Signal Processing 131 (2017) 1–7

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Short communication

High-dimensional variable selection in regression and classification with missing data

Qi Gao, Thomas C.M. Lee *

Department of Statistics, University of California, Davis, One Shields Avenue, Davis, CA 95616, USA

ARTICLE INFO

Article history:
Received 16 November 2015
Received in revised form
5 July 2016
Accepted 11 July 2016
Available online 15 July 2016

Keywords:
Adaptive lasso
Logistic regression
Low rank recovery
Matrix completion

ABSTRACT

Variable selection for high-dimensional data problems, including both regression and classification, has been a subject of intense research activities in recent years. Many promising solutions have been proposed. However, less attention has been given to the case when some of the data are missing. This paper proposes a general approach to high-dimensional variable selection with the presence of missing data when the missing fraction can be relatively large (e.g., 50%). Both regression and classification are considered. The proposed approach iterates between two major steps: the first step uses matrix completion to impute the missing data while the second step applies adaptive lasso to the imputed data to select the significant variables. Methods are provided for choosing all the involved tuning parameters. As fast algorithms and software are widely available for matrix completion and adaptive lasso, the proposed approach is fast and straightforward to implement. Results from numerical experiments and applications to two real data sets are presented to demonstrate the efficiency and effectiveness of the approach.

1. Introduction

High-dimensional data are encountered frequently nowadays in diverse fields of study and applications including biomedical research, finance, machine learning and signal processing. With rapid development of technology and increasing complexity of the contemporary scientific problems, both the number of instances and variables have been growing unprecedentedly. In particular, variable selection in the high-dimensional setting has drawn great attention and become a fertile field of research in recent years. Many methods have been proposed to solve this problem for both regression and classification; e.g., see [1,3,4,8,13,24,28,29].

In this paper we are interested in high-dimensional variable selection with the presence of missing data while the missing fraction can be fairly large (e.g., 50%). Although missing data occur frequently in various signal processing and statistical applications [25–27,30], there is not much work considering this problem in the existing literature, especially for classification. Some notable exceptions include [24] where the expectation–maximization (EM) algorithm is applied to estimate the inverse covariance matrix in sparse linear regression with missing data. However, like many other EM based solutions for complex problems, this EM-algorithm is computationally intensive and the missing fraction is usually relatively small. In [16] an algorithm is developed based on projected gradient descent for the cases of noisy and/or missing data in high-dimensional sparse linear regression, but again the missing fraction they study is relatively small.

In order to handle both high-dimensionality and high missing fraction, in this paper a new procedure that combines matrix completion techniques and adaptive lasso is developed for solving this variable selection problem. The proposed procedure is straightforward to implement, computationally fast, and capable of producing promising empirical results.

The rest of our paper is organized as follows. Section 2 provides a detailed description of the problem of interest, and presents the proposed method for variable selection in high-dimensional regression with missing data. The case for classification using logistic regression is presented in Section 3. The empirical performances of the proposed methods are evaluated by simulation experiments and a real data application in Sections 4 and 5 respectively. Lastly, concluding remarks are given in Section 6.

2. Variable selection for high-dimensional regression with missing data

We first illustrate our methodology with high-dimensional regression. Suppose observed are p predictors, denoted as $\mathbf{x}_1, \ldots, \mathbf{x}_p$. Let $\mathbf{y} = (y_1, \ldots, y_n)$ be the response vector of n observations, $\mathbf{X}$ be the corresponding $n \times p$ design matrix and $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)^T$ be the regression coefficients. We allow the

* Corresponding author.
E-mail addresses: qigao@ucdavis.edu (Q. Gao), tcmlee@ucdavis.edu (T.C.M. Lee).

http://dx.doi.org/10.1016/j.sigpro.2016.07.014
0165-1684/

possibility of $p \gg n$, the so-called "large p small n" situation. With a linear regression model we have $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I}_n)$ is the vector of random errors and $\mathbf{I}_n$ represents the identity matrix of size n. Since $p \gg n$, a sparsity assumption has to be imposed; i.e., many components of $\boldsymbol{\beta}$ are exactly zero or negligibly small. Let s denote the number of significant predictors, in practice we usually have $s < O(p)$ so that the full model can be well approximated by a much smaller submodel. With this assumption, an important problem is to determine which predictors are significant, and to estimate their corresponding regression coefficients.

This paper considers a more challenging and frequently encountered version of this problem: it allows some entries of $\mathbf{X}$ that are missing completely at random. Our proposed solution to this problem consists of two major stages:

1. Use a procedure to impute the missing entries of $\mathbf{X}$; denote the imputed $\mathbf{X}$ as $\hat{\mathbf{X}}$.
2. Given $\mathbf{y}$ and $\hat{\mathbf{X}}$, apply a procedure to select the significant predictors as well as estimating their coefficients.

Since we would like to allow the case when $p \gg n$ with a large n, the choices of the above procedures are limited. After extensive methodological consideration and numerical investigation, we propose using matrix completion for the first stage and adaptive lasso for the second. A full description of our choices is given below. As to be seen, the overall method is very fast and straightforward to implement. We stress that in our proposal the two stages are *not* executed independently, as the choice of the tuning parameters in Stage 1 requires information from Stage 2, and vice versa.

2.1. Missing value imputation using matrix completion

This subsection discusses our method for imputing the missing entries of $\mathbf{X}$. For clarity, we occasionally use a subscript to denote the size of a matrix; e.g., $\mathbf{X}_{n\times p}$.

