PRINCIPAL COMPONENTS ANALYSIS FOR BINARY DATA

A Dissertation

by

SEOKHO LEE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2009

Major Subject: Statistics

PRINCIPAL COMPONENTS ANALYSIS FOR BINARY DATA

A Dissertation

by

SEOKHO LEE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	Jianhua Z. Huang
	Raymond J. Carroll
Committee Members,	Soumendra N. Lahiri
	Alan Dabney
	Ivan V. Ivanov
Head of Department,	Simon J. Sheather

May 2009

Major Subject: Statistics

ABSTRACT

Principal Components Analysis for Binary Data. (May 2009)

Seokho Lee, B.S., Seoul National University; M.S., Seoul National University

Co-Chairs of Advisory Committee: Dr. Jianhua Z. Huang
Dr. Raymond J. Carroll

Principal components analysis (PCA) has been widely used as a statistical tool for the dimension reduction of multivariate data in various application areas and extensively studied in the long history of statistics. One of the limitations of PCA machinery is that PCA can be applied only to the continuous type variables. Recent advances of information technology in various applied areas have created numerous large diverse data sets with a high dimensional feature space, including high dimensional binary data. In spite of such great demands, only a few methodologies tailored to such binary dataset have been suggested. The methodologies we developed are the model-based approach for generalization to binary data. We developed a statistical model for binary PCA and proposed two stable estimation procedures using MM algorithm and variational method. By considering the regularization technique, the selection of important variables is automatically achieved. We also proposed an efficient algorithm for model selection including the choice of the number of principal components and regularization parameter in this study.

Dedicated to Hyejin and Jake

ACKNOWLEDGEMENTS

I am grateful to my dissertation advisors and committee members for their interaction and support during my graduate study. This dissertation is dedicated to my family for their endless encouragement, support and love.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Principal components analysis (PCA) is probably the oldest and best known technique of multivariate analysis. It was introduced by Pearson (1901), and developed independently by Hotelling (1933). The central idea of principal components analysis is to reduce the dimensionality of a data set in which there are a large number of interrelated variables, while retaining as much as possible of the variation present in the data set (Jolliffe, 2004). Its applications include exploratory data analysis, visualization, denoising and feature selections (Hastie et al., 2001; Bishop, 2006).

Although PCA has a lot of possible applications, its computation and interpretation is tailored to only continuous type variables so that there is a need to develop PCA-like dimension reduction machinery for the other type variables, including binary variables in which we are interested in this study. Many attempts to generalize PCA to other type variables can be found in Jolliffe (2004). In our study, we review and discuss the existing generalization of PCA and we give further steps to answer the important and interesting questions in practice, arising from PCA with binary variables, for example, the selection of the number of principal components and the computation of PCA in high-dimensional situation.

Recently, there has been an increasing attention on the sparsity-introduced PCA (Jolliffe et al., 2003; Zou et al., 2006; Shen and Huang, 2008). The standard PCA suffers from the fact that the derived principal component is a linear combination of all the original variables, so that it is often difficult to interpret the results. The idea of sparse principal

The format and style follow that of *Biometrics*.

components analysis is to produce modified principal components with sparse loadings. In other words, sparse PCA seeks principal component loadings with very few non-zero elements. This will not only lead to the simple structure of principal components with an easy interpretation, but also make the extraction of principal components more stable. The existing sparse PCA methods are mostly suitable to continuous type variables and they are not generally appropriate for other types such as binary or counts. The goal of this study is to develop a sparse principal component analysis method for binary data.

To this end, first, we review the formulation of standard PCA problem and explore a possible generalization of it to binary variables.

1.1 Formulations of Principal Components Analysis

There are two commonly adopted definitions of PCA that give rise to the same result. PCA can be defined as the orthogonal projection of the data onto a low dimensional linear subspace, known as the principal subspace, such that the variance of the projected data is maximized (Hotelling, 1933). Equivalently, PCA can be defined as the linear projection that minimizes the mean squared distance between the data points and their projections (Pearson, 1901). The process of orthogonal projection is illustrated in Figure 1. In the following, we consider each of these definitions in turn. These two definitions will shed a light on the generalization of PCA to binary variables and show the relation between PCA problem and regression problem.

1.1.1 Maximum variance formulation

The first formulation of standard PCA, which will be described here, is due to Hotelling (1933). Consider a data set of n observations $\mathbf{y}_1, \cdots, \mathbf{y}_n$ in $\mathbb{R}^d$. In other words, the collected data comprises d variables all of which are continuous. The goal of PCA is to project the data onto a low-dimensional subspace while maximizing the variance of the projected

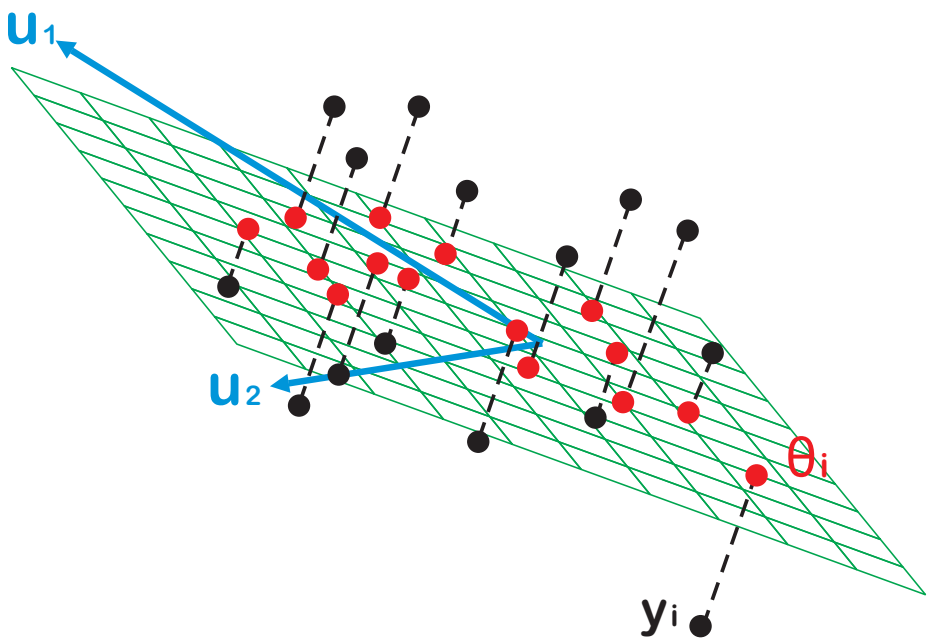


Figure 1: Principal components analysis seeks a space of lower dimensionality, known as principal subspace and denoted by the green grid, such that the orthogonal projection of the data points (black dots) onto this subspace maximizes the variance of the projected points (red dots). An alternative definition of PCA is based on minimizing the sum of squares of the projection errors, indicated by the dashed black lines.

data. To begin with, consider the projection onto a one-dimensional space. We can define the direction of this space using d-dimensional vector $\mathbf{u}_1$, which for convenience (and with-

out loss of generality) we shall choose to be a unit vector so that $\mathbf{u}_1^T\mathbf{u}_1 = 1$ because we are only interested in the "direction" defined by $\mathbf{u}_1$, not in the magnitude of $\mathbf{u}_1$ itself. Each data point $\mathbf{y}_i$ is then projected onto a scalar value $\alpha_{i1} = \mathbf{u}_1^T(\mathbf{y}_i - \bar{\mathbf{y}})$ after subtracting the sample mean $\bar{\mathbf{y}} = \sum_{i=1}^n \mathbf{y}_i/n$. The mean and variance of the projected data α_{i1} $(i = 1, \cdots, n)$ are, then, given by

$$\begin{aligned} \text{mean}(\alpha_1) &= \frac{1}{n}\sum_{i=1}^n \alpha_{i1} = \frac{1}{n}\mathbf{u}_1^T\sum_{i=1}^n(\mathbf{y}_i - \bar{\mathbf{y}}) = 0 \\ \text{var}(\alpha_1) &= \frac{1}{n}\sum_{i=1}^n \alpha_{i1}^2 = \frac{1}{n}\sum_{i=1}^n \mathbf{u}_1^T(\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^T\mathbf{u}_1 = \mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 \end{aligned}$$

where

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^n(\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^T.$$

We now maximize the projected variance $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$ with respect to $\mathbf{u}_1$. This is a constrained maximization to prevent $||\mathbf{u}_1|| \to \infty$. The appropriate constraint comes from the normalization condition $\mathbf{u}_1^T\mathbf{u}_1 = 1$. Therefore, the constrained maximizer becomes

$$\mathbf{u}_1 = \max_{\mathbf{u}:\mathbf{u}^T\mathbf{u}=1} \mathbf{u}^T\mathbf{S}\mathbf{u} = \max_{\mathbf{u}} \frac{\mathbf{u}^T\mathbf{S}\mathbf{u}}{\mathbf{u}^T\mathbf{u}}.$$

To enforce this constraint, one may introduce a Lagrange multiplier that we shall denote by λ_1, and then make an unconstrained maximization of

$$\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 + \lambda_1(1 - \mathbf{u}_1^T\mathbf{u}_1).$$

By setting the derivative with respect to $\mathbf{u}_1$ equal to zero, we see that this quantity will have a stationary point when

$$\mathbf{S}\mathbf{u}_1 = \lambda_1\mathbf{u}_1$$

which says that $\mathbf{u}_1$ must be an eigenvector of $\mathbf{S}$. Using the unity constraint, the variance is given by

$$\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 = \lambda_1$$

and so the variance will be maximized when we set $\mathbf{u}_1$ equal to the eigenvector having the largest eigenvalue. This eigenvector is called as the first principal component.

