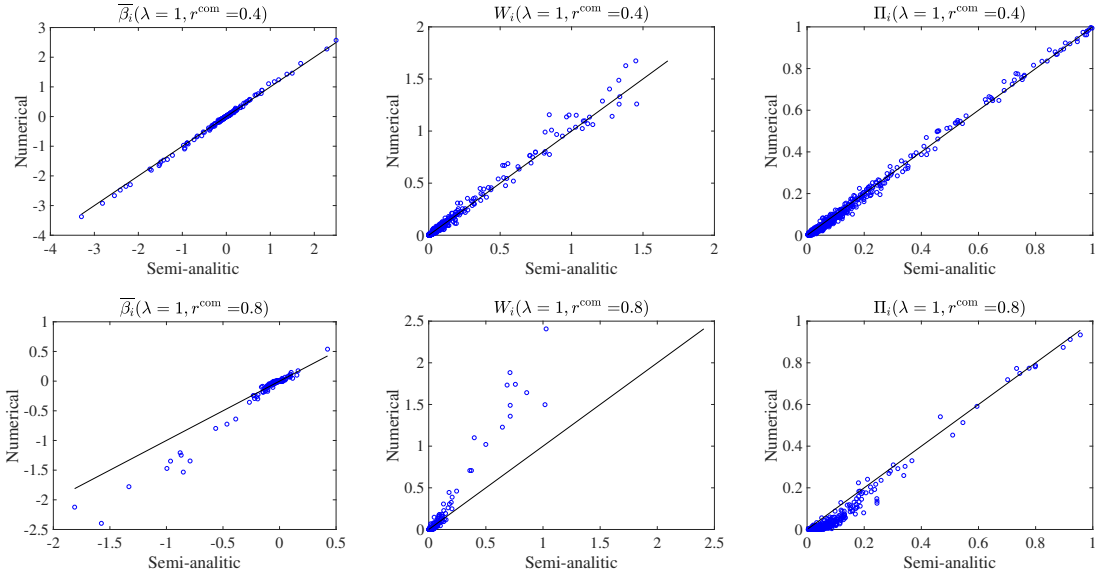


Figure 7: Experimental values of $\overline{\beta_i}$ (left), $W_i$ (middle), and $\Pi_i$ (right) are plotted against those computed by AMPR for correlated covariates with $r^{\mathrm{com}} = 0.4$ (upper) and $r^{\mathrm{com}} = 0.8$ (lower). For the lower panels, there exists a systematic deviation in the AMPR result. Here the non-random penalty case $(w = 1, \tau = 1)$ is treated and the other parameters are $(N, \alpha, \rho_0, \sigma_\xi^2, \lambda) = (1000, 0.5, 0.2, 0.01, 1)$.