Perhaps the most well-known approach for handling missing data problems is the EM-algorithm and its variants. However, these model-based methods are not practical for the present problem as they can be computationally demanding and hard to incorporate large missing fraction, especially under the "large p small n" situation. Therefore, we take a different path and assume that $\mathbf{X}$ has a low rank structure. That is, $\mathbf{X}$ can be well approximated by the product of two matrices $\mathbf{V}$ and $\mathbf{G}$ such that $\mathbf{X}_{n\times p} \approx \mathbf{V}_{n\times r}\mathbf{G}_{r\times p}$ with the rank r of $\mathbf{X}$ satisfies $r \ll \min(n, p)$. In practice, $r < \sqrt{\min(n, p)}$ when $p > \sqrt{n}$. Under this assumption, many fast matrix completion algorithms can be applied to quickly impute the missing entries of $\mathbf{X}$; e.g., see [6,10,12,15,17–19].

In this paper we use the `softImpute-ALS` procedure proposed by [10] which is a relatively new matrix completion algorithm and has the advantage of fast computation compared with other methods. This procedure completes large matrices efficiently by combining two popular methodologies: nuclear-norm-regularized matrix approximation and maximum-margin matrix factorization. Let Ω be the index set of the non-missing entries of $\mathbf{X}$; i.e. $\Omega = \{(i, j)$: the ijth entry X_{ij} is observed$\}$. Also let $P_\Omega(\mathbf{X})$ be the projection of the $n \times p$ matrix which preserves the non-missing entries of $\mathbf{X}$ and replaces the missing entries with 0. Similarly, $P_\Omega^\perp$ denotes the projection onto the complement of Ω. Then `softImpute-ALS` solves the following minimization problem:

$$\underset{\mathbf{A},\mathbf{B}}{\text{minimize}}\left\{ \|P_\Omega(\mathbf{X} - \mathbf{A}\mathbf{B}^T)\|_F^2 + \eta(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2) \right\}, \tag{1}$$

where $\mathbf{A}_{n\times r}$ and $\mathbf{B}_{r\times p}^T$ are each of rank at most $r \le \min(n, p)$, and $\|\cdot\|_F$ is the Frobenius norm. Denote the minimizers as $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$. The imputed $\mathbf{X}$ is then obtained as $\hat{\mathbf{X}} = P_\Omega(\mathbf{X}) + P_\Omega^\perp(\hat{\mathbf{A}}\hat{\mathbf{B}}^T)$. In all our numerical work we use the `R` package `softImpute` to implement this algorithm. A rank r for $\mathbf{X}$ is specified and η is fixed as a small number (0.5) so as not to over-penalize. Next we discuss the choice of r.

Methods are available for selecting the rank r for matrix completion methods, such as the BIC criterion of [18]. However, these methods are suboptimal for our problem as the information in $\mathbf{y}$ is not utilized in the selection. In the context of multivariate response regression models, the Rank Selection Criterion (RSC) is introduced by [2] for selecting the rank of the coefficient matrix estimates. Inspired by this, we propose choosing r as the minimizer of the following modified RSC:

$$\text{RSC}(r) = \|\mathbf{y} - \hat{\mathbf{X}}\hat{\boldsymbol{\beta}}_{\lambda,r}\|_F^2 + \mu r, \tag{2}$$

where $\hat{\boldsymbol{\beta}}_{\lambda,r}$ is an estimate for $\boldsymbol{\beta}$ (to be discussed below). The quantity μ is a tuning parameter and by theoretical consideration the following lower bound is derived by [2]: $\mu^{1/2} > \hat{\sigma}\{\text{ncol}(\hat{\boldsymbol{\beta}})^{1/2} + \text{rank}(\hat{\boldsymbol{\beta}})^{1/2}\}$ where $\hat{\sigma}$ is an estimate for the random noise standard deviation σ which can be estimated by the standard deviation of the residuals $\mathbf{y} - \hat{\mathbf{X}}\hat{\boldsymbol{\beta}}_{\lambda,r}$ and ncol stands for the number of columns. We use the smallest μ that satisfies this bound; i.e., $\mu = 4\hat{\sigma}^2$. We use the RSC in [2] as a starting point to develop a method for choosing r for the problem that we consider, and it turns out this modified RSC possesses very good empirical properties as shown later in numerical experiments. Now we are ready to discuss the problem of selecting significant predictors given an imputed $\hat{\mathbf{X}}$.

2.2. Variable selection with adaptive lasso

As mentioned in the introduction, many methods have been developed for simultaneous variable selection and parameter estimation for the high-dimensional regression problem. Virtually any of these methods could be used here for our problem, but we recommend using the adaptive lasso of [31] for the following reasons. First, its theoretical properties are well studied and it has been shown for example by [11] to perform extremely well for high-dimensional problems and satisfy the oracle properties defined by [7]. Therefore, we believe that it is a reliable option for our high-dimensional variable selection problem. Second, fast algorithms and software exist for computing its solutions; e.g., the *R* package `glmnet` of [20]. And lastly, it can be straightforwardly extended to generalized linear models to handle classification problems.

The adaptive lasso estimate $\hat{\boldsymbol{\beta}}_{\lambda,r}$ for $\boldsymbol{\beta}$ is defined as

$$\hat{\boldsymbol{\beta}}_{\lambda,r} = \underset{\boldsymbol{\beta}}{\text{argmin}}\left(\|\mathbf{y} - \hat{\mathbf{X}}\boldsymbol{\beta}\|_F^2 + \lambda \sum_{j=1}^{p} \hat{w}_j|\beta_j| \right), \tag{3}$$

where λ is a tuning parameter and $\hat{w}_1, \ldots, \hat{w}_p$ are pre-set weights. We follow [31] and use ridge regression to obtain these weights; i.e., for all j we set $\hat{w}_j = 1/|\hat{\beta}_{j,\text{ridge}}|$, where $\hat{\beta}_{j,\text{ridge}}$ is the ridge regression estimate of β_j.

