Statistics and Its Interface Volume 0 (2010) 1–8

# A Penalized Maximum Likelihood Approach to Sparse Factor Analysis

Jang Choi, Hui Zou[*] and Gary Oehlert

Factor analysis is a popular multivariate analysis method which is used to describe observed variables as linear combinations of hidden factors. In applications one usually needs to rotate the estimated factor loading matrix in order to obtain a more understandable model. In this article, an $\ell_1$ penalization method is introduced for performing sparse factor analysis in which factor loadings naturally adopt a sparse representation, greatly facilitating the interpretation of the fitted factor model. A generalized expectation-maximization algorithm is developed for computing the $\ell_1$ penalized estimator. Efficacy of the proposed methodology and algorithm is demonstrated by simulated and real data.

AMS 2000 subject classifications: Primary 62H25; secondary 62J07.

Keywords and phrases: Adaptive Lasso, EM algorithm, Factor analysis, Lasso, Sparse factor loadings.

## 1. INTRODUCTION

Factor analysis [7] models the observed multivariate random variables as linear combinations of some unobserved (hidden) factors plus error terms. Factor analysis was first introduced by Charles Spearman in 1904 [12] to support his psychological theory of intelligence. Since then, factor analysis has been widely used in many research fields such as psychometrics, behavioral sciences, social sciences, political sciences, marketing, economics, finance, and so on.

Suppose we have $n$ independent and identically distributed (i.i.d.) random vectors in $\mathbb{R}^p$: $\{Y_1, \ldots, Y_n\}$. Without loss of generality, assume the mean of $Y_i$ is zero and its covariance is $\mathbf{\Sigma}$ . The factor model is represented by

$$Y_i = \boldsymbol{\beta}^T X_i + e_i, \tag{1}$$

where $X_i$ is an *unobserved* random vector of length $q$, $\boldsymbol{\beta}$ is a $q \times p$ matrix and $e_i$ represents a $p$-dimension random error vector whose mean is zero and covariance is a diagonal matrix denoted by $\boldsymbol{\tau}^2 = \text{diag}(\tau_1^2, \ldots, \tau_p^2)$. $\boldsymbol{\tau}^2$ is called the uniqueness matrix. It is assumed that $X_i$ has zero mean and covariance $\boldsymbol{I}_q$. As a consequence, the covariance of $Y_i$ can be expressed as $\mathbf{\Sigma}_Y = \boldsymbol{\beta}^T\boldsymbol{\beta} + \boldsymbol{\tau}^2$. In a matrix form we write the model as

$$\mathbf{Y}_{n\times p} = \mathbf{X}_{n\times q}\boldsymbol{\beta}_{q\times p} + \boldsymbol{\epsilon}_{n\times p} \tag{2}$$

where the $i$-th rows of $\mathbf{Y}$, $\mathbf{X}$ and $\boldsymbol{\epsilon}$ are $Y_i$, $X_i$ and $e_i$, respectively. In factor analysis, $\mathbf{X}$ is called the factor score matrix and $\boldsymbol{\beta}$ is called the factor loading matrix. For statistical inference, it is usually assumed that the hidden factors are normal distributed and hence $Y_i$s are i.i.d. $N(0, \boldsymbol{\beta}^T\boldsymbol{\beta} + \boldsymbol{\tau}^2)$. The maximum likelihood estimation can be carried out by using the Expectation-Maximization algorithm [11].

The factor model (2) is invariant under an orthogonal rotation, so is the maximum likelihood estimator. This property makes it possible to rotate the estimated factor loading matrix such that the rotated loading matrix exhibits some interesting structure/pattern that can help interpret the fitted factor model. In fact, rotation is often necessary in real applications of factor analysis when the number of factors is not small. The most common rotation technique is *varimax* that aims to yield either large or small loadings. Often, small loadings are further truncated at some threshold (e.g. 0.01), for zero loadings greatly enhance the interpretability. We can understand the idea behind varimax rotation as follows. Suppose that the factor model can be represented by some sparse $\boldsymbol{\beta}$ matrix that makes the factor model easy to interpret. The MLE $\widehat{\boldsymbol{\beta}}$ is an estimator of $U\boldsymbol{\beta}$ with $U$ being an unknown orthonormal matrix. Varimax aims to find $U^T$ such that hopefully we can recover the sparse $\boldsymbol{\beta}$ matrix by $U^T\widehat{\boldsymbol{\beta}}$, the varimax rotated loading matrix.

In this article we introduce a new approach to sparse factor analysis by taking advantage of sparse penalization methods. In recent years penalization methods have been explored in various sparse estimation and modeling problems. The $\ell_1$ penalization (a.k.a. the lasso) [13] is one of the most popular sparse learning techniques. We propose to fit an $\ell_1$ penalized factor model with an $\ell_1$ penalty imposed on the factor loadings. Due to the sparse shrinkage property of the $\ell_1$ penalty, some factor loadings are estimated by exact zero when the penalization parameter is properly chosen. Besides sparsity, the $\ell_1$ penalization also brings a regularization effect, producing a more accurate model. We also consider using the adaptively weighted $\ell_1$ penalty (a.k.a. the adaptive lasso) [14] to further improve the $\ell_1$ penalized estimator.

arXiv: math/0000.0000

[*] Corresponding author. Zou was partially supported by NSF grant DMS-0846068.

The rest of this article is organized as follows. In Section 2 we present the $\ell_1$ penalized factor analysis method. In Section 3 we develop a generalized Expectation-Maximization (GEM) algorithm to compute the $\ell_1$ and adaptive $\ell_1$ penalized estimators. Numerical examples are presented in Section 4.