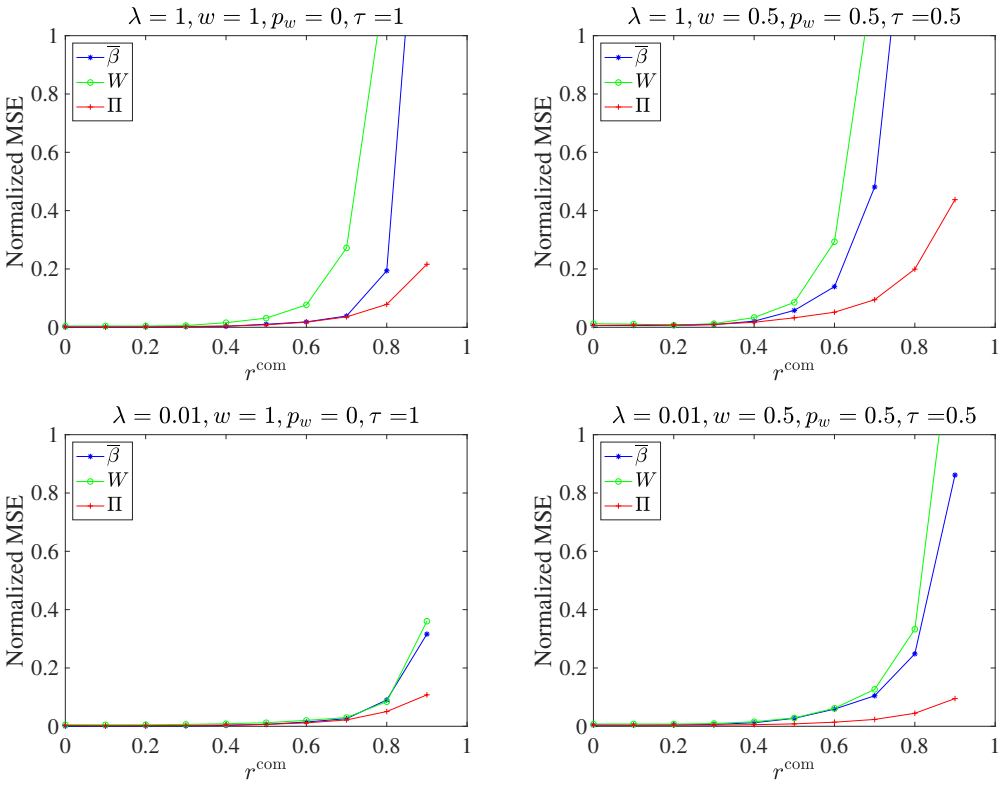


Figure 8: Plots of normalized MSEs of $\overline{\beta}, W, \Pi$ against $r^{\mathrm{com}}$ of the non-randomized (left, $w = 1, p_w = 0, \tau = 1$) and randomized (right, $w = p_w = \tau = 1/2$) penalty cases. The upper panels are for $\lambda = 1$ while the lower ones are of $\lambda = 0.01$. The other parameters are $(N, \alpha, \rho_0, \sigma_\xi^2) = (1000, 0.5, 0.2, 0.01)$.

cases are presented, with two different values of $\lambda$. This suggests that AMPR is practically reliable up to a certain level of correlations: For example if $r^{\rm com} \leq 0.6$, the normalized MSE of $\overline{\beta}$ is well suppressed and is commonly less than 0.2. According to Figure 6, $r^{\rm com} = 0.6$ corresponds to the mean overlap value of about 0.36 which is not small. This speaks for the effectiveness of AMPR even for real-world datasets, as long as the correlations are not too large.

The above discussion clarifies the accuracy of AMPR for correlated covariates, but a more crucial issue is in its convergent property. For the non-correlated case of $r^{\rm com} = 0$, AMPR converges very rapidly as shown in Figure 3, but for correlated cases it tends to badly converge and can even diverge. A common way to overcome this is to introduce a damping factor $\gamma$ in the update (Caltagirone et al., 2014; Rangan et al., 2014), and we adopted this in the above experiments. The update with the damping factor can be symbolized as

$$\boldsymbol{\theta}^{(t+1)} = (1-\gamma)\boldsymbol{\theta}^{(t)} + \gamma G(\boldsymbol{\theta}^{(t)}), \tag{39}$$

where $\boldsymbol{\theta}$ is a variable summarizing $(\overline{\boldsymbol{\beta}}, \boldsymbol{\chi}, \boldsymbol{W})$ and $G$ represents the AMPR operations. The original message passing update corresponds to $\gamma = 1$. Smaller values of $\gamma$ are better for the stability of the algorithm but takes a longer time until the convergence. Our experiments on the simulated dataset seemingly show that a smaller value of $\gamma$ is needed for larger $r^{\rm com}$ and smaller $\lambda$. For example for obtaining Figure 7, we needed to set $\gamma < 0.1$ for avoiding divergence. This value is found experimentally, and it is desired to find a more principled way to choose an appropriate value of $\gamma$ or a more effective way to better control the convergence of AMP type algorithms. Some earlier studies tackled this problem (Caltagirone et al., 2014; Rangan et al., 2014), but the complete understanding of the convergent property is still missing. Investigation along this direction is an important topic but is beyond the purpose of the present paper.

### 4.2. Real-world dataset

In this subsection, AMPR is applied to a real-world dataset, and the stability path is computed for examining the relevant covariates or variables. The dataset treated here is the wine quality dataset (Lichman, 2013): The data size is $M = 4898$ (only white wine is treated), and the number of covariates, which represent physicochemical aspects of wine, is $N_0 = 11$. The basic aim of this dataset is to model wine quality in terms of the physicochemical aspects only, and the response is an integer-valued quality score from zero to ten obtained by human expert evaluation. This dataset can be used in both classification and regression, and the linear model was also tested in earlier studies (Cortez et al., 2009; Melkumova and Shatskikh, 2017). Lasso and SS are applied to this dataset.