Additional principal components can be defined in an incremental fashion by choosing each new direction to be that which maximizes the projected variance among all possible directions orthogonal to those already considered. So, the lth principal component $\mathbf{u}_l$ can be found by solving

$$\mathbf{u}_l \;=\; \max_{\mathbf{u}:\mathbf{u}^T\mathbf{u}=1,\mathbf{u}^T\mathbf{u}_m=0} \mathbf{u}^T\mathbf{S}\mathbf{u}$$

where $m \;=\; 1, \cdots, l-1$. Suppose $\mathbf{u}_1, \cdots, \mathbf{u}_{l-1}$ are previously selected the first $l-1$ principal components. The Lagrangian of this constrained maximization is given by

$$\mathbf{u}_l^T\mathbf{S}\mathbf{u}_l + \lambda_l(1 - \mathbf{u}_l^T\mathbf{u}_l) + \tau_1\mathbf{u}_l^T\mathbf{u}_1 + \cdots + \tau_{l-1}\mathbf{u}_l^T\mathbf{u}_{l-1}$$

by considering the unity constraint $\mathbf{u}_l^T\mathbf{u}_l = 1$ and the orthogonal constraints $\mathbf{u}_l^T\mathbf{u}_m = 0$ for $m = 1, \cdots, l-1$. Setting the derivative with respect to $\mathbf{u}_l$ to zero leads to

$$2\mathbf{S}\mathbf{u}_l - 2\lambda_l\mathbf{u}_l + \tau_1\mathbf{u}_1 + \cdots \tau_{l-1}\mathbf{u}_{l-1} \;=\; \mathbf{0}.$$

From the orthonormality constraints, we can easily see that $\tau_m = 0$ for $m = 1, \cdots, l-1$. So, this leads to

$$\mathbf{S}\mathbf{u}_l \;=\; \lambda_l\mathbf{u}_l$$

and so $\mathbf{u}_l$ must be an eigenvector of $\mathbf{S}$ with eigenvalue λ_l. The variance in the direction $\mathbf{u}_l$ is given by $\mathbf{u}_l^T\mathbf{S}\mathbf{u}_l = \lambda_l$ and so is maximized by choosing $\mathbf{u}_l$ to be the eigenvector having the largest eigenvalue among those are not previously selected.

Thus, if we consider the general case of an k-dimensional projection space, the optimal linear projection for which the variance of the projected data is maximized is now defined by the k eigenvectors $\mathbf{u}_1, \cdots, \mathbf{u}_k$ of the data covariance matrix $\mathbf{S}$ corresponding to the k

largest eigenvalues $\lambda_1, \cdots, \lambda_k$. Algorithms for finding eigenvectors and eigenvalues, as well as additional theorems related to eigenvalue decomposition, can be found in Golub and van Loan (1996). Note that the computational cost of the eigenvalue decomposition is $O(d^3)$. If we only need to project our data onto the first k principal components, then we just need to find the first k eigenvalues and eigenvectors. This can be done with more efficient techniques, such as the power method (Golub and van Loan, 1996; Jolliffe, 2004), that requires $O(kd^2)$.

1.1.2 Minimum error formulation

In this subsection, we discuss an alternative formulation of PCA based on projection error minimization (Pearson, 1901). To this end, consider complete orthonormal basis vectors $\mathbf{u}_1, \cdots, \mathbf{u}_d$ that satisfy $\mathbf{u}_l^T \mathbf{u}_m = \delta_{lm}$ where δ_{lm} is a Kronecker delta function which takes the value 1 if $l = m$ and 0 otherwise. Since this set of bases is complete, each data point can be represented by a linear combination of the basis vectors

$$\mathbf{y}_i \;=\; \bar{\mathbf{y}} + \sum_{l=1}^{d} c_{il}\mathbf{u}_l \tag{1.1}$$

where $\bar{\mathbf{y}}$, the sample mean, is a translation factor and the coefficients c_{il} will be different for different data points. Taking into account the orthonormality, we obtain $c_{il} = (\mathbf{y}_i - \bar{\mathbf{y}})^T \mathbf{u}_l$, and we can write

$$\mathbf{y}_i \;=\; \sum_{l=1}^{d} \{(\mathbf{y}_i - \bar{\mathbf{y}})^T \mathbf{u}_l\}\mathbf{u}_l.$$

Our objective is to approximate this data point using a representation involving a restricted number $k < d$ of variables corresponding to a projection onto a lower-dimensional subspace. The k-dimensional linear subspace can be represented by the first k basis vectors, and so we approximate each data point $\mathbf{y}_i$ by

$$\boldsymbol{\theta}_i \;=\; \bar{\mathbf{y}} + \sum_{l=1}^{k} \alpha_{il}\mathbf{u}_l.$$

We are free to choose $\mathbf{u}_l$ and α_{il} for $l = 1, \cdots, k$ so as to minimize the "loss" from truncation or reduction of dimensionality. As a measure of the loss, we may use the average of the squared distance between the original data point $\mathbf{y}_i$ and its low dimensional representation $\boldsymbol{\theta}_i$, so that our goal is to minimize

$$E \;=\; \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{y}_i - \boldsymbol{\theta}_i||^2 \;=\; \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{y}_i - (\bar{\mathbf{y}} + \alpha_{i1}\mathbf{u}_1 + \cdots + \alpha_{ik}\mathbf{u}_k)||^2. \tag{1.2}$$

For the minimization with respect to the quantity α_{il}, by setting the derivative with respect to α_{il} to zero and making use of the orthonormality, we obtain $\alpha_{il} = (\mathbf{y}_i - \bar{\mathbf{y}})^T\mathbf{u}_l$. If we substitute for α_{il} in (1.2) and make use of the expansion (1.1), we obtain

$$\mathbf{y}_i - \boldsymbol{\theta}_i \;=\; \sum_{l=k+1}^{d} \{(\mathbf{y}_i - \bar{\mathbf{y}})^T\mathbf{u}_l\}\mathbf{u}_l$$

from which we can see that the displacement from $\mathbf{y}_i$ to $\boldsymbol{\theta}_i$ lies in the space orthogonal to the k-dimensional principal subspace because it is a linear combination of $\mathbf{u}_{k+1}, \cdots, \mathbf{u}_d$, as illustrated in Figure 1. This is to be expected because the projected points $\boldsymbol{\theta}_i$ must lie within the principal subspace, but we can move them freely within that subspace, and so the minimum error is given by the orthogonal projection.

Therefore, the squared distance E becomes the form of

$$\begin{aligned} E \;&=\; \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{y}_i - \boldsymbol{\theta}_i||^2 \;=\; \frac{1}{n}\sum_{i=1}^{n}\sum_{l=k+1}^{d} \{(\mathbf{y}_i - \bar{\mathbf{y}})^T\mathbf{u}_l\}^2 \\ &=\; \frac{1}{n}\sum_{i=1}^{n}\sum_{l=k+1}^{d} \mathbf{u}_l^T(\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^T\mathbf{u}_l \;=\; \sum_{l=k+1}^{d} \mathbf{u}_l^T\mathbf{S}\mathbf{u}_l. \end{aligned}$$

The remaining task is to minimize E with respect to $\mathbf{u}_l$ for $l = k+1, \cdots, d$, which must be the constrained minimization otherwise we will get the trivial result $\mathbf{u}_l = \mathbf{0}$. Considering the orthonormality condition, the corresponding Lagrangian is

$$\sum_{l=k+1}^{d} \mathbf{u}_l^T\mathbf{S}\mathbf{u}_l - \sum_{l=k+1}^{d} \lambda_l(\mathbf{u}_l^T\mathbf{u}_l - 1).$$

Thus, the stationary points should satisfy a set of equations $\mathbf{S}\mathbf{u}_l = \lambda_l \mathbf{u}_l$ for $l = k+1, \cdots, d$ so that $\mathbf{u}_l$ must be eigenvectors of $\mathbf{S}$. The orthonormality condition gives the squared distance by

$$E = \sum_{l=k+1}^{d} \lambda_l$$

which is simply the sum of eigenvalues of those eigenvectors. For E to be minimized, the selected eigenvalues must be the $d-k$ smallest eigenvalues and $\mathbf{u}_l$'s are the corresponding eigenvectors. Therefore, two different formulations of PCA, maximum variance formulation and minimum error formulation, are intrinsically equivalent and lead to the eigen problem of the sample covariance matrix $\mathbf{S}$.

Unlike maximum variance formulation, however, minimum error formulation has the maximum likelihood estimation (MLE) interpretation. The objective function to be minimized, E in (1.2), can be viewed as the negative log likelihood multiplied by the constant factor $2/n$, ignoring the additive constant, when we consider Gaussian distribution on the observations $\mathbf{y}_i$, with mean $\boldsymbol{\theta}_i$ and identity covariance. Note that Gaussian distribution assumption is adopted only for the computational convenience, not for representing the actual data generating process. And, moreover, minimization of (1.2) with respect to the principal components $\mathbf{u}_l$ can be connected to the least square estimation as in regression if the coefficients α_{il} are given. These observations give us a cornerstone to develop or generalize the principal components analysis to binary variables, which is discussed in the subsequent section.

1.2 Generalization of Sparse Principal Components Analysis to Binary Variables

There are numerous attempts in the journey to the generalization of the principal components analysis for other type variables. The simple way to do it is adopting the different distribution assumption conforming to the observed variables, for example, Bernoulli dis-

tribution for binary variables, Binomial or Poisson distribution for counts, and gamma distribution for non-negative continuous variables. This approach has been extensively studied in the social science literature (Skrondal and Rabe-Hesketh, 2004, and reference therein) where the principal component scores α_{il} are treated as latent variables. In this model, the canonical parameters $\boldsymbol{\theta}_i$, analogous to mean parameters in Gaussian model, have a low-rank representation so that $\boldsymbol{\theta}_i = \boldsymbol{\mu} + \alpha_{i1}\mathbf{u}_1 + \cdots + \alpha_{ik}\mathbf{u}_k$ with a shift or intercept $\boldsymbol{\mu}$. For example, the distribution of binary variable y_{ij}, conditional on the latent variable $\boldsymbol{\alpha}_i = (\alpha_{i1}, \cdots, \alpha_{ik})^T$ is assumed to be Bernoulli distribution with success probability θ_{ij} which is the jth component of the canonical parameter vector $\boldsymbol{\theta}_i = \boldsymbol{\mu} + \alpha_{i1}\mathbf{u}_1 + \cdots + \alpha_{ik}\mathbf{u}_k$ and the latent variables $\boldsymbol{\alpha}_i$ are commonly assumed to have Gaussian distribution with zero mean and identity or diagonal covariance. With this Gaussian assumption on the latent variable, Tipping and Bishop (1999) prove that the maximum likelihood estimation for the k principal components leads to the first k eigenvectors of the covariance matrix.