## 2. $\ell_1$-PENALIZED FACTOR ANALYSIS

In this section we define the $\ell_1$ penalized factor analysis. Under the normality assumption, the log-likelihood can be written as

(3)
$$LL(\boldsymbol{\tau}^2,\boldsymbol{\beta}) = -\frac{n}{2}\log det(\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta}) - \frac{1}{2}\sum_{i=1}^{n} Y_i^T(\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta})^{-1}Y_i,$$

or equivalently

$$(4)\qquad LL(\boldsymbol{\tau}^2,\boldsymbol{\beta}) = -\frac{n}{2}\left(\log det(\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta}) + tr\left((\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta})^{-1}\boldsymbol{\Sigma}^s\right)\right),$$

where $\boldsymbol{\Sigma}^s = \frac{1}{n}Y_iY_i^T$ is the sample covariance matrix of $Y$. In the above equations we have assumed the mean of $Y_i$ is zero which is done in practice by centering the data matrix. The classical factor analysis uses the maximum likelihood estimator given by

$$(5)\quad \arg\min\left\{\log det(\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta}) + tr\left((\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta})^{-1}\boldsymbol{\Sigma}^s\right)\right\}.$$

It is important to point out that the normal assumption is not critical in (3)–(5). Even when $Y_i$s are non normal, we can still interpret $LL(\boldsymbol{\tau}^2,\boldsymbol{\beta})$ as the log likelihood function. Generally speaking, the objective function in (5) is equivalent to the Kullback-Leibler loss between $\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta}$ and the sample covariance matrix $\boldsymbol{\Sigma}^s$. Without causing any confusion, from now on we still call $LL(\boldsymbol{\tau}^2,\boldsymbol{\beta})$ the log-likelihood.

Interpretability of the factor model becomes very important in applications when the number of factors is not too small. In the classical factor analysis, rotation techniques are often used to obtain more understandable factor loadings. Factor analysis is closely related to principal component analysis in the sense that both methods try to explain the variability among correlated variables by several factors/components. The $\ell_1$ penalization idea has been successfully used to develop sparse principal component analysis [15]. We use the sparse penalization idea to develop sparse factor analysis.

Consider the penalized log-likelihood defined by

$$(6)\qquad PLL(\boldsymbol{\tau}^2,\boldsymbol{\beta}) = -\frac{n}{2}\log det(\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta}) - \frac{n}{2}tr[\boldsymbol{\Sigma}^s(\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta})^{-1}] - \frac{1}{2}\sum_{l=1}^{q}\sum_{j=1}^{p}P_\lambda(|\beta_{lj}|)$$

where $P_\lambda(\cdot)$ is a non-negative penalty function. In recent literature there has been a lot of work on the use of sparsity-inducing penalty functions in various penalized models. The reference list is too long to be listed here. The readers are referred to two good review papers [4, 8]. In this work we use $P_\lambda(|\beta_{lj}|) = \lambda|\beta_{lj}|$ which is the lasso penalty [13]. The lasso estimator, denoted by $(\widehat{\boldsymbol{\tau}^2},\widehat{\boldsymbol{\beta}})$, is then defined as $\arg\max PLL(\boldsymbol{\tau}^2,\boldsymbol{\beta})$, i.e.,

$$(7)\quad (\widehat{\boldsymbol{\tau}^2},\widehat{\boldsymbol{\beta}}) = \arg\min \quad \log det(\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta}) + tr[\boldsymbol{\Sigma}^s(\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta})^{-1}] + \frac{\lambda}{2n}\sum_{l=1}^{q}\sum_{j=1}^{p}|\beta_{lj}|.$$

Note that the $\ell_1$ penalty is not invariant under orthogonal transformation. Therefore, the lasso estimator is no longer rotation invariant.

It has been shown in [14] that the adaptively weighted lasso penalty can achieve better prediction and sparsity trade-off than the lasso and the adaptive lasso estimator enjoys the oracle properties using the language of [5]. In this work we also consider the adaptive lasso (ALasso) estimator in which the adaptive weights are computed from the lasso estimator. The ALasso estimator is computed by the following two-step procedure:

1. Compute the lasso estimator in (7).
2. If $\hat{\beta}_{lj} = 0$ let $\hat{w}_{lj} = \infty$, otherwise $\hat{w}_{lj} = \frac{1}{|\hat{\beta}_{lj}|}$. Then compute the adaptive lasso penalized estimator

$$\arg\min \quad \log det(\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta}) + tr[\boldsymbol{\Sigma}^s(\boldsymbol{\tau}^2+\boldsymbol{\beta}^T\boldsymbol{\beta})^{-1}] + \frac{\lambda^*}{2n}\sum_{l=1}^{q}\sum_{j=1}^{p}\hat{w}_{lj}|\beta_{lj}|.$$

In principle we can also use other penalty functions such as the SCAD [5] to derive sparse factor analysis. In the next section we develop a generalized expectation-maximization (GEM) algorithm for maximizing the objective function in (6) with a general penalty function. An advantage of using the $\ell_1$ penalty is that we do not need to consider the local solution issue in the M-step.

## 3. ALGORITHM

[11] derived an E-M algorithm for computing the MLE for the factor model. In this section we derive an E-M algorithm for computing the $\ell_1$ penalized estimator. It turns out that the penalized estimator can be computed by iterative lasso-penalized least squares.

For convenience we define some notation. We use $M[i,]$ to denote the $i$-th row vector of a matrix $\mathbf{M}$. Likewise, $M[,j]$ represents the $j$-th column vector of $\mathbf{M}$. The $i,j$ entry of $\mathbf{M}$ is $M_{ij}$.