To calculate (3), we need to choose λ, and to do so we use the Extended Bayesian Information Criterion (EBIC) of [3]. In brief EBIC is a modified version of BIC that is tailored to "large p small n" problems. By taking into account the complexity of the enlarged model space in addition to the number of the parameters, EBIC is able to control the false discovery rate. It is also shown to be consistent in [3] under some regularity conditions. For the current problem, EBIC is

$$\text{EBIC}_\gamma(s) = n \log \frac{\text{RSS}}{n} + s \log n + 2\gamma \log \binom{p}{s}, \quad 0 \le \gamma \le 1, \tag{4}$$

where s is the number of variables selected (i.e., number of non-zero entries in $\hat{\boldsymbol{\beta}}_{\lambda,r}$), and RSS is the residual sum of squares $\text{RSS} = \| \mathbf{y} - \hat{\mathbf{X}}\hat{\boldsymbol{\beta}}_{\lambda,r} \|^2$. The tuning parameter γ is pre-specified and it controls the trade-off between Positive Selection Rate (PSR) and False Discovery Rate (FDR). Popular choices are 0.5 and 1 and larger values of γ give tighter control over FDR; e.g., see [3]. Also, $\text{EBIC}_\gamma(s)$ reduces to the classical BIC when $\gamma = 0$.

To sum up, for the current problem of variable selection in high-dimensional regression with missing data, we propose using the matrix completion criterion (1) to impute the missing entries of $\mathbf{X}$, and applying adaptive lasso to the imputed $\hat{\mathbf{X}}$ to estimate $\boldsymbol{\beta}$. In doing so, two parameters need to be selected: the rank r for matrix completion in (1) and the tuning parameter λ for adaptive lasso in (3). These selections are done by minimizing the RSC and the EBIC criteria given, respectively, in (2) and (4). Note that these selections are *not* carried out independently, as the choice of r would affect the choice of λ, and vice versa; the chosen pair (r, λ) should jointly minimize (2). We term the final procedure as `impute.regress` and Algorithm 1 lists its major steps.

Algorithm 1. `impute.regress`.

Description: Perform variable selection for high-dimensional regression with missing values.
Inputs: $\mathbf{y}$ – response vector;
$\mathbf{X}$ – design matrix with missing entries;
max. *rank* – maximum possible rank used for matrix completion (default 50)
Output: $\hat{\boldsymbol{\beta}}_{\lambda,r}$ – estimated $\boldsymbol{\beta}$ (a zero value indicates the corresponding predictor is not selected)
$\hat{\mathbf{X}}$ – imputed $\mathbf{X}$

1: For r in 1 to max. *rank*, perform the following steps:
(a) Obtain $\hat{\mathbf{X}}$ by applying `softImpute-ALS` to $\mathbf{X}$ with rank r.
(b) Apply adaptive lasso (3) to $(\mathbf{y}, \hat{\mathbf{X}})$ to obtain $\hat{\boldsymbol{\beta}}_{\lambda,r}$. Use EBIC (4) to choose the tuning parameter λ and ridge regression to obtain the weights $\hat{w}_j$'s.
(c) With $(\mathbf{y}, \hat{\mathbf{X}}, \hat{\boldsymbol{\beta}}_{\lambda,r})$, calculate $\text{RSC}(r)$ defined by (2).
2: Find the pair $(\hat{\mathbf{X}}, \hat{\boldsymbol{\beta}}_{\lambda,r})$ that gives the smallest value of $\text{RSC}(r)$.
3: Return the pair obtained from the previous step as the outputs of the algorithm.

3. Feature selection for high-dimensional classification with missing data

This section turns to classification using logistic regression. The response now is assumed to follow $y_i \sim \text{Bernoulli}(\pi_i)$ with

$$g(\pi_i) = \log \frac{\pi_i}{(1-\pi_i)} = \sum_{j=1}^{p} x_{ij}\beta_j, \quad i = 1, \ldots, n,$$

where x_{ij} is the jth predictor value of the ith observation; i.e., the ijth element of $\mathbf{X}$. As similar to before, we allow $p \gg n$ and missing entries in $\mathbf{X}$. The methodology introduced in the previous section can be applied similarly to select and estimate the non-zero β_j's.

There are a few modifications to be made. First, the Rank Selection Criterion (2) needs to be changed to tailor for the new random component of the problem:

$$\text{RSC}(r) = \text{deviance} + \mu r = 2 \sum_{i=1}^{n} \left\{ y_i \log\left(\frac{y_i}{\hat{\pi}_i}\right) + (1-y_i)\log\left(\frac{1-y_i}{1-\hat{\pi}_i}\right) \right\} + \mu r, \tag{5}$$

where $\hat{\pi}_i$ is the fitted value of π_i, and can be obtained from $\hat{\pi}_i = \frac{\exp(\mathbf{x}\hat{\boldsymbol{\beta}})}{1+\exp(\mathbf{x}\hat{\boldsymbol{\beta}})}$. Note that the deviance of logistic regression is the counterpart of the residual sum of squares of linear regression. To choose μ we use the same lower bound $\mu^{1/2} > \hat{\sigma}\{\text{ncol}(\hat{\boldsymbol{\beta}})^{1/2} + \text{rank}(\hat{\boldsymbol{\beta}})^{1/2}\}$. Since 0.5 is the maximum standard error for a Bernoulli random variable, we set $\hat{\sigma} = 0.5$ and hence our choice for μ is $\mu = 1$.