This latent variable model for dimension reduction approach is called the generalized latent trait models and this latent model approach is closely connected with factor analysis (Bartholomew, 1984; Moustaki and Knott, 2000). Bartholomew (1984) laid down the foundation of factor analysis with a latent variable methods in the case that the observed variables (or manifest variables in their terminology) are binary, count or ordinal variables. Moustaki and Knott (2000) gave a general framework to provide a unified maximum likelihood method for estimating the parameters of the generalized latent trait model. These models assume that the theoretical concepts, often represented by the latent variables in the model, are not observable directly and the observed responses are treated as proxies for the concepts of interest. Thus, the integration over the latent variables is necessary to obtain the marginal likelihood but the problem is that such integration is infeasible in the case of the non-Gaussian response variables. Therefore, numerical integration techniques (such as Gauss-Hermite quadrature) or Monte Carlo integrations are often used to approximate the

integration with a high cost of computational resources. In order to detour such difficulties, Huber et al. (2004) suggest an approximation of the marginal likelihood using Laplace approximation. However, their estimating equations do not give closed-form solutions so iterative method (e.g., quasi-Newton procedure or fixed-point algorithm) has to be used to solve the implicit equations they proposed in every iteration step. In Chapter III, we use variational method for the marginal likelihood approximation, which was introduced by Jaakkola and Jordan (1997) in a Bayesian logistic regression model.

Another approach which can avoid the intractable integration is to treat the principal component scores α_{il} ($i = 1, \cdots, n;\ l = 1, \cdots, k$) as fixed parameters in the model, which was studied by several researchers. Collins et al. (2001) suggested a generalization of principal components analysis to the exponential family distribution where the Bregman loss function is minimized to obtain the low rank representation of the canonical parameters in the exponential family distribution. Schein et al. (2003) proposed a logistic PCA in the similar way with Collins et al. (2001) but they maximized an auxiliary function in order to derive the alternating least square updates for model parameters. This approach was also used for PCA of binary data in de Leeuw (2006) in the name of Majorization or MM algorithm with more compact and rigorous treatments. This approach is studied in Chapter II.

Both of approaches, fixed or random principal component scores, binary principal components analysis methods suffer from lots of non-zero principal component loadings as the standard principal components analysis. since we see that the minimization criterion in (1.2) can be regarded as the maximum Gaussian likelihood estimation. This can be also interpreted as the least square estimations when the principal component scores are given. Thus, we may introduce the sparsity-inducing penalty, for instance L_1 penalty, on the principal components, which leads to LASSO solution. This can be viewed as the penalized likelihood estimation when we consider the minimization of the sum of squares

of reconstruction errors is equivalent to maximization of a Gaussian likelihood.

In the following, we will review two bound optimization algorithms, called MM algorithm and variational method, which will be extensively exploited in the whole study.

1.3 Review of Estimation Procedures

1.3.1 MM algorithm

In this section, we briefly review an optimization method which will be used in Chapter II, called the MM algorithm. The MM algorithm relies on convexity arguments and is particularly useful in high-dimensional problem such as image reconstruction (Lange et al., 2000; Hunter and Lange, 2004). This acronym does double duty. In minimization problems, the first M of MM stands for majorize and the second M for minimize. In maximization problems, the first M stands for minimize and the second M for maximize. When it is successful, the MM algorithm substitutes a simple optimization problem for a difficult optimization problem. In simplifying the original problem, we must pay the price of iteration or iteration with a slower rate of convergence. The well-known EM algorithm is a special case of the MM algorithm which does not necessarily involves around notions of missing data.

A function $g(x|x^{(m)})$ is said to majorize a function $f(x)$ at $x^{(m)}$ when g satisfies

$$\begin{aligned} f(x^{(m)}) &= g(x^{(m)}|x^{(m)}) \\ f(x) &\leq g(x|x^{(m)}). \end{aligned} \tag{1.3}$$

In other words, the function surface $x \mapsto g(x|x^{(m)})$ lies above the surface $f(x)$ and is tangent to it at the point $x = x^{(m)}$. In the iterative algorithm, $x^{(m)}$ represents the current iterate in a search of the surface $f(x)$. Figure 2 provides a simple one-dimensional example.

In the minimization version of the MM algorithm, we minimize the surrogate majorizing function $g(x|x^{(m)})$ rather than the actual function $f(x)$. If $x^{(m+1)}$ denotes the minimum

of the surrogate $g(x|x^{(m)})$, then we can show that the MM procedure forces $f(x)$ downhill. Indeed, the inequality

$$\begin{aligned} f(x^{(m+1)}) &= g(x^{(m+1)}|x^{(m)}) + f(x^{(m+1)}) - g(x^{(m+1)}|x^{(m)}) \\ &\leq g(x^{(m)}|x^{(m)}) + f(x^{(m)}) - g(x^{(m)}|x^{(m)}) \\ &= f(x^{(m)}) \end{aligned}$$

follows directly from the fact $g(x^{(m+1)}|x^{(m)}) \leq g(x^{(m)}|x^{(m)})$ and definition (1.3). Or such driving force on the MM algorithm can be seen by looking at

$$g(x^{(m+1)}|x^{(m)}) - g(x^{(m)}|x^{(m)}) \ \geq \ f(x^{(m+1)}) - f(x^{(m)})$$

which can be verified from (1.3) easily. In other words, any decrease in the value of $g(x|x^{(m)})$ guarantees a decrease in the value of the actual function $f(x)$. For implementation of the MM algorithm, therefore, finding a majorizing function which is easy to be optimized is a crucial step determining usefulness of the MM algorithm.

In order to help understanding, consider a simple one-dimensional example that finds the median of data $x_1, \cdots, x_n$. It is well known that finding minimum of the function $f(x) = \sum_{i=1}^{n} |x - x_i|$ leads to median. However, minimizing $f(x)$ is not analytical to solve because it is piecewise linear. This function is illustrated in Figure 2 with a small dataset comprising 1, 3, 4, 8 and 10, which gives the median as 4. Using the relation

$$|x| \ \leq \ \frac{x^2 + y^2}{2|y|},$$

the original function $f(x)$ has a quadratic majorizing function at $x^{(m)}$ as

$$f(x) \ \leq \ \sum_{i=1}^{n} \frac{(x - x_i)^2 + (x^{(m)} - x_i)^2}{2|x^{(m)} - x_i|},$$

which is depicted in Figure 2 at the tangent point $x^{(m)} = 6$. This technique, finding a quadratic majorizing function of the absolute value function, will be used in finding a

quadratic upper bound of L_1 penalty function in Chapter II. For the binary principal components analysis, we will find a quadratic majorizing function of the negative log of inverse logit function in order to exploit the MM algorithm.

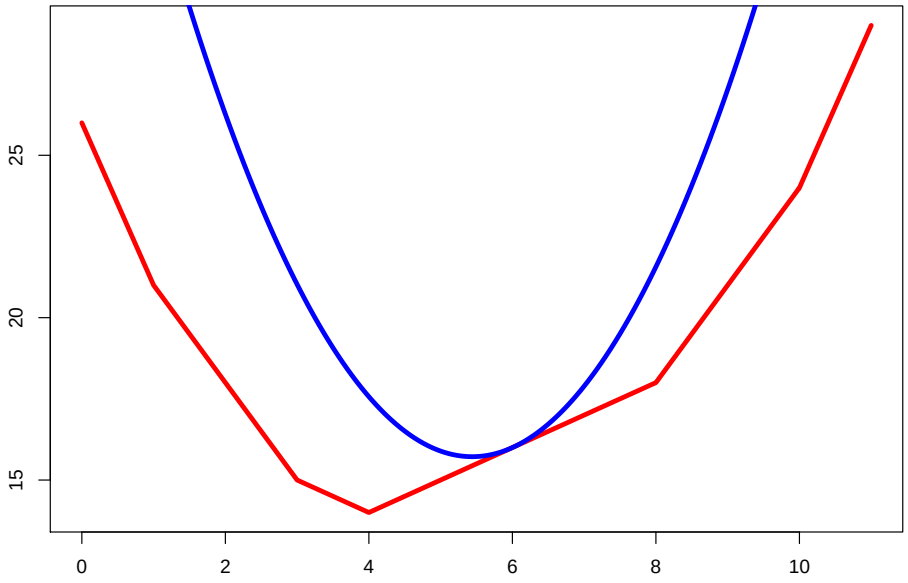


Figure 2: The piecewise linear function $f(x) = |x-1|+|x-3|+|x-4|+|x-8|+|x-10|$ is shown in red line and its quadratic majorizing function at the tangent point $x^{(m)} = 6$ is drawn in blue.

1.3.2 Variational method

Variational methods have their origins in the 18^{th} century with the work of Euler, Lagrange, and others on the calculus of variations. Standard calculus is concerned with finding derivatives of functions. They are a family of techniques for approximating intractable integrals arising in Bayesian statistics and machine learning. They can be used to find a lower bound for the marginal likelihood of several models with a view to performing model selection, and often provide an analytical approximation to the parameter posterior probability which is useful for prediction. It is an alternative to Monte Carlo sampling methods for making use of a posterior distribution that is difficult to sample from directly. There are huge lit-

erature on this topic which can be found in Jordan (1999), Bishop (2006) and references therein.

Such variational methods find a 'global' solution in the sense that it directly seeks an approximation to the full posterior distribution over all random variables. In this study, we use an alternative 'local' approach which involves finding bounds on functions over individual variables or groups of variables within a model. The purpose of introducing the bound is to simplify the resulting distribution.