Finding the "missing data" is a key component in the derivation of an E-M algorithm. From the model (2) we naturally take $\mathbf{X}$ as the missing data. By $X_i \sim N(0, \boldsymbol{I}_q)$ we write down the joint likelihood of $(\mathbf{Y}, \mathbf{X})$ as

$$\begin{aligned}
&L_{y,x}(\tau^2, \beta) = \\
&[2\pi \prod_{j=1}^{p} \tau_j^2]^{-n/2} \exp[-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\frac{(Y_{ij} - X[i,]\beta[,j])^2}{\tau_j^2}] \\
&\times [2\pi det\mathbf{I}]^{-n/2} \exp[-\frac{1}{2}\sum_{i=1}^{n} X[i,]X[i,]^T]
\end{aligned}$$

EM algorithms iterate between the E-step and the M-step. Let $(\boldsymbol{\beta}_{(k)}, \boldsymbol{\tau}^2_{(k)})$ be the estimates of step $k$. At the E-step, we need to compute the conditional expectation of the log-likelihood given $\mathbf{Y}$ and $(\boldsymbol{\beta}_{(k)}, \boldsymbol{\tau}^2_{(k)})$. Let $ELL_{(k)}$ be the conditional expectation of the log-likelihood. We have

$$\begin{aligned}
& ELL_{(k)}(\boldsymbol{\beta}, \boldsymbol{\tau}^2) \\
= \;& E(\log P(\mathbf{X}, \mathbf{Y}|\boldsymbol{\beta}, \boldsymbol{\tau}^2)|\mathbf{Y}, \boldsymbol{\beta}_{(k)}, \boldsymbol{\tau}^2_{(k)}) \\
= \;& -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\frac{1}{\tau_j^2}\Big(Y_{ij}^2 - 2Y_{ij}E(X[i,]|\mathbf{Y}, \boldsymbol{\beta}_{(k)}, \boldsymbol{\tau}^2_{(k)})\beta[,j]\Big) \\
& +\beta[,j]^T E(X[i,]^T X[i,]|\mathbf{Y}, \boldsymbol{\beta}_{(k)}, \boldsymbol{\tau}^2_{(k)})\beta[,j]] \\
& -\frac{1}{2}\sum_{i=1}^{n} E(X[i,]X[i,]^T|\mathbf{Y}, \boldsymbol{\beta}_{(k)}, \boldsymbol{\tau}^2_{(k)}) - \frac{n}{2}\sum_{j=1}^{p}\log\tau_j^2 \\
& +constant
\end{aligned}$$

Since

$$\mathbf{X}|\mathbf{Y}, \boldsymbol{\beta}, \boldsymbol{\tau}^2 \sim N(\mathbf{Y}(\boldsymbol{\tau}^2 + \boldsymbol{\beta}^T\boldsymbol{\beta})^{-1}\boldsymbol{\beta}^T, \mathbf{I} - \boldsymbol{\beta}(\boldsymbol{\tau}^2 + \boldsymbol{\beta}^T\boldsymbol{\beta})^{-1}\boldsymbol{\beta}^T)$$

we can write

$$\begin{aligned}
E(X[i,]|\mathbf{Y}, \boldsymbol{\beta}_{(k)}, \boldsymbol{\tau}^2_{(k)}) &= \boldsymbol{\delta}^T Y[i,]^T \\
Var(X[i,]|\mathbf{Y}, \boldsymbol{\beta}_{(k)}, \boldsymbol{\tau}^2_{(k)}) &= \boldsymbol{\Delta} \\
E(X[i,]^T X[i,]|\mathbf{Y}, \boldsymbol{\beta}_{(k)}, \boldsymbol{\tau}^2_{(k)}) &= \boldsymbol{\Delta} + \boldsymbol{\delta}^T Y[i,]^T Y[i,]\boldsymbol{\delta}
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{\delta} &= (\boldsymbol{\tau}^2_{(k)} + \boldsymbol{\beta}^T_{(k)}\boldsymbol{\beta}_{(k)})^{-1}\boldsymbol{\beta}^T_{(k)}, \\
\boldsymbol{\Delta} &= \mathbf{I} - \boldsymbol{\beta}_{(k)}(\boldsymbol{\tau}^2_{(k)} + \boldsymbol{\beta}^T_{(k)}\boldsymbol{\beta}_{(k)})^{-1}\boldsymbol{\beta}^T_{(k)}.
\end{aligned}$$

We treat $\frac{1}{2}\sum_{i=1}^{n} E(X[i,]X[i,]^T|\mathbf{Y}, \boldsymbol{\beta}_{(k)}, \boldsymbol{\tau}^2_{(k)})$ as a constant because it does not involve $\boldsymbol{\beta}$ or $\boldsymbol{\tau}^2$. Hence without the constants $ELL_{(k)}(\boldsymbol{\beta}, \boldsymbol{\tau}^2)$ can be expressed as

$$\begin{aligned}
ELL_{(k)}(\boldsymbol{\beta}, \boldsymbol{\tau}^2) &= -\frac{1}{2}\sum_{j=1}^{p} n\log\tau_j^2 \\
& -\frac{1}{2}\sum_{j=1}^{p}\sum_{i=1}^{n}\frac{Y_{ij}^2 - 2Y_{ij}Y[i,]\boldsymbol{\delta}\beta[,j]}{\tau_j^2} \\
& -\frac{1}{2}\sum_{j=1}^{p}\sum_{i=1}^{n}\frac{\beta[,j]^T[\boldsymbol{\Delta} + \boldsymbol{\delta}^T Y[i,]^T Y[i,]\boldsymbol{\delta}]\beta[,j]}{\tau_j^2}.
\end{aligned}$$