Next the definition of the adaptive lasso estimate $\hat{\boldsymbol{\beta}}_{\lambda,r}$ in (3) needs to be modified. Denote the ith element of the vector $\mathbf{u}$ as $\{\mathbf{u}\}_{\{i\}}$. The new definition is

$$\hat{\boldsymbol{\beta}}_{\lambda,r} = \underset{\boldsymbol{\beta}}{\text{argmin}} \left[2 \sum_{i=1}^{n} \log\left\{ 1 + \exp\left(\{\hat{\mathbf{X}}\boldsymbol{\beta}\}_{\{i\}} \right) \right\} - 2\mathbf{y}^T\hat{\mathbf{X}}\boldsymbol{\beta} + \lambda \sum_{j=1}^{p} \hat{w}_j|\beta_j| \right], \tag{6}$$

where the weights $\hat{w}_j$'s are now obtained from logistic ridge regression and λ is chosen with the following modified EBIC:

$$\text{EBIC}_\gamma(s) = 2 \sum_{i=1}^{n} \log\left\{ 1 + \exp\left(\{\hat{\mathbf{X}}\hat{\boldsymbol{\beta}}_{\lambda,r}\}_{\{i\}} \right) \right\} - 2\mathbf{y}^T\hat{\mathbf{X}}\hat{\boldsymbol{\beta}}_{\lambda,r} + s \log n + 2\gamma \log \binom{p}{s}, \quad 0 \le \gamma \le 1. \tag{7}$$

The overall procedure is termed `impute.classi` and is summarized in Algorithm 2. Lastly we note that fast algorithms and software exist for carrying out logistic ridge regression and logistic adaptive lasso; e.g., the *R* packages `ridge` and `glmnet`.

Algorithm 2. `impute.classi`.

Description: Perform feature selection for high-dimensional classification with missing values.
Inputs: $\mathbf{y}$ – response vector of 0 or 1;
$\mathbf{X}$ – design matrix with missing entries;
max. *rank* – maximum possible rank used for matrix completion (default 50)
Output: $\hat{\boldsymbol{\beta}}_{\lambda,r}$ – estimated $\boldsymbol{\beta}$ (a zero value indicates the corresponding predictor is not selected)
$\hat{\mathbf{X}}$ – imputed $\mathbf{X}$

1: For r in 1 to *max. rank*, perform the following steps:
(a) Obtain $\hat{\mathbf{X}}$ by applying `softImpute-ALS` to $\mathbf{X}$ with rank r.
(b) Apply logistic adaptive lasso (6) to $(\mathbf{y}, \hat{\mathbf{X}})$ to obtain $\hat{\boldsymbol{\beta}}_{\lambda,r}$. Use EBIC (7) to choose the tuning parameter λ and logistic ridge regression to obtain the weights $\hat{w}_j$'s.
(c) With $(\mathbf{y}, \hat{\mathbf{X}}, \hat{\boldsymbol{\beta}}_{\lambda,r})$, calculate $\text{RSC}(r)$ defined by (5).
2: Find the pair $(\hat{\mathbf{X}}, \hat{\boldsymbol{\beta}}_{\lambda,r})$ that gives the smallest value of $\text{RSC}(r)$.
3: Return the pair obtained from the previous step as the outputs of the algorithm.

4. Numerical experiments

This section reports results from a series of numerical experiments conducted to study the practical performances of the proposed algorithms `impute.regress` and `impute.classi`. To provide informative and concise quantitative measures for the quality of variable selection, we follow [5] and use sensitivity of selection (SEN) and specificity of selection (SPE), defined respectively as

$$\text{SEN} = \frac{\text{number of selected significant variables}}{\text{number of true significant variables}} \quad \text{and} \quad \text{SPE} = \frac{\text{number of removed insignificant variables}}{\text{number of true insignificant variables}}.$$

Note that SEN can be interpreted as the ability of detecting true significant variables while SPE as the ability of removing those insignificant ones.

Moreover, for the regression setting we have additional test data to calculate empirical averaged prediction error, defined as $\text{PE} = \frac{1}{n_{\text{test}}} \| \mathbf{y}_{\text{test}} - \mathbf{X}_{\text{test}}\hat{\boldsymbol{\beta}} \|^2$, where n_{test} is the number of observations in the test data. A big design matrix is simulated for each setting as well as the corresponding response and some of the observations are taken as test data. Therefore, each set of test data is simulated under the same setting as the training data but with no entries removed as missing values. For the classification setting we also provide mis-classification rate (MCR), defined as the percentage of mis-classified $\mathbf{y}_{\text{test}}$ in our test data.

4.1. Linear regression

We test the performance of `impute.regress` with three experimental settings.

For the first two settings, the complete design matrix $\mathbf{X}$ was simulated as a low rank matrix with additive random noise so that the low rank assumption behind matrix completion approximately holds. The response $\mathbf{y}$ was then obtained from a linear regression model specified below, with different levels of noise variance σ^2. The complete $\mathbf{X}$ remained fixed for each repetition once it was generated. A fraction (denoted as δ below) of the entries from $\mathbf{X}$ is removed randomly as missing values. We report SEN, SPE and PE. As a benchmark comparison, we also report the corresponding results obtained from the complete $\mathbf{X}$ (i.e., no missing values); in this case no matrix completion was performed and the variables were selected using adaptive lasso with EBIC choice of the tuning parameter λ.

Setting 1: Here $n=1000$, $n_{\text{test}} = 200$ and $p=20$. The rank r is set to be 5 which means that $\mathbf{X}$ is approximately the product of two matrices both of rank 5. The first ten entries of $\boldsymbol{\beta}$ were $\boldsymbol{\beta} = (0, 1.5, 0, 0, 1, 0, 0.8, 0, 1.2, 0.7)$ and the rest of the entries were set to be zero. Therefore there were five nonzero entries and thus five significant variables. As in this case $p \ll n$, the original BIC is used; i.e., $\text{EBIC}_\gamma(s)$ in (4) with $\gamma = 0$. The averaged SEN, SPE and PE values based on 500 simulations are summarized in Table 1. The standard errors of SEN and SPE are also reported in parentheses.

Setting 2: Here $n=1000$, $n_{\text{test}} = 200$ and $p=2000$. The rank r is set to be 10 which means that $\mathbf{X}$ is approximately the product of two matrices both of rank 10. The first 20 entries of $\boldsymbol{\beta}$ were $\boldsymbol{\beta} = (0, 1.5, 0, 0, 1, 0, 0.8, 0, 1.2, 0.7, 0, 1.5, 0, 0, 1, 0, 0.8, 0, 1.2, 0.7)$ and the rest of the entries are set to be zero. Therefore there are 10 nonzero entries and thus 10 significant variables. In this case $p \gg n$ and the original BIC tends to be too liberal in variable selection. Therefore EBIC is used and γ is chosen by $\gamma = 1 - 1/(2\kappa)$ with κ satisfying $p = n^\kappa$, as suggested in [3]. In this case $\gamma \approx 0.44$. The results based on 500 simulations are summarized in Table 2.