As preprocessing, $N_{\rm noise}$ noise variables are added into the dataset. Each component of the noise variables is i.i.d. from $\mathcal{N}(0, 1/N)$. The usual standardization, zeroing the mean of all variables and responses and normalizing the variables to be of unit norm, is also conducted.

The reason of the introduction of the noise variables is to make an objective criterion for judging relevant stability paths. The noise variables are by definition irrelevant for describing the responses and hence if some stability paths of original variables behave similarly to the ones of the noise variables then those variables can be regarded as irrelevant. A distribution of stability paths of the noise variables thus defines a kind of "rejection region",

resolving the arbitrariness of judging criterion of the positive probability both in Bolasso and SS. As far as we have searched, this kind of active construction of rejection regions has never been observed in literatures, and hence this proposition itself can be regarded as a part of new results of this paper.

This sort of manipulation to dataset is usually unfavored because it increases the computational cost and tends to contaminate the dataset. However thanks to AMPR, the first computational issue becomes less serious since the computational time of AMPR increases just linearly with respect to the number of added noise variables $N_{\rm noise}$. The second issue does not seem to be serious either for the wine quality dataset, because the dataset size $M = 4898$ is very large compared to the number of original variables $N_0 = 11$. Below, we experimentally check if this strategy works well or not.

Let us start from checking how the noise variables influence on the dataset. To this end, we plot the solution paths and the generalization errors estimated by 10-fold cross validation (CV) in Figure 9 in the cases with and without the noise variables. This figure

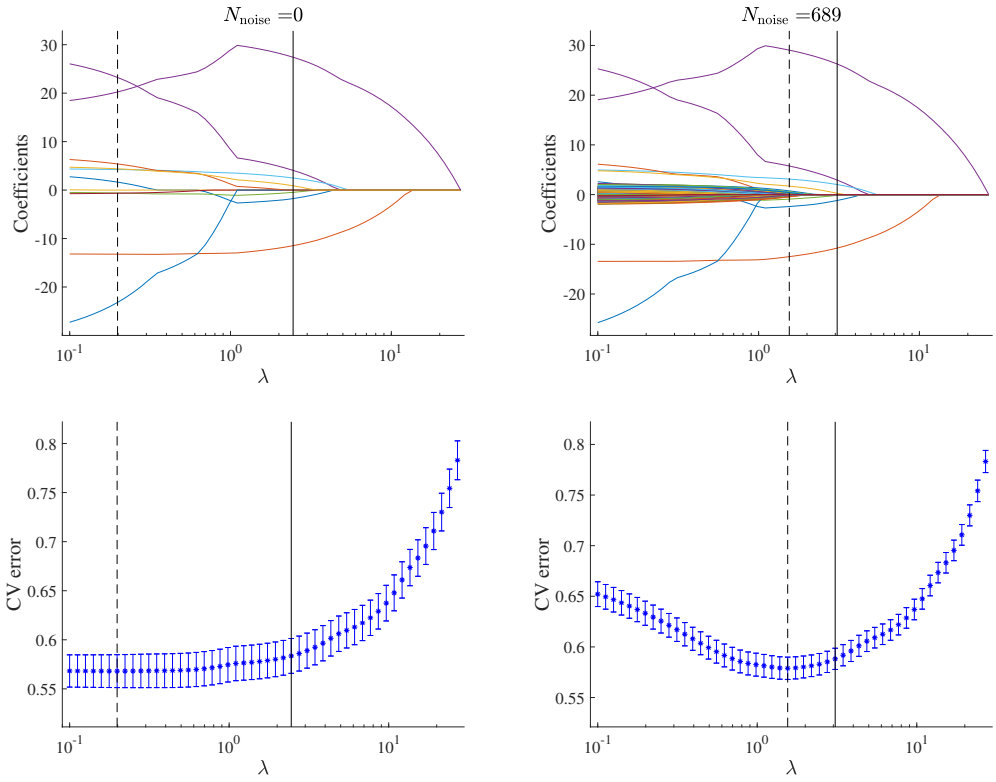


Figure 9: Solution paths (top) and CV errors plotted against $\lambda$ (bottom), for $N$noise $= 0$ (left) and $N$noise $= 689$ (right). The vertical dotted line denotes the location of the CV error minimum, $\lambda_{\rm min}$, whereas the vertical straight line represents the location selected by the one-standard-error rule that "defines" the optimal $\lambda$ value, $\lambda_{\rm opt}$. The effect of the noise variables addition is weak around $\lambda_{\rm opt}$.