As the M step, we maximize the so-called $Q$ function defined as

$$Q(\boldsymbol{\beta}, \boldsymbol{\tau}^2) = ELL_{(k)}(\boldsymbol{\beta}, \boldsymbol{\tau}^2) - \frac{1}{2}P_\lambda(\boldsymbol{\beta}). \tag{8}$$

However, it would take another iterative process to find the maximizer of the $Q$ function. To mitigate the computation difficulty, we just find an update to increase the $Q$ function rather than maximize it. This idea was introduced in the original EM paper [2]. First, we find $\boldsymbol{\tau}^2_{(k+1)}$ by letting

$$\boldsymbol{\tau}^2_{(k+1)} = \arg\max_{\tau^2}[Q(\boldsymbol{\beta}, \boldsymbol{\tau}^2)|\boldsymbol{\beta} = \boldsymbol{\beta}_{(k)}]. \tag{9}$$

Then we compute $\boldsymbol{\beta}_{(k+1)}$ by

$$\boldsymbol{\beta}_{(k+1)} = \arg\max_{\beta}[Q(\boldsymbol{\beta}, \boldsymbol{\tau}^2)|\boldsymbol{\tau}^2 = \boldsymbol{\tau}^2_{(k+1)}]. \tag{10}$$

It is easy to see that

$$\begin{aligned}
&\tau^2_{(k+1),j} = \\
&\frac{1}{n}\sum_{i=1}^{n} Y_{ij}^2 - 2Y_{ij}Y[i,]\boldsymbol{\delta}\beta_{(k)}[,j] \\
&+\frac{1}{n}\sum_{i=1}^{n}\beta_{(k)}[,j]^T[\boldsymbol{\Delta} + \boldsymbol{\delta}^T Y[i,]^T Y[i,]\boldsymbol{\delta}]\beta_{(k)}[,j].
\end{aligned} \tag{11}$$

Given $\boldsymbol{\tau}^2_{(k+1)}$, we solve $p$ separate maximization problems to get $\beta_{(k+1)}[,j], j = 1, 2, \ldots, p$. By straightforward calculations, we have

$$\begin{aligned}
&\beta_{(k+1)}[,j] = \\
&\arg\min_\beta \quad -\frac{2(\sum_{i=1}^{n} Y_{ij}Y[i,])\boldsymbol{\delta}\beta}{\tau^2_{(k+1),j}} \\
&\qquad +\frac{\beta^T[n\boldsymbol{\Delta} + \boldsymbol{\delta}^T\mathbf{Y}^T\mathbf{Y}\boldsymbol{\delta}]\beta}{\tau^2_{(k+1),j}} + \sum_{l=1}^{q}\lambda|\beta_{lj}|.
\end{aligned} \tag{12}$$

Note that (12) can be regarded as a lasso-penalized least square problem. Thus we can efficiently compute $\beta_{(k+1)}[,j]$ by using the LARS-Lasso algorithm [3].

If let $\boldsymbol{\Sigma}^s = \frac{1}{n}\mathbf{Y}^T\mathbf{Y}$. We can rewrite (11) and (12) as

$$\begin{aligned}
&\tau^2_{(k+1),j} = \\
&\boldsymbol{\Sigma}^2_{jj} - 2\boldsymbol{\Sigma}[j,]\boldsymbol{\delta}\beta_{(k)}[,j] + \beta_{(k)}[,j]^T[\boldsymbol{\Delta} + \boldsymbol{\Sigma}]\beta_{(k)}[,j].
\end{aligned} \tag{13}$$

$$\beta_{(k+1)}[,j] = \arg\min_{\beta} -\frac{2\Sigma^s[j,]\boldsymbol{\delta}}{\tau^2_{(k+1),j}}\beta + \frac{\beta^T[\boldsymbol{\Delta}+\boldsymbol{\Sigma}]\beta}{\tau^2_{(k+1),j}} + \sum_{l=1}^{q}\frac{\lambda}{n}|\beta_{lj}|. \tag{14}$$

The above procedure is summarized in Algorithm 1. We call algorithm 1 a generalized expectation-maximization (GEM) algorithm, because in the M-step the $Q$ function is penalized condition log-likelihood and we increase the $Q$ function rather than maximize it.

Algorithm 1 can also be used to compute the penalized estimator using a general penalty function $P_\lambda(|\beta|)$. We just replace $\lambda|\beta_l|$ with $P_\lambda(|\beta_l|)$ in step (3.a). The $\ell_1$ penalty enjoys great computational advantages, for we can use the LARS-Lasso algorithm to solve the $\ell_1$-penalized least squares problem in the same order of computations of an ordinary least-squares fit[3].

As a generalized E-M algorithm, algorithm 1 enjoys a nice ascent property which is formally proven in the Appendix. We should also point out that the ascent property has nothing to do with the normality assumption of the data, although we interpret the objective function as penalized log-likelihood of normal data.