Setting 3: This simulation set-up was used in [3] with $n=200$, $p=1000$ and the first six entries of $\boldsymbol{\beta}$ were $\boldsymbol{\beta} = (0, 1.0, 0.7, 0.5, 0.3, 0.2)$ while the rest are zero. The complete $\mathbf{X}$ was generated as follows. First the predictors were divided into two equal halves with each half consists of 10 groups of size 50. Then the first 10 groups were generated from $N(0, 1)$ with the covariance structure $\text{cov}(x_{ij}, x_{ik}) = \rho^{|k-j|}$ for all pairs of k and j. Lastly, the other 10 groups were scaled values from a distribution over $(-2, 0, 4)$ with corresponding probabilities $(0.5, 0.25, 0.25)$.

Table 1
Simulation results for Setting 1 in linear regression. In below δ is the missing fraction, σ is the standard deviation of noise, SEN is the sensitivity of selection, SPE is the specificity of selection, and PE is the prediction error. Numbers in parentheses are standard errors.

Design of x	$p=20$, $n=1000$, $n_{\text{test}} = 200$				
	δ	σ	SEN	SPE	PE
Incomplete $\mathbf{X}$	0.25	0.75	99.68% (0.10%)	92.51% (0.33%)	0.578
	0.25	1	99.51% (0.10%)	90.82% (0.35%)	1.020
	0.25	1.5	98.19% (0.20%)	87.63% (0.41%)	2.289
	0.5	0.75	94.22% (0.34%)	82.30% (0.44%)	0.674
	0.75	0.75	50.00% (0.67%)	66.68% (0.50%)	11.256
Complete $\mathbf{X}$	0	0.75	100% (0%)	98.89% (0.09%)	0.567
	0	1	100% (0%)	97.23% (0.14%)	1.005
	0	1.5	99.80% (0.06%)	95.34% (0.78%)	2.265

Table 2
Similar to Table 1 but for Setting 2 of linear regression.

Design of x	$p=2000$, $n=1000$, $n_{\text{test}} = 200$				
	δ	σ	SEN	SPE	PE
Incomplete $\mathbf{X}$	0.25	0.75	96.28% (0.31%)	98.35% (0.17%)	0.691
	0.25	1	87.85% (0.51%)	98.24% (0.12%)	1.252
	0.25	1.5	57.12% (0.56%)	99.06% (0.10%)	3.088
	0.5	0.75	77.78% (0.60%)	98.55% (0.16%)	1.073
	0.75	0.75	41.15% (0.48%)	99.23% (0.04%)	1.734
Complete $\mathbf{X}$	0	0.75	99.90% (0.03%)	99.64% (0.01%)	0.597
	0	1	97.40% (0.21%)	99.37% (0.13%)	1.089
	0	1.5	61.97% (0.56%)	99.60% (0.01%)	2.565

As in [3], two values of $\rho = (0.2, 0.4)$ and two values of $\gamma = (0.5, 1)$ for $\text{EBIC}_\gamma(s)$ were used. The missing fraction δ was 0.25. The results are summarized in Table 3.

Setting 4: We compared our method with [24] and model 6 in their paper is used here, i.e., $n = 100$, $p = 50$ and $p=200$; the covariance matrix of the predictors $\Sigma_{jj'} = 0.8 \times I_{(j,j' \le 9)}$ for $j \neq j'$ and $\Sigma_{jj} = 1$; $\beta_j = 2$ for $j = 1, \ldots, 8$ and zero elsewhere; $\sigma = 0.5$. Since their method is EM-based and computationally expensive, we did not use larger values of p and n as in previous settings. Moreover, their method is much slower when the missing fraction is increased, so we also limit the missing fraction in this setting. The average running time is also reported and the results based on 50 simulations can be found in Table 4.

Setting 5: Here we have similar dimensions as in setting 2 where $n=1000$ and $p=2000$. We tried a few combinations of true approximate rank of $\mathbf{X}$ and the number of significant variables

Table 3
Similar to Table 1 but for Setting 3 of linear regression. In below ρ controls the correlation between the predictors and γ is a pre-specified constant for $\text{EBIC}_\gamma(s)$.

Design of x	$p=1000$, $n=200$, $n_{\text{test}} = 50$, $\delta = 0.25$				
	ρ	γ	SEN	SPE	PE
Incomplete $\mathbf{X}$	0.2	0.5	76.81% (1.32%)	94.51% (0.59%)	1.768
	0.2	1	68.16% (1.38%)	98.80% (0.38%)	1.445
	0.4	0.5	80.12% (1.11%)	91.17% (0.78%)	1.268
	0.4	1	77.07% (1.11%)	98.66% (0.28%)	1.303
Complete $\mathbf{X}$	0.2	0.5	78.18% (1.00%)	99.86% (0.12%)	1.130
	0.2	1	73.16% (0.96%)	99.90% (0.10%)	1.168
	0.4	0.5	84.32% (0.81%)	99.82% (0.15%)	1.079
	0.4	1	80.88% (1.38%)	99.99% (0.01%)	1.091

Table 4
Similar to Table 1 but for Setting 4 of linear regression (model 6 in [24]).