It is instructive to illustrate variational method considering a simple example, the function $f(x) = \exp(-x)$, which is a convex function of x, and which is shown in the left penal of Figure 3. Our goal is to approximate $f(x)$ by a simpler function, in particular a linear function of x. From Figure 3, we see that this linear function will be a lower bound on $f(x)$ if it corresponds to a tangent. We can obtain the tangent line $y(x)$ at a specific value of x, say $x = \xi$, by making a first order Taylor expansion

$$y(x) \;=\; f(\xi) + f'(\xi)(x - \xi)$$

so that $y(x) \leq f(x)$ with equality when $x = \xi$. For our example function $f(x) = \exp(-x)$, we therefore obtain the tangent line in the form

$$y(x, \xi) \;=\; \exp(-\xi) - \exp(-\xi)(x - \xi)$$

which is a linear function parametrized by ξ. For consistency with subsequent discussion, let us define $\lambda = -\exp(-\xi)$ so that

$$y(x, \lambda) \;=\; \lambda x - \lambda + \lambda \ln(-\lambda).$$

Different values of λ correspond to different tangent lines, and because all such lines are lower bounds on the function, we have $f(x) \geq y(x, \lambda)$. Thus we can write the function in the form

$$f(x) \;=\; \max_{\lambda} \{\lambda x - \lambda + \lambda \ln(-\lambda)\}. \tag{1.4}$$

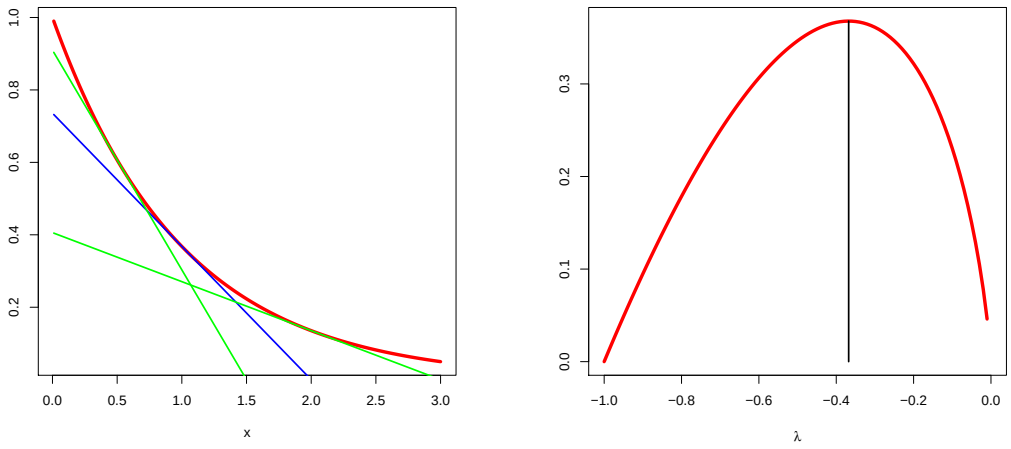


Figure 3: In the left panel, red curve shows the function $\exp(-x)$, and the blue line shows the tangent at $x = \xi$ with $\xi = 1$. This line has slope $\lambda = f'(\xi) = -\exp(-\xi)$. Note that any other tangent line, for example the ones shown in green, will have a smaller value of y at $x = \xi$. The right panel shows the corresponding plot of the function $\lambda\xi - g(\lambda)$ versus λ for $\xi = 1$, in which the maximum corresponds to $\lambda = -\exp(-\xi) = -1/e$.

We have succeeded in approximating the convex function $f(x)$ by a simpler, linear function $y(x, \lambda)$. The price we have to pay is that we have introduced a variational parameter λ, and to obtain the tightest bound we must optimize with respect to λ.

We can formulate this approach more generally using the framework of convex duality (Rockafella, 1972; Jordan et al., 1999). Consider the illustration of a convex function $f(x)$ shown in he left panel in Figure 4. In this example, the function λx is a lower bound on $f(x)$ but it is not the best lower bound that can be achieved by a linear function having slope λ, because the tightest bound is given by the tangent line. Let us write the equation of the tangent line, having slope λ as $\lambda x - g(\lambda)$ where the (negative) intercept $g(\lambda)$ clearly depends on the slope λ of the tangent. To determine the intercept, we note that the line must be moved vertically by an amount equal to the smallest vertical distance between the

line and the function, as shown in Figure 4. Thus,

$$\begin{aligned} g(\lambda) &= -\min_x\{f(x) - \lambda x\} \\ &= \max_x\{\lambda x - f(x)\}. \end{aligned} \tag{1.5}$$

Now, instead of fixing λ and varying x, we can consider a particular x and then adjust λ until the tangent plane is tangent at that particular x. Because the y value of the tangent line at a particular x is maximized when that value coincides with its contact point, we have

$$f(x) = \max_\lambda\{\lambda x - g(\lambda)\}. \tag{1.6}$$

We see that the function $f(x)$ and $g(\lambda)$ play a dual role, and are related through (1.5) and (1.6).

Let us apply these duality relations to our example $f(x) = \exp(-x)$. From (1.4) we see that the maximizing value of x is given by $\xi = -\ln(-\lambda)$, and back-substituting we obtain the conjugate function $g(\lambda)$ in the form

$$g(\lambda) = \lambda - \lambda\ln(-\lambda) \tag{1.7}$$

as obtained previously. The function $\lambda\xi - g(\lambda)$ is shown, for $\xi = 1$ in the right panel in Figure 3. As a check, we can substitute (1.7) into (1.6), which gives the maximizing value of $\lambda = -\exp(-x)$, and back-substituting then recovers the original function $f(x) = \exp(-x)$.

If the function of interest is not convex, then we cannot directly apply the method above to obtain a bound. However, we can first seek invertible transformations either of the function or of its argument which change it into a convex form. We then calculate the conjugate function and then transform back to the original variables.

An important example, which arises in our study in Chapter III, is the inverse logit function defined by

$$\pi(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$$

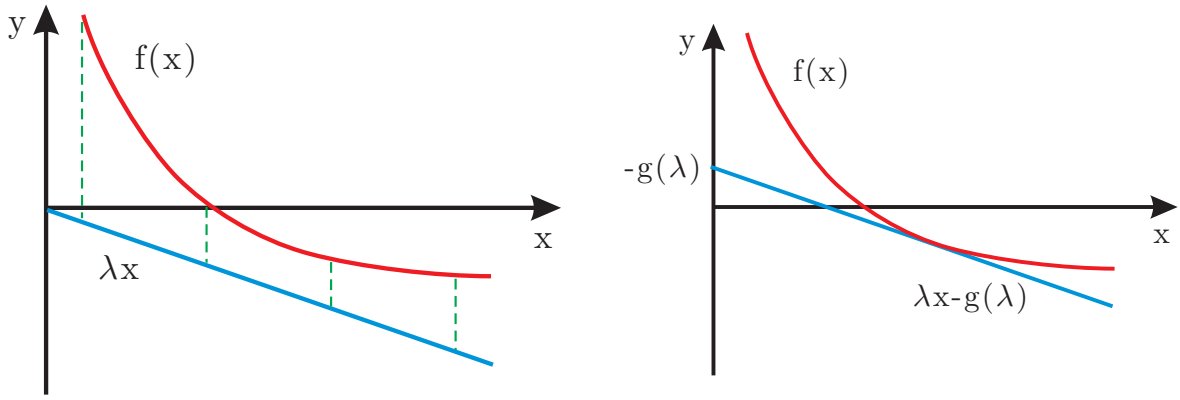


Figure 4: In the left panel the red curve shows a convex function $f(x)$, and the blue line represents the linear function λx, which is a lower bound on $f(x)$ because $f(x) > \lambda x$ for all x. For the given value of slope λ the contact point of the tangent line having the same slope is found by minimizing with respect to x the discrepancy (shown by the green dashed lines) given by $f(x) - \lambda x$. This defines the dual function $g(\lambda)$, which corresponds to the (negative of the) intercept of the tangent line having slope λ.

which will be used in latent variable model for binary principal components analysis. We can obtain a quadratic lower bound on it having the functional form of a normal distribution. This was introduced and studied in Jaakkola and Jordan (2000). First we consider

$$f(x) \;=\; \log \pi(x) - \frac{x}{2}.$$

Note that the function $f(x)$ is a convex function in terms of x^2, as can be verified by finding the second derivative. This leads to a lower bound on $f(x)$, which is a linear function of x^2 whose conjugate function is given by

$$g(\lambda) \;=\; \max_{x^2}\{\lambda x^2 - f(\sqrt{x^2})\}$$

from (1.5). The stationary condition leads to

$$0 \;=\; \lambda - \frac{dx}{dx^2}\frac{d}{dx}f(x) \;=\; \lambda - \frac{1-2\pi(x)}{4x}.$$

If we denote this value of x, corresponding to the contact point of the tangent line for this particular value of λ, by ξ, then we have

$$\lambda(\xi) \;=\; \frac{1-2\pi(\xi)}{4\xi}. \tag{1.8}$$

Instead of thinking of λ as the variational parameter, we can let ξ play this role since this leads to simpler expressions for the conjugate function, which is then given by

$$g(\lambda) \;=\; \lambda(\xi)\xi^2 - f(\xi).$$

Thus, from (1.6), the bound on $f(x)$ can be written as

$$f(x) \;\geq\; \lambda x^2 - g(\lambda) \;=\; \lambda x^2 - \lambda\xi^2 + f(\xi).$$

The lower bound of the inverse logit function, therefore, is

$$\pi(x) \;\geq\; \pi(\xi)\exp\{(x-\xi)/2 + \lambda(\xi)(x^2-\xi^2)\} \tag{1.9}$$

where $\lambda(\xi)$ is defined in (1.8). This bound is illustrated in Figure 5. We see that the bound has the form of the exponential of a quadratic function of x, which will prove useful when we seek Gaussian representation of the conditional distribution defined through the inverse logit function in Chapter III.

1.4 Overview of Dissertation

The goal of this study is to develop the generalization of principal components analysis for binary data with special efforts paid on the simple structure of principal components. Especially, our method which will be described in next sections is the model-based approach where we will propose two different formulations, each of which is dealt separately

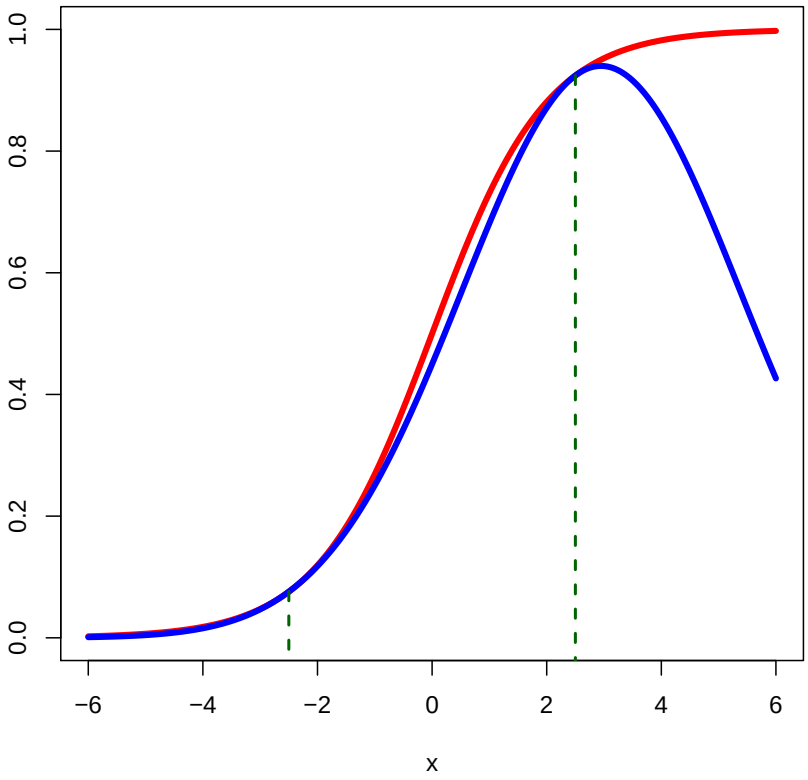


Figure 5: This figure shows the inverse logit function in red together with the Gaussian lower bound (1.9) shown in blue. Here the parameter $\xi = 2.5$, and the bound is exact at $x = \xi$ and $x = -\xi$, denoted by the dashed green lines.

in different section, as a sole article. In Chapter II, we present the sparse binary principal components analysis by regarding principal component scores as fixed parameters. A stable estimation procedure is introduced by using MM algorithm. And we deal with the principal component scores as random variables and we provide the approximation of the marginal likelihood and its estimation procedure by using variational method, where we also suggest a unified algorithm for principal components analysis for the data comprising disparate variables, including binomial and normal variables as well as binary. In both of two ways of generalization, we give a model selection procedure and missing data treat-

ment coherently with the proposed algorithm.