**Algorithm 1: GEM for sparse factor analysis**

```
Step 0. Compute $\boldsymbol{\Sigma}^s = \mathbf{Y}^T\mathbf{Y}/n$.
Step 1 : Set initial values for $\boldsymbol{\beta}$ and $\boldsymbol{\tau}^2$.
Step 2 : Calculate $\boldsymbol{\delta}$, $\boldsymbol{\Delta}$ :
    $\boldsymbol{\delta} = (\boldsymbol{\tau}^2 + \boldsymbol{\beta}^T\boldsymbol{\beta})^{-1}\boldsymbol{\beta}^T$
    $\boldsymbol{\Delta} = \mathbf{I} - \boldsymbol{\beta}(\boldsymbol{\tau}^2 + \boldsymbol{\beta}^T\boldsymbol{\beta})^{-1}\boldsymbol{\beta}^T$
    Compute the Cholesky decomposition: $\mathbf{Z}^T\mathbf{Z} = \boldsymbol{\Delta} + \boldsymbol{\Sigma}^s$.
Step 3 : For $j = 1, \dots, p$
    (3.a) Compute $\tau_j^2$ by (13).
    (3.b) Compute $\tilde{Y} = (\Sigma^s[j,]\boldsymbol{\delta}\mathbf{Z}^{-1})^T$. Then solve the following penalized least squares problem:
          $\beta[,j] = \arg\min_{\beta} \|\tilde{Y} - \mathbf{Z}\beta\|_2^2 + \frac{\tau_j^2}{n}\sum_{l=1}^{q}\lambda|\beta_l|.$
Step 4 : Repeat Steps 2–3 till convergence.
```

## 4. NUMERICAL EXAMPLES

In this section we use both simulated and real data to demonstrate the proposed $\ell_1$ penalized estimators.

### 4.1 Simulation data

We examine the performance of the lasso and ALasso estimators in the situation where the factor loadings matrix is sparse. The simulation data are generated by taking i.i.d. random vectors $Y_i$ of length 12 from normal distribution with zero mean and covariance $\boldsymbol{\Sigma} = \boldsymbol{\beta}^T\boldsymbol{\beta} + \boldsymbol{\tau}^2$ where

$$\boldsymbol{\beta}^T = \begin{bmatrix} 1.8 & 0 & 0 & 0 \\ 1.8 & 0 & 0 & 0 \\ 1.8 & 0 & 0 & 0 \\ 0 & 1.7 & 0 & 0 \\ 0 & 1.7 & 0 & 0 \\ 0 & 1.7 & 0 & 0 \\ 0 & 0 & 1.6 & 0 \\ 0 & 0 & 1.6 & 0 \\ 0 & 0 & 1.6 & 0 \\ 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 1.5 \end{bmatrix},$$

$$\boldsymbol{\tau}^2 = \operatorname{diag}\ (1.27, 0.61, 0.74, 0.88, 0.65, 0.81, 0.74, 1.30, 1.35, 0.74, 0.92, 1.32).$$

The interpretation of this factor model is that variables $3k-2$, $3k-1$ and $3k$ are random perturbations of factor $k$, $k = 1, 2, 3, 4$. Within each of 100 replications we generated 100 training data and an independent 100 validation data. In this simulation study we compared four methods: the ordinary MLE, the lasso and ALasso estimators and the oracle estimator which is defined as the MLE when knowing which entries of the factor loading matrix should be zero.

Suppose a method $\mu$ produces an estimator $\widehat{\boldsymbol{\beta}}(\mu)$ and $\widehat{\boldsymbol{\tau}^2}(\mu)$. Write

$$\boldsymbol{\Sigma}(\mu) \equiv \widehat{\boldsymbol{\beta}}(\mu)^T\widehat{\boldsymbol{\beta}}(\mu) + \widehat{\boldsymbol{\tau}^2}(\mu).$$

We define two K-L measurements of $\mu$ as follows

$$KL(\mu) = \frac{1}{2}\log(det(\boldsymbol{\Sigma}(\mu)) + \frac{1}{2}tr(\boldsymbol{\Sigma}(\mu)^{-1}\boldsymbol{\Sigma})) - \frac{1}{2}\log(det(\boldsymbol{\Sigma})) - \frac{p}{2}. \tag{15}$$

$$KL(\mu)_{\mathrm{v}} = \frac{1}{2}\log(det(\boldsymbol{\Sigma}(\mu)) + \frac{1}{2}tr(\boldsymbol{\Sigma}(\mu)^{-1}\boldsymbol{\Sigma}_{\mathrm{v}})) - \frac{1}{2}\log(det(\boldsymbol{\Sigma}_{\mathrm{v}})) - \frac{p}{2}. \tag{16}$$

where $\boldsymbol{\Sigma}_{\mathrm{v}}$ is the sample covariance matrix computed using the validation data. The $KL$ loss in (15) measures the goodness of fit of $\mu$ and $KL_{\mathrm{v}}$ in (16) is used to select meta-parameters (if any) of $\mu$. We report the relative K-L loss (RKL) defined as $\frac{KL(\mu)}{KL(\text{mle})}$.

The true model has $q = 4$ factors. We did not use this information in our study. We treated $q$ as a meta-parameter of the MLE and used the one minimizing $KL(\text{mle})_{\mathrm{v}}$. In all 100 replications, the selected $q$ number was four. Figure 1 shows a typical plot of $KL(\text{mle})_{\mathrm{v}}$ vs. $q$.

| method | RKL | Number of zeros |
|---|---|---|
| Oracle | 0.415 (0.009) | 36 |
| Lasso | 0.874 (0.009) | 15 (0.49) |
| ALasso | 0.499 (0.010) | 34 (0.28) |

*Table 1. Averages based on 100 replications. Numbers in $(\cdot)$ are standard errors.*