Method	$n=100$, $\sigma=0.5$				
	p	δ	SEN	SPE	Running time
impute.regress	50	0.1	100% (0%)	98.48% (0.76%)	20 s
	50	0.2	99.50% (0.35%)	88.62% (2.81%)	28 s
	50	0.3	96.00% (0.97%)	87.48% (1.85%)	29 s
	200	0.1	100% (0%)	98.82% (0.52%)	4 min
Method in [24]	50	0.1	100% (0%)	81.29% (1.69%)	38 min
	50	0.2	100% (0%)	74.19% (1.78%)	3.5 h
	50	0.3	100% (0%)	79.19% (1.67%)	13.8 h
	200	0.1	100% (0%)	91.75% (0.74%)	3 days

(denoted by s). The first 10 entries of $\boldsymbol{\beta}$ were $\boldsymbol{\beta}_1^* = (0, 1.5, 0, 0, 1, 0, 0.8, 0, 1.2, 0.7)$ and the rest of the entries are set to be zero when $s=5$. For $s=10$ and $s=20$, the nonzero entries are $\boldsymbol{\beta}_2^* = (\boldsymbol{\beta}_1^*, \boldsymbol{\beta}_1^*)$ and $\boldsymbol{\beta}_3^* = (\boldsymbol{\beta}_2^*, \boldsymbol{\beta}_2^*)$ respectively. SEN, SPE and the average $\hat{r}$ are reported for each combination and the results based on 200 simulations can be found in Table 5. Note that $\mathbf{X}$ is approximately of low rank in our simulations so the rank might not be recovered perfectly, but among all simulations, the mostly frequently occurred $\hat{r}$ is always r.

Empirical conclusions for linear regression: From Tables 1 to 3 some obvious conclusions can be drawn: the performance of impute.regress is better with smaller missing fraction δ and noise standard deviation σ. With 25% missing data, the SEN, SPE and PE obtained from impute.regress are actually reasonably close to their counterparts obtained with no missing data. Even with 50% missing data, impute.regress did produce quite satisfactory results.

From Table 4 we can see that our method is much faster than the method proposed in [24] with competitive performance. Actually our SPE is significantly higher across all settings while our SEN is slightly lower in two of them. The increase in speed is crucial for analyzing high-dimensional data, especially with high missing fraction. Table 5 shows that the average estimated ranks are reasonably close to the true ranks, which confirms the good empirical properties of our RSC proposed in (2).

Table 5
Similar to Table 2 but for Setting 5 of linear regression. s stands for the number of significant variables.

$n=1000$, $p=2000$, $\sigma=0.75$, $\delta=0.25$				
r	s	SEN	SPE	Avg ($\hat{r}$)
5	5	96.90% (0.71%)	99.16% (0.23%)	5.52
5	10	92.61% (0.95%)	98.77% (0.21%)	6.45
5	20	90.35% (0.84%)	98.45% (0.20%)	7.38
10	5	97.36% (0.50%)	99.01% (0.21%)	10.72
10	10	96.95% (0.57%)	98.10% (0.46%)	11.59
10	20	93.16% (0.83%)	98.39% (0.20%)	12.92
20	5	99.90% (0.10%)	99.88% (0.02%)	20.13
20	10	99.80% (0.12%)	99.05% (0.06%)	21.19
20	20	96.05% (0.68%)	96.71% (0.61%)	23.8

Table 6
Simulation results for Setting 1 in logistic regression. In below δ is missing fraction, SEN is sensitivity of selection, SPE is specificity of selection, and MCR is mis-classification rate. Numbers in parentheses are standard errors.

	$p=20$, $n=1000$, $n_{\text{test}}=200$			
	δ	SEN	SPE	MCR
Incomplete $\mathbf{X}$	0.25	94.47% (0.39%)	75.38% (0.39%)	3.70%
	0.5	77.08% (0.49%)	75.58% (0.35%)	4.11%
	0.75	68.33% (0.50%)	72.18% (0.50%)	7.96%
Complete $\mathbf{X}$	0	98.33% (0.23%)	82.80% (0.17%)	3.50%

4.2. Classification

For classification problems, the proposed impute.classi was tested with two settings.

Setting 1: Here $n=1000$, $n_{\text{test}}=200$ and $p=20$. The complete $\mathbf{X}$ was first generated as a low rank matrix of rank 5 plus additive noise, and $\mathbf{y}$ was then simulated as Bernoulli random variables as described in Section 3. The first seven entries of $\boldsymbol{\beta}$ were $\boldsymbol{\beta}=(0, 3, 0, 1.5, 0, 0, 7)$ and the remaining thirteen entries were zero. The original BIC is used here as $p \ll n$, and the results based on 500 simulations are reported in Table 6.

Setting 2: This simulation set-up is adopted from [4] to mimic a case-control study with the same number of cases and controls. The responses $\mathbf{y}$ represent disease status while $\mathbf{X}$ are single nucleotide polymorphisms (SNPs) in the human genome; see [4] for details. There were 500 observations for both cases and controls; i.e., $n=1000$, and four values of p were considered: $p=(500, 1000, 1500, 2000)$. Also, two choices of $\boldsymbol{\beta}$ were used: $\boldsymbol{\beta}^{(1)}=(0.5, 0.6, 0.7, 0, \ldots, 0)$ and $\boldsymbol{\beta}^{(2)}=(0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0, \ldots, 0)$. The missing fraction δ was 0.25 and we used $\gamma=0.5$ for EBIC. The results based on 500 simulations are presented in Table 7. We list the results using impute.classi with and without missing values, and those using the method of [4] with the same sets of complete data.

Empirical conclusions for classification: Table 6 shows that impute.classi worked quite well for Setting 1; e.g., with 50% missing data the MCR is 4.11%, comparing to 3.50% obtained with no missing data. For Setting 2, impute.classi also worked well, as both SENs and SPEs with missing data are quite close to their complete data counterparts.