CHAPTER II

SPARSE PRINCIPAL COMPONENTS ANALYSIS FOR BINARY DATA

In this chapter, we develop a new PCA type dimension reduction method for binary data. Different from the standard PCA which is directly defined on the observed data, our new PCA is defined indirectly on the logit scale of the success probabilities of the binary observations. We also introduce sparsity to the principal component (PC) loading vectors for enhanced interpretability and more stable extraction of the principal components. Our sparse PCA is formulated as solving an optimization problem with a criterion function motivated from penalized Bernoulli likelihood. We develop a Majorization-Minimization algorithm to efficiently solve the optimization problem. The effectiveness of our sparse PCA method is illustrated using a simulation study and three real data examples.

2.1 Introduction

Principal components analysis (PCA) is a widely used method for dimensionality reduction, feature extraction and visualization of multivariate data. Several sparse PCA methods have recently been introduced to improve the standard PCA (e.g., Jolliffe et al., 2003; Zou et al., 2006; Shen and Huang, 2008). By requiring the principal component loading vectors to be sparse, sparse PCA methods yield PCs that are more easily interpretable. Sparsity also regularizes the extraction of PCs and thus makes the extraction more stable. Such stability is more beneficial when the dimension is high, especially in the so-called high-dimension low-sample-size settings. As extensions of the standard PCA, however, these sparse PCA methods are mostly suitable to variables of continuous type, they are not generally appropriate for other data types such as binary data or counts. The goal of this chapter is to develop a sparse PCA method for binary data.

There are two commonly used definitions of PCA that give rise to the same result. PCA can be defined as the orthogonal projection of the data onto a low dimensional linear subspace, known as the principal space, such that the variance of the projected data is maximized (Hotelling, 1933). Equivalently, PCA can be defined as the linear projection that minimizes the mean squared distance between the data points and their projections (Pearson, 1901). Shen and Huang (2008) developed their sparse PCA method following the viewpoint of Pearson. For binary variables, one may follow these two directions selectively. As along the Hotelling's direction, the standard PCA is often applied to the binary data directly for the descriptive purpose. However, the direct application of the standard PCA to binary variables is not satisfactory nor desirable in the sense that the covariance matrix of the observed data has especial relevance for continuous type variables and the linear functions of binary variables are less readily interpretable. Some interesting variants of this approach to binary variables can be found in Jolliffe (2004).

For Pearson's approach, it is instructive to consider its geometrical interpretation. Suppose $\mathbf{y}_1, \cdots, \mathbf{y}_n \in \mathbb{R}^d$ are the n data points and consider a k-dimensional $(k < d)$ linear manifold spanned by an orthogonal bases $\tilde{\mathbf{b}}_1, \cdots, \tilde{\mathbf{b}}_k$ with a shift vector $\boldsymbol{\mu}$. According to Pearson, the PCA minimizes the following reconstruction error

$$\sum_{i=1}^{n} ||\mathbf{y}_i - (\boldsymbol{\mu} + a_{i1}\tilde{\mathbf{b}}_1 + \cdots + a_{iK}\tilde{\mathbf{b}}_k)||^2. \tag{2.1}$$

This is a least squares regression if a_{ik}'s were known. In light of this connection to regression and borrowing idea from LASSO (Tibshirani, 1996), Shen and Huang (2008) proposed to add a L_1 penalty $||\tilde{\mathbf{b}}_1||_1 + \cdots + ||\tilde{\mathbf{b}}_k||_1$ to the reconstruction error (2.1) to obtain sparse loading vectors $\tilde{\mathbf{b}}_1, \cdots, \tilde{\mathbf{b}}_k$. Since the reconstruction error (2.1) can be viewed as the negative log likelihood up to a constant for the Gaussian distributions with mean vectors $\boldsymbol{\theta}_i = \boldsymbol{\mu} + a_{i1}\tilde{\mathbf{b}}_1 + \cdots + a_{ik}\tilde{\mathbf{b}}_k$ for $i = 1, \cdots, n$ and identity covariance, the method of Shen and Huang (2008) can be interpreted as a penalized likelihood approach for sparse PCA.

The key idea of the current chapter is to replace the Gaussian likelihood by the Bernoulli likelihood. The relationship of the proposed sparse PCA for binary data to the sparse PCA of Shen and Huang (2008) is analogous to the relationship between logistic and linear LASSO regression. We thus will refer to the proposed PCA method as sparse logistic PCA.

We develop an iterative weighted least squares algorithm to perform the proposed sparse logistic PCA. Since the log likelihood is not quadratic and the penalty function is non-differentiable, the optimization problem for the sparse logistic PCA is not straightforward to solve. Our algorithm applies the general idea of optimization transfer or Majorization-Minimization (MM) algorithm (Lange et al., 2000; Hunter and Lange, 2004). By iteratively replacing the complex objective function with suitably defined quadratic surrogates, each step of our algorithm solves a weighted least squares problem and has closed form. The algorithm is easy to implement and guaranteed at each iteration to improve the penalized PCA log-likelihood. We show that the same MM algorithm is applicable when there are missing data. We also develop a method for choosing the penalty parameters and for choosing the number of important principal components. PCA of binary data using Bernoulli likelihood has previously been studied by Collins et al. (2001), Schein et al. (2003) and de Leeuw (2006), but none of these works considered sparse loading vectors. As we demonstrate using simulation and real data, sparsity can enhance interpretation of results and improve the stability and accuracy of the extracted principal components.

Other approaches of sparse PCA are not as easily extendible to binary data. Jolliffe et al. (2003) modified the defining maximum variance problem of the standard PCA by applying a L_1-norm constraint on the PC loading vectors to obtain PCA with sparse loadings. Its use of sample variance makes it unappealing for binary data. Zou et al. (2006) rewrote PCA as a regression-type optimization problem and then applied the LASSO penalty (Tibshirani, 1996) to obtain sparse loadings. However, since the data appear both as regressors and responses in their regression-type problem, the connection of their approach to penal-

ized likelihood is not as natural as Shen and Huang (2008).

The rest of this chapter is organized as follows. In Section 2, we introduce the optimization problem that yields the sparse PCA for binary data and also provide an efficient Majorization-Minimization algorithm for computation. Section 3 addresses the important issue of tuning parameter selection. Section 4 discusses how to handle missing data. The proposed methodology is illustrated by using a simulation study in Section 5 and using three real data sets in Section 6.

2.2 Sparse Logistic PCA with Penalized Likelihood

2.2.1 Model setup

Consider the $n \times d$ binary data matrix $\mathbf{Y} = (y_{ij})$ each row of which represents a vector of observations from binary variables. We assume that entries of $\mathbf{Y}$ are realizations of mutually independent random variables and that y_{ij} follows the Bernoulli distribution with success probability π_{ij}. Let $\theta_{ij} = \log\{\pi_{ij}/(1-\pi_{ij})\}$ be the logit transformation of π_{ij}. Then the individual data generating probability becomes

$$\Pr(Y_{ij} = y_{ij}) \;=\; \pi(\theta_{ij})^{y_{ij}}\{1-\pi(\theta_{ij})\}^{1-y_{ij}} \;=\; \pi(q_{ij}\theta_{ij})$$

with $q_{ij} = 2y_{ij} - 1$ since $\pi(-\theta) = 1 - \pi(\theta)$. This representation leads to the compact form of the log likelihood as

$$\ell \;=\; \sum_{i=1}^{n}\sum_{j=1}^{d} \log \pi(q_{ij}\theta_{ij}).$$

Note that the Bernoulli distributions are in the exponential family and θ_{ij} are the corresponding canonical parameters.

To build a probabilistic model for principal components analysis of binary data, the d-dimensional canonical parameter vectors $\boldsymbol{\theta}_i = (\theta_{i1}, \cdots, \theta_{id})^T$ are constrained to reside in the low dimensional manifold of $\mathbb{R}^d$ with the dimensionality k. (The choice of k will

be discussed later in Section 2.4.2.) Specifically, we assume that, for some vectors $\boldsymbol{\mu}$, $\tilde{\mathbf{b}}_1, \cdots, \tilde{\mathbf{b}}_k \in \mathbb{R}^d$, the vector of canonical parameters satisfies $\boldsymbol{\theta}_i = \boldsymbol{\mu} + a_{i1}\tilde{\mathbf{b}}_1 + \cdots + a_{ik}\tilde{\mathbf{b}}_k$ for $i = 1, \ldots, n$. We call $\tilde{\mathbf{b}}_1, \cdots, \tilde{\mathbf{b}}_k$ the principal component loading vectors and the coefficients $\mathbf{a}_i = (a_{i1}, \cdots, a_{ik})^T$ the principal component scores (PC scores) for the ith observation. Geometrically, the vectors of canonical parameters $\boldsymbol{\theta}_i$ are projected onto the k-dimensional manifold which is the affine subspace spanned by k PC loading vectors and translated by the intercept vector $\boldsymbol{\mu}$. In matrix form, the canonical parameter matrix $\boldsymbol{\Theta} = \big(\theta_{ij}\big)_{\substack{i=1,\cdots,n \\ j=1,\cdots,d}} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_n)^T$ is represented as

$$\boldsymbol{\Theta} = \mathbf{1}_n \otimes \boldsymbol{\mu}^T + \mathbf{A}\mathbf{B}^T \tag{2.2}$$

where $\mathbf{A} = (\mathbf{a}_1, \cdots, \mathbf{a}_n)^T$ is the $n \times k$ principal component score matrix and $\mathbf{B} = (\tilde{\mathbf{b}}_1, \cdots, \tilde{\mathbf{b}}_k)$ is the $d \times k$ principal component loading matrix. The notation $\otimes$ denotes the Kronecker product.