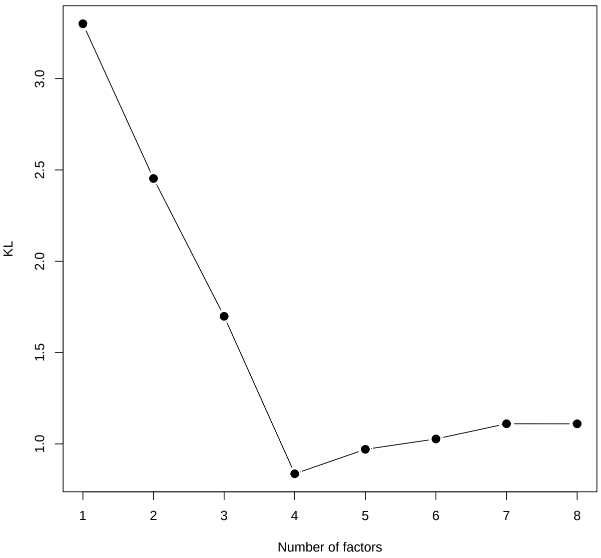

*Figure 1. The $y$-axis shows the value of $KL(\mathrm{mle})_{\mathrm{v}}$ for $q$ factors.*

From Figure 2 and the second column of Table 1, it is very clear that both the Lasso and the ALasso estimators are more accurate than the MLE. Moreover, ALasso is more accurate than Lasso and is also very close to the oracle. Figure 3 displays the pairwise comparison of the ALasso and the oracle in the 100 replications. It is interesting to see that nine out of 100 times the ALasso did slightly better than the oracle.

From the third column of Table 1 we see that the ALasso discovered many more zero loadings than the Lasso did. To visualize their difference, we made the histogram of the number of estimated zero loadings for both Lasso and ALasso, as shown in Figure 4.

## 4.2 Real data

As an application, we apply the proposed sparse factor analysis method to analyze Oxford Parkinson data [10]. The data can be downloaded from UCI Data Repository [1]. This dataset has 195 samples and 23 features. We randomly split the data into a training set (130 observations) and a validation set (65 observations). The accuracy of each model is measured by its K-L loss evaluated on the validation set. Before fitting any factor model, we standardized the data such that each feature has zero mean and standard deviation one.

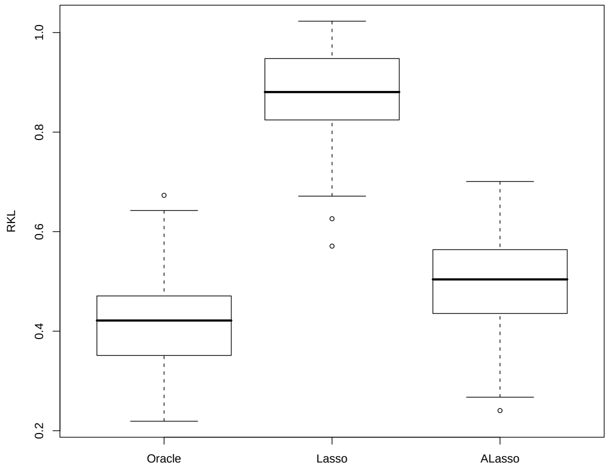

*Figure 2. Boxplots of RKLs of the Lasso, ALasso and oracle estimators with respect to the MLE.*

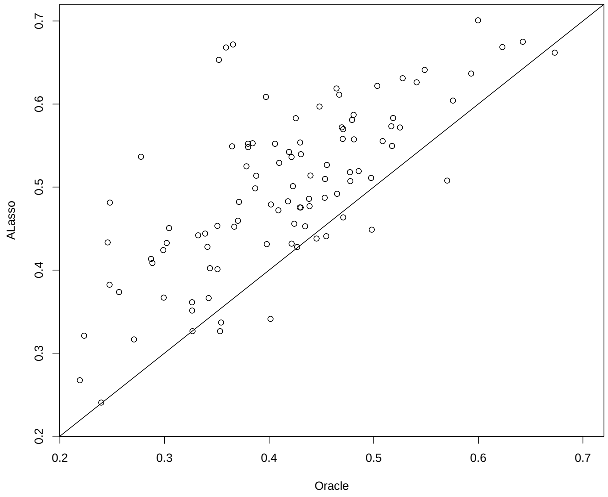

*Figure 3. Pairwise comparison of ALasso and oracle. The solid straight line is the 45 degree line.*

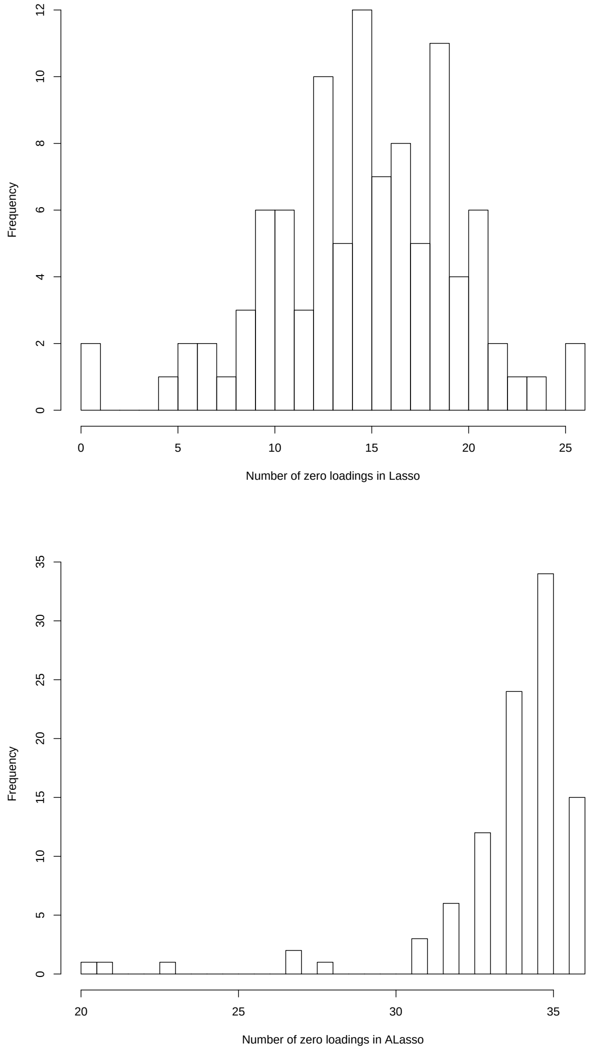

*Figure 4. Comparing the sparsity pursuit performance of Lasso and ALasso (top histogram: Lasso, bottom histogram: ALasso).*