5. Real data analysis

5.1. Linear regression

We firstly applied impute.regress to the Affymetrix GeneChip Rat Genome 230 2.0 Array data set reported in [22], where 31,042 probe sets were measured from eye tissue of 120 rats subjects. As in [11], we are interested in using regression analysis to find the probes whose expression levels are most related to gene TRIM32 which is found to cause Bardet–Biedl syndrome. We also used the screening method introduced in [11] and worked with 200 covariates which have the largest marginal correlation coefficient with TRIM32. Therefore here $n=120$, $p=200$ and we have a large-p-small-n scenario. Cross-validation was used to estimate the predictive mean squared errors (MSEs) of our method, to be compared with those in [11]. Each time two-thirds of the observations were selected randomly as a training set and 10% of the complete gene expression matrix were deleted completely at random following [24]. The rest of the observations were used as a testing set to calculate MSE. Among 300 random partitions, the median MSE is 0.007 and the median number of nonzero variables

Table 7
Similar to Table 6 but for Setting 2 of logistic regression. The missing percentage is 25%.

$n=1000$				
X	p	β_0	SEN	SPE
With NAs	500	$\beta_0^{(1)}$	83.33% (0.93%)	99.85% (0.007%)
	500	$\beta_0^{(2)}$	63.07% (0.54%)	99.74% (0.010%)
	1000	$\beta_0^{(1)}$	79.07% (0.71%)	99.92% (0.003%)
	1000	$\beta_0^{(2)}$	59.86% (0.36%)	99.87% (0.004%)
	1500	$\beta_0^{(1)}$	78.83% (0.60%)	99.96% (0.001%)
	1500	$\beta_0^{(2)}$	59.74% (0.31%)	99.92% (0.002%)
	2000	$\beta_0^{(1)}$	74.63% (0.53%)	99.97% (0.001%)
	2000	$\beta_0^{(2)}$	57.72% (0.25%)	99.94% (0.001%)
Without NAs	500	$\beta_0^{(1)}$	89.00% (0.77%)	99.91% (0.006%)
	500	$\beta_0^{(2)}$	65.95% (0.56%)	99.91% (0.007%)
	1000	$\beta_0^{(1)}$	87.24% (0.58%)	99.95% (0.002%)
	1000	$\beta_0^{(2)}$	63.78% (0.39%)	99.95% (0.002%)
	1500	$\beta_0^{(1)}$	86.63% (0.50%)	99.97% (0.001%)
	1500	$\beta_0^{(2)}$	62.32% (0.30%)	99.97% (0.001%)
	2000	$\beta_0^{(1)}$	84.00% (0.46%)	99.98% (0.001%)
	2000	$\beta_0^{(2)}$	61.22% (0.24%)	99.98% (0.001%)
Without NAs (from [4])	500	$\beta_0^{(1)}$	90.00%	99.97%
	500	$\beta_0^{(2)}$	65.30%	99.97%
	1000	$\beta_0^{(1)}$	90.00%	99.96%
	1000	$\beta_0^{(2)}$	64.68%	99.96%
	1500	$\beta_0^{(1)}$	88.53%	99.96%
	1500	$\beta_0^{(2)}$	63.33%	99.96%
	2000	$\beta_0^{(1)}$	86.53%	99.94%
	2000	$\beta_0^{(2)}$	63.46%	99.95%

is 8, while those numbers are 0.005 and 17 in [11]. Note that [11] used cross validation on the training set to select the tuning parameter in adaptive lasso while we used EBIC with $\gamma = 0.5$ as the criterion. Therefore our method tends to select fewer variables to control the false discovery rate. Considering this, our method performs satisfactorily in terms of prediction accuracy with missing data.

5.2. Classification

We also applied `impute.classi` to the Pima Indian diabetes data set contributed by [23] which is downloadable from the UCI Machine Learning repository. The original data set contains eight physiological measurements and medical test results of 768 female patients who were at least 21 years old with Pima Indian heritage. Seven of these attributes are listed in the left column of Table 8 and are used to predict whether the patients would test positive for diabetes. The responses are their true test results (positive or negative) which are also recorded in the data set. This data set has been analyzed by different researchers such as [9,14,21]. However, because zero values are physically impossible for some attributes and thus generally considered as missing values, only 532 complete records were used in most of the literature; i.e., about 31% of the records are incomplete. Since `impute.classi` is applicable to data sets with missing values, we included all the records in our analysis (however, following the experimental setup in [14], the eighth variable serum insulin is still removed since zero values prevail).

Table 8
Variable selection results for Pima Indian diabetes data set.

Variable	Percentage of times being selected
Number of times pregnant	29
Plasma glucose concentration	100
Diastolic blood pressure	27
Triceps skin fold thickness	1
Body mass index (BMI)	100
Diabetes pedigree function	100
Age	100

Ten-fold cross validation was used to estimate the mis-classification rate (MCR) of `impute.classi` in the following manner. The whole data set was randomly divided into 10 roughly equal sub-groups using stratified sampling. The first sub-group was set to be the test data while the rest were used as training data to obtain a fitted model. Then the second sub-group was set as the test data and the remaining as training data. The same process continued for the third sub-group, fourth, till the tenth. Therefore 10 fitted models were obtained with 10 estimates of MCR. This whole ten-fold cross validation process was repeated 10 times, resulting in 100 fitted models and testing results. The averaged MCR based on these 100 results is 21.65% with a standard error of 0.034%. Given the presence of missing data, this is very competitive when comparing with the 33 methods discussed in [14] where the MCRs are ranging from 22% to 31%.

Lastly, we computed the percentage of times a variable was selected by our method in the 100 fitted models. The results are presented in Table 8. One can see that the four variables plasma glucose concentration (x_2), BMI (x_5), diabetes pedigree function (x_6) and Age (x_7) were selected all the times, providing a strong indication that these four physiological measurements are highly related to diabetes. This is in fact consistent with many scientific studies about diabetes.

6. Concluding remarks

In this paper a general approach is proposed for high-dimensional variable selection with a relatively large fraction of missing data. In particular two algorithms `impute.regress` and `impute.classi` are developed, respectively, for the regression and classification settings. These two algorithms are based on the matrix completion procedure `softImpute-ALS` and the variable selection method adaptive lasso, and produced very satisfactorily results in a sequence of numerical experiments. Lastly, it worth mentioning that the proposed approach allows "easy upgrade" of the algorithms if improved matrix completion and/or variable selection techniques are invented; e.g., one could simply replace `softImpute-ALS` with a superior matrix completion procedure.