The factorization of the rank k matrix $\boldsymbol{\Theta}_0 \triangleq \mathbf{A}\mathbf{B}^T$ in (2.2) is not unique, since for any $k \times k$ orthogonal matrix $\mathbf{H}$, $\mathbf{A}\mathbf{B}^T = \mathbf{A}^*\mathbf{B}^{*T}$ for $\mathbf{A}^* = \mathbf{A}\mathbf{H}$ and $\mathbf{B}^* = \mathbf{B}\mathbf{H}$. To make the factorization unique, we perform the singular value decomposition $\boldsymbol{\Theta}_0 = \mathbf{U}\mathbf{D}\mathbf{V}^T$ where $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns and $\mathbf{D}$ is diagonal, and then let $\mathbf{A} = \mathbf{U}$ and $\mathbf{B} = \mathbf{V}\mathbf{D}$. This procedure makes the model unique up to the sign change, which does not have a practical importance in the interpretation.

We target a method that can produce a sparse loading matrix, a loading matrix with many zero elements. A sparse loading matrix implies variable selection in principal components analysis, since each principal component only involves those variables corresponding to the nonzero elements of the loading vector. Variable selection using L_1 penalty has been widely used for regression type of problems since the introduction of LASSO by Tibshirani (1996). Let $\mathbf{b}_j^T$ denote the jth row of $\mathbf{B}$. Then (2.2) implies that $\theta_{ij} = \mu_j + \mathbf{a}_i^T\mathbf{b}_j$ where μ_j

is the jth element of $\boldsymbol{\mu}$. The log likelihood can be written as

$$\ell(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) \;=\; \sum_{j=1}^{d}\sum_{i=1}^{n} \log \pi\{q_{ij}(\mu_j + \mathbf{a}_i^T \mathbf{b}_j)\}. \tag{2.3}$$

If $\mathbf{a}_i$ were observable, (2.3) is the log likelihood for d logistic regressions

$$\text{logit} P(Y_{ij} = 1) = \mu_j + \mathbf{a}_i^T \mathbf{b}_j.$$

This connection with logistic regression suggests use of the L_1 penalty to get a sparse loading matrix, as in LASSO regression.

Specifically, consider the penalty

$$P_{\boldsymbol{\lambda}}(\mathbf{B}) \;=\; \sum_{l=1}^{k} \lambda_l ||\tilde{\mathbf{b}}_l||_1 \;=\; \lambda_1 \sum_{j=1}^{d} |b_{j1}| + \cdots + \lambda_k \sum_{j=1}^{d} |b_{jk}|,$$

where λ_l are regularization parameters whose selection will be discussed later. We generate sparse principal components by maximizing the following penalized log likelihood

$$f(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) \;=\; \ell(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) - nP_{\boldsymbol{\lambda}}(\mathbf{B}).$$

Equivalently, we minimize the following criterion function

$$S(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) \;=\; -\ell(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) + nP_{\boldsymbol{\lambda}}(\mathbf{B}), \tag{2.4}$$

where the negative log likelihood can be interpreted as a loss function and the L_1 penalties increase the loss for nonzero elements of $\mathbf{B}$ according to their magnitude. This penalized loss interpretation is also appealing in the sense that the independent Bernoulli trials assumption for obtaining the likelihood (2.3) need not be a realistic representation of actual data generating process but rather a device for generating a suitable loss function. We shall focus on the minimization problem (2.4) for the rest of this chapter.

2.2.2 Majorization-Minimization algorithm

We develop a majorization-minimization (MM) algorithm for minimizing (2.4), which iteratively minimizes a suitably defined quadratic upper bound of (2.4). Instead of directly dealing with the non-quadratic log likelihood and the non-differentiable sparsity inducing L_1 penalty, the MM algorithm sequentially optimizes a quadratic surrogate objective function. A function $g(x|y)$ is said to majorize a function $f(x)$ at y if

$$g(x|y) \geq f(x) \quad \text{for all } x \qquad \text{and} \qquad g(y|y) = f(y).$$

In the geometrical view, the function surface $g(x|y)$ lies above the function $f(x)$ and is tangent to it at the point y so $g(x|y)$ becomes an upper bound of $f(x)$. To minimize $f(x)$, the MM algorithm starts from an initial guess $x^{(0)}$ of x, and iteratively minimizes $g(x|x^{(m)})$ until convergence, where $x^{(m)}$ is the estimate of x at the mth iteration. The MM algorithm decreases the objective function in each step and is guaranteed to converge to a local minimum of $f(x)$. In application of the MM-algorithm, the majorizing function $g(x|y)$ is chosen to be easier to minimize than the original objective function $f(x)$. See Hunter and Lange (2004) for an introductory description of the MM algorithm.

To find a suitable majorizing function of (2.4), we treat the log likelihood term and the penalty term separately. For the log likelihood term, note that, for a given point y,

$$-\log \pi(x) \leq -\log \pi(y) - \{1-\pi(y)\}(x-y) + \tfrac{2\pi(y)-1}{4y}(x-y)^2 \tag{2.5}$$

$$\leq -\log \pi(y) - \{1-\pi(y)\}(x-y) + \tfrac{1}{8}(x-y)^2, \tag{2.6}$$

and the equalities hold when $x = y$ (Jaakkola and Jordan, 2000; de Leeuw, 2006). These inequalities provide quadratic upper bounds for the negative log inverse logit function at the tangent point y. We refer to the former bound as the tight bound, and the latter bound as the uniform bound since its curvature does not change with y. To show the above inequality relations, first we will prove the following lemmas:

LEMMA **II.1**. *The function* $\pi(x)\{1-\pi(x)\}$ *is decreasing in* $x \geq 0$ *where* $\pi(x) = \{1+\exp(-x)\}^{-1}$.

Proof. The first derivative is $\pi'(x)\{1-\pi(x)\}-\pi(x)\pi'(x) = \pi'(x)\{1-2\pi(x)\} = \pi(x)\{1-\pi(x)\}\{1-2\pi(x)\}$. By observing $1/2 \leq \pi(x) \leq 1$ on $x \geq 0$, the derivative is negative. $\diamondsuit$

LEMMA **II.2**. *The function* $r(x) = \log \pi(\sqrt{x}) - \sqrt{x}/2$ *is convex.*

Proof. The second derivative of $r(x)$ is given as

$$r''(x) = \frac{1}{4x}\left[\frac{2\pi(\sqrt{x})-1}{2\sqrt{x}} - \pi(\sqrt{x})\{1-\pi(\sqrt{x})\}\right].$$

Note that $\{2\pi(\sqrt{x})-1\}/2\sqrt{x} = \{\pi(\sqrt{x})-\pi(-\sqrt{x})\}/2\sqrt{x} = \pi'(\xi) = \pi(x)\{1-\pi(\xi)\}$ with $\xi \in (-\sqrt{x}, \sqrt{x})$ from the mean value theorem. From $\xi < \sqrt{x}$ and Lemma II.1, the second derivative of $r(x)$ is positive, which completes the proof. $\diamondsuit$

Thus, from the convexity of function $r(x)$, we get $r(x) \geq r(y) + r'(y)(x-y)$ at any y, so that

$$\log \pi(\sqrt{x}) - \frac{\sqrt{x}}{2} \;\geq\; \log \pi(\sqrt{y}) - \frac{\sqrt{y}}{2} + \frac{1-2\pi(\sqrt{x})}{4\sqrt{y}}(x-y)$$

$$\Rightarrow \quad -\log \pi(\sqrt{x}) \;\leq\; -\log \pi(\sqrt{y}) - \frac{\sqrt{x}-\sqrt{y}}{2} + \frac{2\pi(\sqrt{x})-1}{4\sqrt{y}}(x-y)$$

and by changing variables $\sqrt{x}$ by x we obtain (2.5). The curvature of the tight bound function becomes

$$\frac{2\pi(y)-1}{4y} \;=\; \frac{\pi(y)-\pi(-y)}{4y} \;=\; \frac{2y\pi'(y)}{4y} \;=\; \frac{1}{2}\pi(\xi)\{1-\pi(\xi)\} \;\leq\; \frac{1}{8}$$

by the mean value theorem and $\xi \in (-y, y)$. This completes to prove the inequality (2.6). At $y = 0$, the curvature of the tight bound is not defined properly. In such case, it takes its limit when y approaches zero. By L'hopital's theorem we get

$$\lim_{y\to 0}\frac{2\pi(y)-1}{4y} \;=\; \lim_{y\to 0}\frac{2\pi'(y)}{4} \;=\; \lim_{y\to 0}\frac{\pi(y)\{1-\pi(y)\}}{2} \;=\; \frac{1}{8}.$$

For the penalty term, the inequality

$$|x| \leq \frac{x^2 + y^2}{2|y|}, \qquad y \neq 0, \tag{2.7}$$

gives an upper bound for $|x|$ and the equality holds when $x = y$ (Hunter and Li, 2005). Application of (2.5), (2.6), and (2.7) yields a suitable majorizing function of (2.4) and an MM algorithm, as stated below in Theorem II.1.