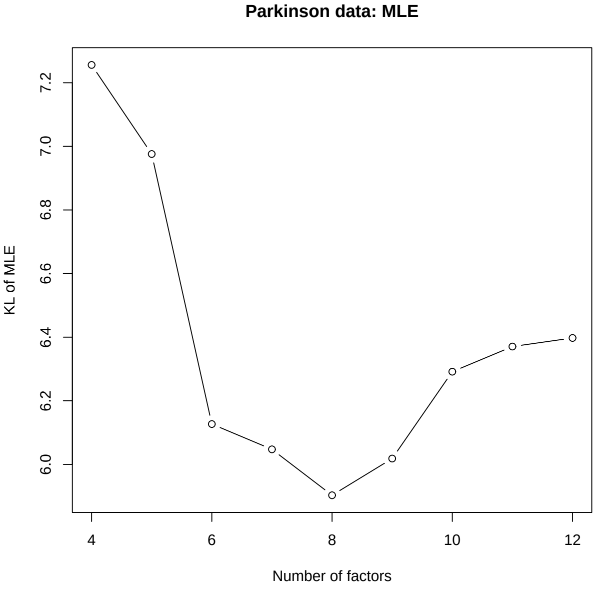

*Figure 5. The best MLE model uses 8 factors.*

Figure 5 suggests that we should consider a factor model with 8 factors. We fit the lasso and ALasso factor models using 8 factors for a grid of penalization parameters. The smallest KL-loss by the lasso and the ALasso models is 5.83 while the K-L loss of the MLE model is 5.90. Since there is little room for improving the accuracy of the MLE by $\ell_1$ penalization, it seems more reasonable to use the sparsity-first rule, namely that we use a sparse factor model with the highest sparsity as long as its K-L loss is smaller than that of the MLE. The sparsity-first rule chooses a lasso model with 19 zero loadings and an ALasso model with 32 zero loadings.

Lastly, we provide a numerical demonstration of the ascent property of the GEM algorithm. We monitored the GEM iterations when computing a Lasso model using $\lambda = 3n$ ($n$ is the training sample size). The GEM algorithm started at the MLE. Figure 6 displays the value of $PLL$ after every 50 GEM iterations. It is clear that the PLL curve is monotonically increasing.

## 5. DISCUSSION

We have proposed an $\ell_1$ penalized maximum likelihood estimation method to do sparse factor analysis. We have shown that the $\ell_1$ penalized maximum likelihood estimation can be done via a generalized E-M algorithm that is equivalent to iterative $\ell_1$-penalized least squares. We have observed that if the data are generated from a true sparse factor model, the $\ell_1$ penalized models not only discover zero loadings but also are significantly more accurate than the MLE model. The ALasso model performs similarly to the oracle MLE. In some applications, $\ell_1$ penalization only slightly

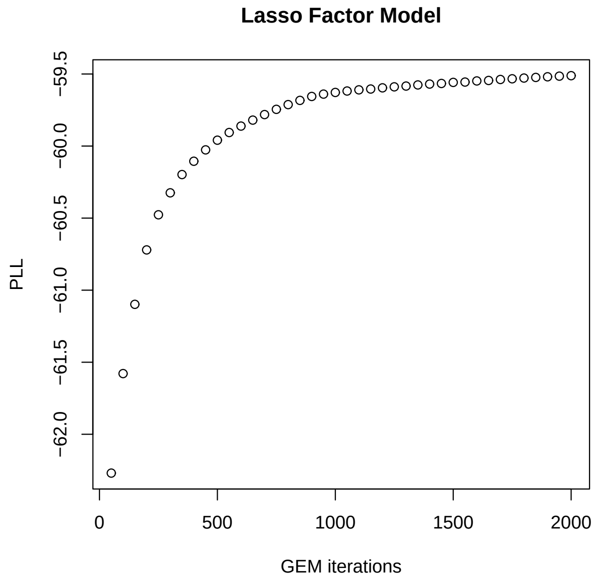

*Figure 6. A numerical demonstration of Algorithm 1's ascent property using Parkinson data.*

improves the accuracy of the MLE model. In such situations we suggest to use the sparsity-first rule to pick the optimal penalization parameter in $\ell_1$ penalized models, because $\ell_1$ penalization is primarily employed for pursuing sparsity.

## APPENDIX

**Ascent property of Algorithm 1.** The E-M algorithm is usually used for maximum likelihood estimation. [6] showed that the E-M algorithm can also be used for penalized maximum likelihood estimation. Here we provide a self-contained proof of the ascent property of the generalized E-M algorithm considered in Section 3.