Table 1: Overlap of the 3rd covariate (citric acid) and the others

| index $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| overlap $\boldsymbol{x}_i^\top \boldsymbol{x}_3$ | 0.29 | -0.15 | 1.00 | 0.09 | 0.11 | 0.09 | 0.12 | 0.15 | -0.16 | 0.06 | -0.07 |

indicates that the solution paths of the original variables and the CV error value are stable against the introduction of noise variables, particularly in the relevant range of $\lambda$. At the optimal $\lambda$ ($\lambda_{\mathrm{opt}}$), chosen by the one-standard-error rule (Trevor et al., 2009), the variables in the support are common in both cases: They are indexed as $1, 2, 4, 5, 6, 10$, and 11, which represent fixed acidity, volatile acidity, residual sugar, chlorides, free sulfur dioxide, sulphates, and alcohol, respectively. Hence, we can conclude that the introduction of the noise variables hardly affects the estimates of the original variables, and we can effectively use the noise variables to judge the significance of each original variable.

Subsequently, the result of applying SS with $(\tau, w, p_w) = (0.5, 0.5, 0.5)$ to this dataset is examined. Figure 10 shows the plots of the stability paths according to the three categories of variables: The "important" variables are defined as being in the support, at $\lambda_{\mathrm{opt}}$ in the Lasso analysis, and thus their indices are $1, 2, 4, 5, 6, 10, 11$; the "neutral" variables are the other variables in the original dataset, and the corresponding indices are $3, 7, 8, 9$; the remaining are the noise variables. Both the results of AMPR (circle) and the direct numerical resampling (asterisk) are shown (the same variable is depicted in the same color).

Consequences of Figure 10 are three-fold. The first one is consistent agreement between the results of AMPR and the direct numerical resampling. For all categories, the two curves of each stability path $\Pi_i(\lambda)$ by AMPR and the direct numerical resampling are very close to each other. This is an additional evidence supporting the accuracy of the proposed semi-analytic method in real-world datasets with correlated covariates. The computational time for obtaining these results in an experiment is 1077 s by AMPR and 2859 s by the numerical resampling, demonstrating the efficiency of AMPR. It should be stressed that this efficiency can be enhanced by optimizing the implementation of AMPR. The second consequence is the different behaviors of stability paths in different categories. The positive probabilities of the important variables (upper left) are growing largely even for $\lambda > \lambda_{\mathrm{min}}$, while those of the noise variables (lower left) do not grow well unless $\lambda$ drops below $\lambda_{\mathrm{min}}$. The behavior of the neutral variables (upper right) is somewhat elusive, and a better interpretation is provided by utilizing the noise variables, which is the third consequence: As discussed above, we can define a rejection region from the distribution of stability paths of the noise variables; an actual definition here is the $q$-percentiles with $q = 16$ and 84 of the distribution, which is in the lower right panel depicted by red bars with the median (red markers) of the distribution, the legend of which is given as FP, in the same manner as Figure 5. This analysis shows that citric acid and total sulfur dioxide of the neutral variables tend to be in the rejection region, implying that they are irrelevant for modeling wine quality. Moreover, density and pH are well beyond the interval of the rejection region, and they can be regarded as relevant, even though the behavior of density's path is rather tricky.

The above conclusion of citric acid's irrelevance contradicts that in Cortez et al. (2009), where citric acid was concluded to be the fourth most important variable. This may be explained by the collinearity of the covariates. In Table 1, the covariates' overlap between citric acid and the others is summarized. This shows a strong collinearity of the 3rd variable