To present details of the MM algorithm, we introduce some notations. Let $\mathbf{\Theta}^{(m)}$ be the estimate of $\mathbf{\Theta}$ obtained in the mth step of the algorithm, with the entries $\theta_{ij}^{(m)} = \mu_j^{(m)} + \mathbf{a}_i^{(m)T}\mathbf{b}_j^{(m)}$. Define

$$x_{ij}^{(m)} = \begin{cases} \frac{\theta_{ij}^{(m)}}{2\pi(q_{ij}\theta_{ij}^{(m)})-1} & \text{for tight bound,} \\ \theta_{ij}^{(m)} + 4q_{ij}\{1 - \pi(q_{ij}\theta_{ij}^{(m)})\} & \text{for uniform bound,} \end{cases} \tag{2.8}$$

and

$$w_{ij}^{(m)} = \begin{cases} \frac{2\pi(\theta_{ij}^{(m)})-1}{4\theta_{ij}^{(m)}} & \text{for tight bound,} \\ \frac{1}{8} & \text{for uniform bound.} \end{cases} \tag{2.9}$$

In both definitions, the superscript m indicates the dependence on $\mathbf{\Theta}^{(m)}$. For the tight bound case, $x_{ij}^{(m)}$ and $w_{ij}^{(m)}$ are not well defined when $\theta_{ij}^{(m)} = 0$ and will be replaced by the limit of the corresponding quantities when $\theta_{ij}^{(m)} \to 0$. To be specific, applying

$$\lim_{\theta \to 0} \frac{2\pi(\theta) - 1}{\theta} = \frac{1}{2},$$

we define

$$\begin{aligned} x_{ij}^{(m)} &= \lim_{\theta_{ij}^{(m)} \to 0} \frac{\theta_{ij}^{(m)}}{2\pi(q_{ij}\theta_{ij}^{(m)}) - 1} = \frac{2}{q_{ij}}, \\ w_{ij}^{(m)} &= \lim_{\theta_{ij}^{(m)} \to 0} \frac{2\pi(\theta_{ij}^{(m)}) - 1}{4\theta_{ij}^{(m)}} = \frac{1}{8} \end{aligned}$$

when $\theta_{ij}^{(m)} = 0$. The working variable z's in the uniform bound can be seen as the first-order Taylor approximation to those of the standard iterative reweighted least squares (IRLS) algorithm for the generalized linear models (GLMs) with Bernoulli distribution. In such case, the working variable z has the form of

$$z_{ij} \;=\; \theta_{ij} + (y_{ij} - \pi_{ij}) \cdot \frac{1}{\pi_{ij}(1-\pi_{ij})}$$

with $\pi_{ij} = \pi(\theta_{ij})$. The last term is approximated by $4q_{ij}(1 - \pi(q_{ij}\theta_{ij}))$ when we apply the Taylor's expansion to it at $\pi_{ij} = 1/2$.

Now, let

$$\begin{aligned} &g(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B} | \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)}) \\ &\quad = \sum_{i=1}^{n} \sum_{j=1}^{d} \Big[w_{ij}^{(m)} \big\{ x_{ij}^{(m)} - (\mu_j + \mathbf{a}_i^T \mathbf{b}_j) \big\}^2 + \mathbf{b}_j^T \mathbf{D}_{\boldsymbol{\lambda},j}^{(m)} \mathbf{b}_j \Big], \end{aligned} \tag{2.10}$$

where $\mathbf{D}_{\boldsymbol{\lambda},j}^{(m)}$ are diagonal matrices with diagonal elements $\lambda_l/2|b_{jl}^{(m)}|$ for $l = 1, \cdots, k$.

THEOREM **II.1**. (i) *Up to a constant that depends on* $\boldsymbol{\mu}^{(m)}$, $\mathbf{A}^{(m)}$, *and* $\mathbf{B}^{(m)}$ *but not on* $\boldsymbol{\mu}$, $\mathbf{A}$, *and* $\mathbf{B}$, *the function* $g(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B} | \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$ *defined in* (2.10) *majorizes* $S(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$ *at* $(\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$.

(ii) *Let* $(\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$, $m = 1, 2, \ldots$, *be a sequence obtained by iteratively minimizing the majorizing function. Then* $S(\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$ *increases with* m *and it converges to a local minimum of* $S(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$ *as* m *goes to infinity.*

Proof. Applications of (2.5) and (2.6) yield the following majorizing functions of the negative log likelihood $-\ell(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$:

$$\sum_{i=1}^{n} \sum_{j=1}^{d} \Big[-\log \pi(q_{ij}\theta_{ij}^{(m)}) - q_{ij}\{1 - \pi(q_{ij}\theta_{ij}^{(m)})\}(\theta - \theta_{ij}^{(m)}) + \tfrac{2\pi(q_{ij}\theta_{ij}^{(m)})-1}{4q_{ij}\theta_{ij}^{(m)}}(\theta - \theta_{ij}^{(m)})^2 \Big]$$

for the tight bound, and

$$\sum_{i=1}^{n} \sum_{j=1}^{d} \Big[-\log \pi(q_{ij}\theta_{ij}^{(m)}) - q_{ij}\{1 - \pi(q_{ij}\theta_{ij}^{(m)})\}(\theta - \theta_{ij}^{(m)}) + \tfrac{1}{8}(\theta - \theta_{ij}^{(m)})^2 \Big]$$

for the uniform bound. Note that

$$\{2\pi(q_{ij}\theta_{ij}^{(m)}) - 1\}/\{4q_{ij}\theta_{ij}^{(m)}\} = \{2\pi(\theta_{ij}^{(m)}) - 1\}/\{4\theta_{ij}^{(m)}\}$$

for $q_{ij} = \pm 1$. By completing the squares and using the definitions of $x_{ij}^{(m)}$ and $w_{ij}^{(m)}$, these majorizing functions can be rewritten as

$$\begin{aligned} &- \tilde{\ell}(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B} | \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)}) \\ &\quad = -\ell(\boldsymbol{\Theta}^{(m)}) - 2\sum_{i=1}^{n}\sum_{j=1}^{d}\{1 - \pi(q_{ij}\theta_{ij}^{(m)})\}^2 + \sum_{i=1}^{n}\sum_{j=1}^{d} w_{ij}^{(m)}(\theta_{ij} - x_{ij}^{(m)})^2. \end{aligned}$$

On the other hand, application of (2.7) yields the following majorizing function of $P_{\boldsymbol{\lambda}}(\mathbf{B})$:

$$\begin{aligned} \tilde{P}_{\boldsymbol{\lambda}}(\mathbf{B}|\mathbf{B}^{(m)}) &= \lambda_1 \sum_{j=1}^{d} \frac{b_{j1}^2 + b_{j1}^{(m)2}}{2|b_{j1}^{(m)}|} + \cdots + \lambda_k \sum_{j=1}^{d} \frac{b_{jk}^2 + b_{jk}^{(m)2}}{2|b_{jk}^{(m)}|} \\ &= \sum_{j=1}^{d} \mathbf{b}_j^{(m)T} \mathbf{D}_{\boldsymbol{\lambda},j}^{(m)} \mathbf{b}_j^{(m)} + \sum_{j=1}^{d} \mathbf{b}_j^T \mathbf{D}_{\boldsymbol{\lambda},j}^{(m)} \mathbf{b}_j. \end{aligned}$$

Since the majorization relation between functions is closed under the formation of sums, $-\tilde{\ell} + n\tilde{P}_{\boldsymbol{\lambda}}(\mathbf{B}|\mathbf{B}^{(m)})$ majorizes $S(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$ at $(\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$. Noticing that $-\tilde{\ell} + n\tilde{P}_{\boldsymbol{\lambda}}(\mathbf{B}|\mathbf{B}^{(m)})$ equals $g(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}|\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$ up to a constant independent of $(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$, we complete the proof of part (i). Part (ii) of the theorem follows from the general property of the MM algorithm (Hunter and Lange, 2004). ♢

The majorizing function given in (2.10) is quadratic in each of $\boldsymbol{\mu}$, $\mathbf{A}$, and $\mathbf{B}$ when the other two are fixed and thus alternating minimization of (2.10) with respect to $\boldsymbol{\mu}$, $\mathbf{A}$, and $\mathbf{B}$ has closed-form solutions. We now drop the superscript in $x_{ij}^{(m)}$ for notational convenience. For fixed $\mathbf{A}$ and $\mathbf{B}$, set $x_{ij}^{\star} = x_{ij} - \mathbf{a}_i^T \mathbf{b}_j$, the optimal $\hat{\mu}_j$ is given by

$$\hat{\mu}_j = \arg\min_{\mu_j} \sum_{i=1}^{n} w_{ij}\big(x_{ij}^{\star} - \mu_j\big)^2 = \frac{\sum_{i=1}^{n} w_{ij} x_{ij}^{\star}}{\sum_{i=1}^{n} w_{ij}}, \quad j = 1, \ldots, d. \tag{2.11}$$

To update $\mathbf{A}$ and $\mathbf{B}$ for fixed $\boldsymbol{\mu}$, set $x_{ij}^{*} = x_{ij} - \mu_j$ or in matrix form, $\mathbf{X}^{*} = (x_{ij}^{*}) = \mathbf{X} - \mathbf{1}_n \otimes \boldsymbol{\mu}^T$. Denote the ith row vector of $\mathbf{X}^{*}$ as $\mathbf{x}_i^{*}$ and let $\mathbf{W}_i = \mathrm{diag}(\mathbf{w}_i)$ where

$\mathbf{w}_i = (w_{i1}, \ldots, w_{id})^T$. For fixed $\boldsymbol{\mu}$ and $\mathbf{B}$, the ith row of $\mathbf{A}$ is updated by solving the following weighted least squares problem

$$\min_{\mathbf{a}_i} \sum_{j=1}^{d} w_{ij}\big(x_{ij}^* - \mathbf{a}_i^T\mathbf{b}_j\big)^2 \quad \text{or} \quad \min_{\mathbf{a}_i}\big(\mathbf{x}_i^* - \mathbf{B}\mathbf{a}_i\big)^T\mathbf{W}_i\big(\mathbf{x}_i^* - \mathbf{B}\mathbf{a}_i\big),$$

which has a closed form solution

$$\hat{\mathbf{a}}_i = \big(\mathbf{B}^T\mathbf{W}_i\mathbf{B}\big)^{-1}\mathbf{B}^T\mathbf{W}_i\mathbf{x}_i^*, \qquad i = 1, \cdots, n. \tag{2.12}$$

The columns of updated $\mathbf{A}$ can be made orthonormal by using the QR decomposition. Denote the jth column vector of $\mathbf{X}^*$ as $\tilde{\mathbf{x}}_j^*$ and let $\widetilde{\mathbf{W}}_j = \text{diag}(\tilde{\mathbf{w}}_j)$ with $\tilde{\mathbf{w}}_j = (w_{1j}, \ldots, w_{nj})^T$. For fixed $\boldsymbol{\mu}$ and $\mathbf{A}$, the jth row of $\mathbf{B}$ is updated by solving the following weighted ridge regression problem

$$\min_{\mathbf{b}_j} \sum_{i=1}^{n} w_{ij}\big(x_{ij}^* - \mathbf{a}_i^T\mathbf{b}_j\big)^2 + n\sum_{l=1}^{k}\lambda_l\frac{b_{jl}^2}{2|b_{jl}^{(m)}|}$$

or

$$\min_{\mathbf{b}_j} \big(\tilde{\mathbf{x}}_j^* - \mathbf{A}\mathbf{b}_j\big)^T\widetilde{\mathbf{W}}_j\big(\tilde{\mathbf{x}}_j^* - \mathbf{A}\mathbf{b}_j\big) + n\mathbf{b}_j^T\mathbf{D}_{\boldsymbol{\lambda},j}\mathbf{b}_j,$$

which has a closed form solution

$$\hat{\mathbf{b}}_j = \big(\mathbf{A}^T\widetilde{\mathbf{W}}_j\mathbf{A} + n\mathbf{D}_{\boldsymbol{\lambda},j}\big)^{-1}\mathbf{A}^T\widetilde{\mathbf{W}}_j\tilde{\mathbf{x}}_j^* \qquad j = 1, \cdots, d. \tag{2.13}$$

The MM algorithm will alternate between (2.11), (2.12), and (2.13) until convergence. The details are summarized in **Algorithm 1**.