Acknowledgment

The work of Lee was partially supported by the National Science Foundation under grants 1512945 and 1513484.

References

[1] J. Bradic, J. Fan, W. Wang, Penalized composite quasi-likelihood for ultrahigh dimensional variable selection, J. R. Stat. Soc.: Ser. B 73 (2011) 325–349.
[2] F. Bunea, Y. She, M.H. Wegkamp, Optimal selection of reduced rank estimators of high-dimensional matrices, Ann. Stat. 39 (2) (2011) 1282–1309.

[3] J. Chen, Z. Chen, Extended Bayesian information criteria for model selection with large model spaces, Biometrika 95 (3) (2008) 759–771.
[4] J. Chen, Z. Chen, Extended Bayesian information criteria for small-n-large-p sparse GLM, Stat. Sin. 22 (2012) 555–574.
[5] Q. Chen, S. Wang, Variable selection for multiply-imputed data with application to dioxin exposure study, Stat. Med. 32 (2013) 3646–3659.
[6] W. Dai, O. Milenkovis, E. Kerman, Subspace evolution and transfer (set) for low-rank matrix completion, IEEE Trans. Signal Process. 59 (7) (2011) 3120–3132.
[7] J. Fan, H. Peng, Nonconcave penalized likelihood with a diverging number of parameters, Ann. Stat. 32 (3) (2004) 928–961.
[8] J. Fan, R. Samworth, Y. Wu, Ultrahigh dimensional feature selection: beyond the linear model, J. Mach. Learn. Res. 10 (2009) 2013–2038.
[9] E. Gurbuz, E. Kilic, Diagnosis of diabetes by using adaptive svm and feature selection, in: 2011 IEEE 19th Conference on Signal Processing and Communications Applications (SIU), 2011, pages 42–45.
[10] T. Hastie, R. Mazumder, J. Lee, R. Zadeh, Matrix completion and low-rank SVD via fast alternating least squares, J. Mach. Learn. Res. (2015), to appear.
[11] J. Huang, S. Ma, C.-H. Zhang, Adaptive lasso for sparse high-dimensional regression models, Stat. Sin. 18 (2008) 1603–1618.
[12] R.H. Keshavan, A. Montanari, S. Oh, Matrix completion from noisy entries, J. Mach. Learn. Res. 11 (1) (2010) 2057–2078.
[13] Z. Lei, R. Li, X.S. Ni, X. Huo, High-dimensional semi-supervised learning via a fusion-refinement procedure, Signal Process. 114 (2015) 171–182.
[14] T.-S. Lim, W.-Y. Loh, Y.-S. Shih, A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms, Mach. Learn. 40 (2000) 203–229.
[15] X. Lin, G. Wei, Accelerated reweighted nuclear norm minimization algorithm for low rank matrix recovery, Signal Process. 114 (2015) 24–33.
[16] P.-L. Loh, M.J. Wainwright, High-dimensional regression with noisy and missing data: provable guarantees with nonconvexity, Ann. Stat. 40 (3) (2012) 1637–1664.
[17] A. Majumdar, R.K. Ward, Some empirical advances in matrix completion, Signal Process. 91 (2011) 1334–1338.
[18] G. Marjanovic, V. Solo, On l_q optimization and matrix completion, IEEE Trans. Signal Process. 60 (11) (2012) 5714–5724.
[19] R. Mazumder, T. Hastie, R. Tibshirani, Spectral regularization algorithms for learning large incomplete matrices, J. Mach. Learn. Res. 11 (2010) 2287–2322.
[20] M.Y. Park, T. Hastie, L1 regularization path algorithm for generalized linear models, J. R. Stat. Soc.: Ser. B 69 (2007) 659–677.
[21] K. Polat, S. Gunes, An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease, Digit. Signal Process. 17 (July) (2007) 702–710.
[22] T.E. Sheetz, K.-Y.A. Kim, R.E. Swiderski, A.R. Philp, Terry A. Braun, K.L. Knudtson, A.M. Dorrance, G.F. DiBona, J. Huang, T.L. Casavant, V.C. Sheffield, E.M. Stone, Regulation of gene expression in the mammalian eye and its relevance to eye disease, Proc. Natl. Acad. Sci. USA 103(39) (2009) 14429–14434.
[23] V. Sigillito, UCI Machine Learning Repository, 2015.
[24] N. Stadler, P. Buhlmann, Missing values: sparse inverse covariance estimation and an extension to sparse regression, Stat. Comput. 22 (2012) 219–235.
[25] L. Stankovic, S. Stankovic, M. Amin, Missing samples analysis in signals for applications to L-estimation and compressive sensing, Signal Process. 94 (2014) 401–408.
[26] A. Torokhti, P. Howlett, H. Laga, Estimation of stochastic signals under partially missing information, Signal Process. 111 (2015) 199–209.
[27] J. Wood, I.R. White, P. Cutler, A likelihood-based approach to defining statistical significance in proteomic analysis where missing data cannot be disregarded, Signal Process. 84 (2004) 1777–1788.
[28] L. Xue, A. Qu, Variable selection in high-dimensional varying-coefficient models with global optimality, J. Mach. Learn. Res. 13 (2012) 1973–1998.
[29] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: Proceedings of the Twentieth International Conference on Machine Learning, 2003, pp. 856–863, 2003.
[30] S.-Z. Yu, H. Kobayashi, A hidden semi-Markov model with missing data and multiple observation sequences for mobility tracking, Signal Process. 83 (2003) 235–250.
[31] H. Zou, The adaptive lasso and its oracle properties, J. Am. Stat. Assoc. 101 (476) (2006) 1418–1429.