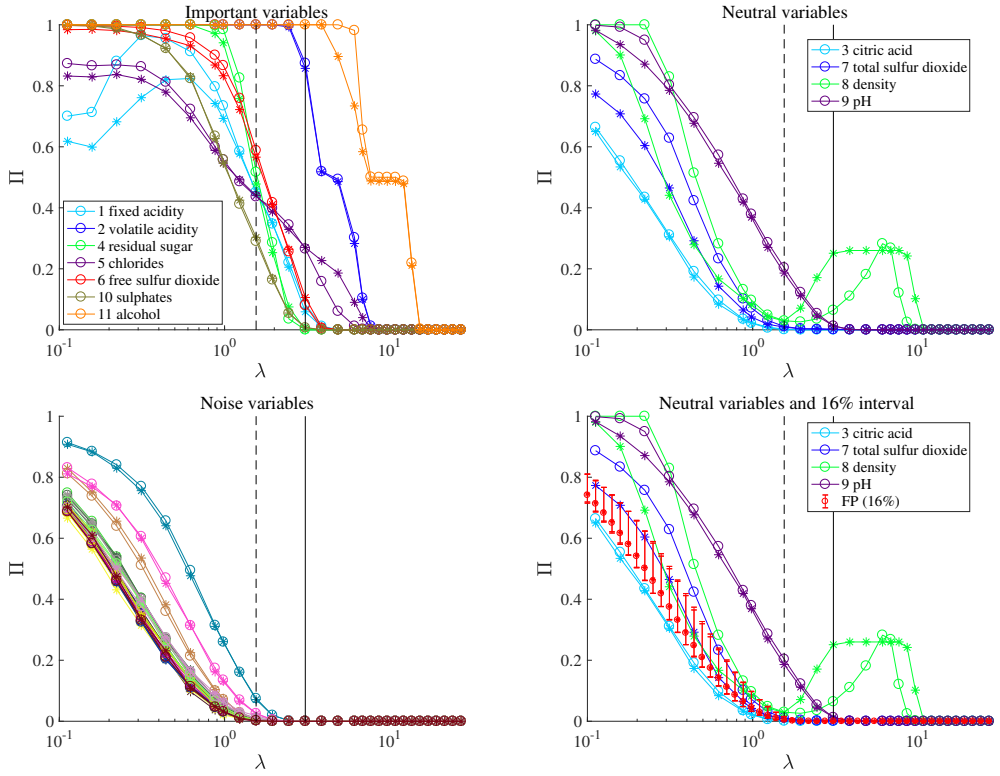


Figure 10: Stability paths for the wine quality data with $N_{\rm noise} = 689$. The paths of the important variables are in the upper left, those of the neutral variables are in the upper right, and those of the noise variables are in the lower left panels. For the lower left panel, the paths of only a part of the noise variables are shown. The vertical dotted and straight lines denote $\lambda_{\rm min}$ and $\lambda_{\rm opt}$, respectively, as in Figure 9. Circles denote the result by AMPR and asterisks represent that by the direct numerical resampling, and the same variable is depicted in the same color; the semi-analytic and numerical results show a consistent agreement for all cases. The lower right panel is the simultaneous plot of the stability paths of the neutral variables and the FP interval computed from paths of the noise variables.

with several others, particularly with the 1st variable. This implies that citric acid can be replaced by the 1st variable or fixed acidity. This is a plausible explanation because the proposed model puts considerably more weight on fixed acidity than on citric acid, whereas the opposite is the case in Cortez et al. (2009). Further thorough comparison will be required to determine which model is better.

Table 1 also implies why AMPR works well for the wine quality dataset: The maximum value of the covariates's overlap is about 0.29 which corresponds $r^{\mathrm{com}} \approx 0.5$ in Figure 6; Figure 8 shows that the accuracy of AMPR is commonly good around that value of $r^{\mathrm{com}}$, explaining the accuracy of AMPR. This consideration indicates that given a new dataset we can judge whether AMPR will give an accurate result or not for the dataset from the covariates' overlap and Figures 6 and 8.

Overall, the analysis using stability paths provides richer information that cannot be obtained by solely using Lasso. A new objective criterion could be proposed for determining the relevance of variables by utilizing the distribution of stability paths of the added noise variables. These facts highlight the effectiveness of the resampling strategy in variable selection, and the proposed semi-analytic method can implement this strategy in a computationally efficient manner.