For simplicity, we use $f$ to denote a generic density function. The penalized log-likelihood function is

$$PLL(\boldsymbol{\tau}^2, \boldsymbol{\beta}) = \log(f(\mathbf{Y}|\boldsymbol{\tau}^2, \boldsymbol{\beta})) - \frac{1}{2}P_\lambda(\boldsymbol{\beta})$$

Given the $k$-th estimate $\boldsymbol{\tau}^2_{(k)}, \boldsymbol{\beta}_{(k)}$, the $Q$ function is constructed by

$$Q(\boldsymbol{\tau}^2, \boldsymbol{\beta}) = \int \log(f(\mathbf{X}, \mathbf{Y}|\boldsymbol{\tau}^2, \boldsymbol{\beta})) f(\mathbf{X}|\mathbf{Y}, \boldsymbol{\tau}^2_{(k)}, \boldsymbol{\beta}_{(k)}) d\mathbf{X} - \frac{1}{2}P_\lambda(\boldsymbol{\beta}).$$

We can write

$$\begin{aligned} & Q(\boldsymbol{\tau}^2, \boldsymbol{\beta}) = PLL(\boldsymbol{\tau}^2, \boldsymbol{\beta}) \\ & + \int \log(f(\mathbf{X}|\mathbf{Y}, \boldsymbol{\tau}^2_{(k)}, \boldsymbol{\beta}_{(k)})) f(\mathbf{X}|\mathbf{Y}, \boldsymbol{\tau}^2_{(k)}, \boldsymbol{\beta}_{(k)}) d\mathbf{X} \\ & + \int \log\left(\frac{f(\mathbf{X}|\mathbf{Y}, \boldsymbol{\tau}^2, \boldsymbol{\beta})}{f(\mathbf{X}|\mathbf{Y}, \boldsymbol{\tau}^2_{(k)}, \boldsymbol{\beta}_{(k)})}\right) f(\mathbf{X}|\mathbf{Y}, \boldsymbol{\tau}^2_{(k)}, \boldsymbol{\beta}_{(k)}) d\mathbf{X}. \end{aligned}$$

By Jensen's inequality, the third term is non-negative and hence we have

$$Q(\boldsymbol{\tau}^2, \boldsymbol{\beta}) \quad \leq \quad PLL(\boldsymbol{\tau}^2, \boldsymbol{\beta}) + C_k$$

where $C_k = \int \log(f(\mathbf{X}|\mathbf{Y}, \boldsymbol{\tau}^2_{(k)}, \boldsymbol{\beta}_{(k)})) f(\mathbf{X}|\mathbf{Y}, \boldsymbol{\tau}^2_{(k)}, \boldsymbol{\beta}_{(k)}) d\mathbf{X}$ is a constant. By (9) and (10) we have

$$Q(\boldsymbol{\tau}^2_{(k+1)}, \boldsymbol{\beta}_{(k)}) - C_k \geq Q(\boldsymbol{\tau}^2_{(k)}, \boldsymbol{\beta}_{(k)}) - C_k$$

$$Q(\boldsymbol{\tau}^2_{(k+1)}, \boldsymbol{\beta}_{(k+1)}) - C_k \geq Q(\boldsymbol{\tau}^2_{(k+1)}, \boldsymbol{\beta}_{(k)}) - C_k$$

Thus, we conclude

$$\begin{aligned} PLL(\boldsymbol{\tau}^2_{(k+1)}, \boldsymbol{\beta}_{(k+1)}) & \geq & Q(\boldsymbol{\tau}^2_{(k+1)}, \boldsymbol{\beta}_{(k+1)}) - C_k \\ & \geq & Q(\boldsymbol{\tau}^2_{(k)}, \boldsymbol{\beta}_{(k)}) - C_k \\ & = & PLL(\boldsymbol{\tau}^2_{(k)}, \boldsymbol{\beta}_{(k)}). \end{aligned}$$

## REFERENCES

[1] Asuncion, A. and Newman, D.J. (2007). *UCI Machine Learning Repository* [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Dempster, A., Laird, N. and Rubin, D. (1977). *Maximum Likelihood from Incomplete Data via the EM Algorithm (with discussion). Journal of the Royal Statistical Society, series B* **39**, 1–38.

[3] Efron, B.,Hastie, T., Johnstone, I. and Tibshirani,R. (2003). *Least Angle Regression. The Annals of Statistics* **32** 407–499.

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

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

[6] Green, P. (1990). *On Use of the EM Algorithm for Penalized Likelihood Estimation. Journal of the Royal Statistical Society, series B* **52** 443–452.

[7] Gorsuch, R. (1983). *Factor Analysis.* Hillsdale, NJ: Lawrence Erlbaum.

[8] Hesterberg, T., Choi, N., Meier, L. and Fraley, C. (2008). *Least angle and $\ell_1$ penalized regression: A review. Statistics Survey* **2** 61–93.

[9] Johnson, R. and Wichern, D. (2007). *Applied Multivariate Statistical Analysis.* 6th ed. New Jersey: Pearson Education, Inc.

[10] Little, M., McSharry, P., Hunter, E. and Ramig, L. (2008). *Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. IEEE Transactions on Biomedical Engineering* **56** 1015–1022.

[11] Rubin, D. and Thayer, D. (1982). *EM Algorithms For ML Factor Analysis. Psychometrika* **47** 69–76.

[12] Spearman, C. (1904). *General Intelligence, Objectively Determined and Measured. The American Journal of Psychology* **15** 201–293.

[13] Tibshirani, R. (1996). *Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society, series B* **58** 267–288.

[14] Zou, H. (2006). *The Adaptive Lasso and Its Oracle properties. Journal of the American Statistical Association* **101** 1418–1429.

[15] Zou, H., Hastie, T. and Tibshirani, R. (2006). *Sparse Principal Component Analysis. Journal of Computational and Graphical Statistics* **15** 265–286.

Jang Choi
School of Statistics
University of Minnesota
E-mail address: jangchoi@stat.umn.edu

Hui Zou
School of Statistics
University of Minnesota
E-mail address: hzou@stat.umn.edu

Gary Oehlert
Address of the Third author
School of Statistics
University of Minnesota
E-mail address: gary@stat.umn.edu