## 5. Conclusion

An approximate method was developed for performing resampling in Lasso in a semi-analytic manner. The replica method enables us to analytically take the resampling average over the given data and the average over the penalty coefficient randomness, and the resultant replicated model is approximately handled by a Gaussian approximation using the cavity method. A message passing algorithm named AMPR is thus derived, the computational cost of which is $O(NM)$ per iteration and is sufficiently reasonable. Its convergence in $O(1)$ iterations is guaranteed by a state evolution analysis when covariates are given as zero-mean i.i.d. Gaussian random variables. We demonstrated how it actually works through numerical experiments using simulated and real-world datasets. Comparison with direct numerical resampling has evidenced its approximation accuracy and efficiency in terms of computational cost, even for covariates with correlations of a moderate level. AMPR was also employed to approximately perform Bolasso and SS, and it was applied to the wine quality dataset (Lichman, 2013; Cortez et al., 2009). To provide a finer quantitative analysis of the dataset, an objective criterion was proposed for determining the relevance of the stability paths by processing the added noise variables, yielding reasonable results in satisfactory computational time.

An advantage of the present framework is its generality. For example, its extension to a generalized linear model is straightforward. This is an immediate future research direction. Extensions to other resampling techniques, such as the multiscale bootstrap method (Shimodaira et al., 2004), would also be interesting.

An unsatisfactory aspect of the present AMPR is that the correlations between covariates are neglected by the approximation. This is a clear drawback, and certain issues arise when the present AMPR is applied to problems involving significantly correlated covariates, although numerical experiments showed that the accuracy of AMPR is still good in the presence of correlations of a moderate level. This drawback may be overcome using more

sophisticated approximations, such as the expectation propagation or the adaptive TAP method (Opper and Winther, 2001a,b, 2005; Kabashima and Vehkapera, 2014; Çakmak et al., 2014; Cespedes et al., 2014; Rangan et al., 2016; Ma and Ping, 2017; Takeuchi, 2017). Applying those approximations with retaining the benefit of message passing algorithms, the low computational cost, is still a nontrivial challenge and promising future work.

Resampling is a very versatile framework applicable to various contexts and models in statistics and machine learning. Reducing its computational cost by extending the present method will thus be beneficial in various fields, and can even be imperative, as available data in society will continue to increase rapidly.

## Acknowledgement

This work was supported by JSPS KAKENHI Nos. 25120013 and 17H00764. TO is also supported by a Grant for Basic Science Research Projects from the Sumitomo Foundation. The authors thank Hideitsu Hino for useful comments.

## References

Francis R Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the 25th international conference on Machine learning*, pages 33–40. ACM, 2008.

Onureena Banerjee, Laurent El Ghaoui, Alexandre d'Aspremont, and Georges Natsoulis. Convex optimization techniques for fitting sparse gaussian graphical models. In *Proceedings of the 23rd international conference on Machine learning*, pages 89–96. ACM, 2006.

Jean Barbier, Nicolas Macris, Mohamad Dia, and Florent Krzakala. Mutual information and optimality of approximate message-passing in random linear estimation. *arXiv preprint arXiv:1701.05823*, 2017.

Mohsen Bayati and Andrea Montanari. The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Transactions on Information Theory*, 57 (2):764–785, 2011.

Francesco Caltagirone, Lenka Zdeborová, and Florent Krzakala. On convergence of approximate message passing. In *2014 IEEE International Symposium on Information Theory*, pages 1812–1816, June 2014. doi: 10.1109/ISIT.2014.6875146.

Burak Çakmak, Ole Winther, and Bernard H Fleury. S-amp: Approximate message passing for general matrix ensembles. In *Information Theory Workshop (ITW), 2014 IEEE*, pages 192–196. IEEE, 2014.

Javier Cespedes, Pablo M Olmos, Matilde Sánchez-Fernández, and Fernando Perez-Cruz. Expectation propagation detection for high-order high-dimensional mimo systems. *IEEE Transactions on Communications*, 62(8):2840–2849, 2014.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

David L Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.

Viktor Dotsenko. *Introduction to the replica theory of disordered statistical systems*, volume 4. Cambridge University Press, 2005.

Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

Max Grazier G'Sell, Jonathan Taylor, and Robert Tibshirani. Adaptive testing for the graphical lasso. *arXiv preprint arXiv:1307.4765*, 2013.

Edwin Hewitt and Leonard J Savage. Symmetric measures on cartesian products. *Transactions of the American Mathematical Society*, 80(2):470–501, 1955.

Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014a.

Adel Javanmard and Andrea Montanari. Hypothesis testing in high-dimensional regression under the gaussian random design model: Asymptotic theory. *IEEE Transactions on Information Theory*, 60(10):6522–6554, 2014b.

Adel Javanmard and Andrea Montanari. De-biasing the lasso: Optimal sample size for gaussian designs. *arXiv preprint arXiv:1508.02757*, 2015.

Yoshiyuki Kabashima. A cdma multiuser detection algorithm on the basis of belief propagation. *Journal of Physics A: Mathematical and General*, 36(43):11111, 2003.

Yoshiyuki Kabashima and Mikko Vehkapera. Signal recovery using expectation consistent approximation for linear observations. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 226–230. IEEE, 2014.

Keith Knight and Wenjiang Fu. Asymptotics for lasso-type estimators. *Annals of statistics*, pages 1356–1378, 2000.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Richard Lockhart, Jonathan Taylor, Ryan J Tibshirani, and Robert Tibshirani. A significance test for the lasso. *Annals of statistics*, 42(2):413, 2014.

Junjie Ma and Li Ping. Orthogonal amp. *IEEE Access*, 5:2020–2033, 2017.

Dörthe Malzahn and Manfred Opper. An approximate analytical approach to resampling averages. *Journal of Machine Learning Research*, 4(Dec):1151–1173, 2003.

Nicolai Meinshausen and Peter Bühlmann. Consistent neighbourhood selection for sparse high-dimensional graphs with the lasso. Seminar für Statistik, Eidgenössische Technische Hochschule (ETH), Zürich, 2004.

Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, pages 246–270, 2009.

LE Melkumova and S Ya Shatskikh. Comparing ridge and lasso estimators for data analysis. *Procedia Engineering*, 201:746–755, 2017.

Marc Mézard, Giorgio Parisi, and Miguel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987.

Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

Hidetoshi Nishimori. *Statistical physics of spin glasses and information processing: an introduction*, volume 111. Clarendon Press, 2001.

Tomoyuki Obuchi. Matlab package of AMPR. `https://github.com/T-Obuchi/AMPR_lasso_matlab`, 2018.

Manfred Opper and Ole Winther. Adaptive and self-averaging thouless-anderson-palmer mean-field theory for probabilistic modeling. *Physical Review E*, 64(5):056131, 2001a.

Manfred Opper and Ole Winther. Tractable approximations for probabilistic models: The adaptive thouless-anderson-palmer mean field approach. *Physical Review Letters*, 86(17): 3695, 2001b.

Manfred Opper and Ole Winther. Expectation consistent approximate inference. *Journal of Machine Learning Research*, 6(Dec):2177–2204, 2005.

Sundeep Rangan, Philip Schniter, and Alyson Fletcher. On the convergence of approximate message passing with arbitrary matrices. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 236–240. IEEE, 2014.

Sundeep Rangan, Philip Schniter, and Alyson Fletcher. Vector approximate message passing. *arXiv preprint arXiv:1610.03082*, 2016.

Hidetoshi Shimodaira et al. Approximately unbiased tests of regions using multistep-multiscale bootstrap resampling. *The Annals of Statistics*, 32(6):2616–2641, 2004.

Takashi Takahashi and Yoshiyuki Kabashima. A statistical mechanics approach to de-biasing and uncertainty estimation in lasso for random measurements. *Journal of Statistical Mechanics: Theory and Experiment*, 2018(7):073405, 2018.

Keigo Takeuchi. Rigorous dynamics of expectation-propagation-based signal recovery from unitarily invariant measurements. In *Information Theory (ISIT), 2017 IEEE International Symposium on*, pages 501–505. IEEE, 2017.

Michel Talagrand. *Spin glasses: a challenge for mathematicians: cavity and mean field models*, volume 46. Springer Science & Business Media, 2003.

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

Hastie Trevor, Tibshirani Robert, and Friedman Jerome. *The Elements of Statistical Learning; Data Mining, Inference, and Prediction*. Springer-Verlag New York, 2009. doi: 10.1007/978-0-387-84858-7.

Martin J Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using $\ell_1$–constrained quadratic programming (lasso). *IEEE transactions on information theory*, 55(5):2183–2202, 2009.

Ming Yuan and Yi Lin. On the non-negative garrotte estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):143–161, 2007.

Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov):2541–2563, 2006.

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