When the uniform bound is used in the majorization of the negative log inverse logit function, computation in the MM algorithm can be simplified, because the weight matrices $\mathbf{W}_i$ and $\widetilde{\mathbf{W}}_j$ are equal to the identity matrix multiplied by a constant. The updating formula (2.11) of $\boldsymbol{\mu}$ becomes $\hat{\boldsymbol{\mu}} = \frac{1}{n}\mathbf{X}^{\star T}\mathbf{1}_n$, which is obtained by taking the column means of $\mathbf{X}^\star = (x_{ij}^\star)$. The updating formula (2.12) becomes $\hat{\mathbf{a}}_i = \big(\mathbf{B}^T\mathbf{B}\big)^{-1}\mathbf{B}^T\mathbf{x}_i^*$, $i = 1, \cdots, n$, which can be obtained by a single matrix calculation $\widehat{\mathbf{A}} = \mathbf{X}^*\mathbf{B}\big(\mathbf{B}^T\mathbf{B}\big)^{-1}$. The updating formula

Algorithm 1 *Sparse Logistic PCA Algorithm I*

1. Initialize $\boldsymbol{\mu}$, $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_n)^T$ and $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_d)^T$.
2. Compute x_{ij} using (2.8) and w_{ij} using (2.9).
3. Set $x^\star_{ij} = x_{ij} - \mathbf{a}_i^T \mathbf{b}_j$. Update $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)^T$ using
$$\mu_j = \frac{\sum_{i=1}^n w_{ij} x^\star_{ij}}{\sum_{i=1}^n w_{ij}}, \qquad j = 1, \cdots, d.$$
4. Set $\mathbf{X}^* = (x^*_{ij}) = \mathbf{X} - \mathbf{1}_n \otimes \boldsymbol{\mu}^T$.
5. Denote the ith row vector of $\mathbf{X}^*$ as $\mathbf{x}_i^*$. Set $\mathbf{W}_i = \text{diag}(\mathbf{w}_i)$ with $\mathbf{w}_i = (w_{i1}, \ldots, w_{id})^T$. Update $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_n)^T$ using
$$\mathbf{a}_i = \left(\mathbf{B}^T \mathbf{W}_i \mathbf{B}\right)^{-1} \mathbf{B}^T \mathbf{W}_i \mathbf{x}_i^*, \qquad i = 1, \cdots, n.$$
Compute the QR decomposition $\mathbf{A} = \mathbf{QR}$ and let $\mathbf{A} \leftarrow \mathbf{Q}$.
6. Denote the jth column vector of $\mathbf{X}^*$ as $\tilde{\mathbf{x}}_j^*$. Set $\widetilde{\mathbf{W}}_j = \text{diag}(\tilde{\mathbf{w}}_j)$ with $\tilde{\mathbf{w}}_j = (w_{1j}, \ldots, w_{nj})^T$. Compute $\mathbf{D}_{\boldsymbol{\lambda},j}$ as in (2.10). Update $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_d)^T$ using
$$\mathbf{b}_j = \left(\mathbf{A}^T \widetilde{\mathbf{W}}_j \mathbf{A} + n\mathbf{D}_{\boldsymbol{\lambda},j}\right)^{-1} \mathbf{A}^T \widetilde{\mathbf{W}}_j \tilde{\mathbf{x}}_j^*, \qquad j = 1, \cdots, d.$$
7. Repeat steps 2 and 6 until convergence.

(2.13) becomes $\hat{\mathbf{b}}_j = (\mathbf{I}_k + 8n\mathbf{D}_{\boldsymbol{\lambda},j})^{-1} \mathbf{A}^T \tilde{\mathbf{x}}_j^*$, $j = 1, \cdots, d$. Here, since the matrices to be inverted are diagonal matrices, $\hat{\mathbf{b}}_j$ can be obtained by component-wise shrinkage

$$\hat{b}_{jl} = \frac{|b_{jl}^{(m)}|}{|b_{jl}^{(m)}| + 4n\lambda_l} \tilde{\mathbf{a}}_l^T \tilde{\mathbf{x}}_j^*, \qquad l = 1, \cdots, k, \quad j = 1, \cdots, d,$$

where $\tilde{\mathbf{a}}_l$ is the lth column of $\mathbf{A}$. The simplified algorithm is summarized in **Algorithm 2**.

Our experience is that the MM algorithm using the uniform bound takes more iterations to converge, but because of the computational simplicity of each iteration, its actual computing time is less than the MM algorithm using the tight bound. We used the MM algorithm with the uniform bound (i.e., **Algorithm 2**) to produce all numerical results to

Algorithm 2 *Sparse Logistic PCA Algorithm II*

1. Initialize $\boldsymbol{\mu}$, $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_n)^T$ and $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_d)^T$.
2. Compute x_{ij} using (2.8).
3. Set $\mathbf{X}^\star = (X^\star_{ij})$ with $x^\star_{ij} = x_{ij} - \mathbf{a}_i^T\mathbf{b}_j$. Update $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)^T$ using $\boldsymbol{\mu} = \frac{1}{n}\mathbf{X}^{\star T}\mathbf{1}_n$.
4. Set $\mathbf{X}^* = (x^*_{ij}) = \mathbf{X} - \mathbf{1}_n \otimes \boldsymbol{\mu}^T$.
5. Update $\mathbf{A}$ by $\mathbf{A} = \mathbf{X}^*\mathbf{B}\big(\mathbf{B}^T\mathbf{B}\big)^{-1}$. Compute the QR decomposition $\mathbf{A} = \mathbf{QR}$ and let $\mathbf{A} \leftarrow \mathbf{Q}$.
6. Set $\mathbf{C} = (c_{jl}) = \mathbf{X}^{*T}\mathbf{A}$. Update $\mathbf{B} = (b_{jl})$ using
$$b_{jl} = \frac{|b^{(m)}_{jl}|}{|b^{(m)}_{jl}| + 4n\lambda_l} c_{jl} \qquad l = 1, \cdots, k, \quad j = 1, \cdots, d,$$
7. Repeat steps 2 and 6 until convergence.

be reported later in this chapter.

2.3 Geometry of MM Algorithm for Sparse Solutions

In this section, we examine how the quadratic approximated penalty function can give a sparse solution in MM algorithm, although it has a quadratic form. In order to obtain the sparse solution of principal component loadings, L_1 penalty function which is not differentiable at zero is introduced here, as in many regression problems. Nondifferentiabilty at zero is crucial for the sparse solution, which is addressed in many literature (Tibshirani, 1996; Fan and Li, 2001). Thus, it is instructive to mention how the ridge type penalty can produce the sparse solution by the iteration procedure although it is quadratic and differentiable at zero.

At the $m+1$th iteration step, when $\mathbf{A}$ is given by the previous estimate at the mth

step, the estimation procedure of the principal component loading $\mathbf{B}$ in (2.10) becomes the penalized weighted least square problem given as

$$\min_{\mathbf{b}} (\tilde{\mathbf{x}} - \mathbf{A}\mathbf{b})^T \widetilde{\mathbf{W}} (\tilde{\mathbf{x}} - \mathbf{A}\mathbf{b}) + n\mathbf{b}^T \mathbf{D}_{\boldsymbol{\lambda},j} \mathbf{b}. \tag{2.14}$$

Here we deliberately ignored the subscript j since each row of $\mathbf{B}$ is updated separately. To make our arguments simple, we assume all of λ_l's are the same here. Then (2.14) is equivalent to the weighted least square problem with the elliptical constraint, i.e.,

$$\min_{\mathbf{b}} (\tilde{\mathbf{x}} - \mathbf{A}\mathbf{b})^T \widetilde{\mathbf{W}} (\tilde{\mathbf{x}} - \mathbf{A}\mathbf{b}) \quad \text{subject to} \quad \sum_{l=1}^{k} b_l^2 / |b_l^{(m)}| \leq \tau \tag{2.15}$$

where τ is a constant depending on the regularization parameter λ. The constraint term appears as k-dimensional ellipsoid centered at the origin whose axes are proportional to the magnitude of the previous estimate of $\mathbf{b}_l$. The artificial example of $k = 2$ case is depicted in Figure 6. The elliptical contours show the quadratic objective function in (2.15) which is minimized. It is centered at the ordinary least square estimator which is obtained without constraints. The constraint regions appear as shaded ellipsoids. The ellipsoid with the dotted boundary stands for the constraint region of the optimization at the current iteration step. The solution, which is marked as "cross", occurs at the first point that the contours touch the ellipse. In the next iteration step the constraint region is constructed based on this new solution. Since b_2 is estimated larger than b_1, the constraint region is more shrunken along the b_1 axis, which is shown as the ellipse with the dashed boundary. At the next iteration, the solution occurs at "plus" mark. If b_1 is estimated small enough, the constraint region in the next step will collapse toward the origin along the b_1 axis, which is illustrated in the right panel of Figure 6. In that case b_1 has little chance to have large values. This mechanism generally explains how to generate a sparse solution even though the majorizing penalty function is differentiable at zero. Note that Figure 6 describes the regression situation with fixed covariates. In our PCA problem, the principal component

score matrix $\mathbf{A}$ constantly changes depending on the previous estimates so the elliptical contours are not the same at every iteration. However the main message from this figure still holds in such case.

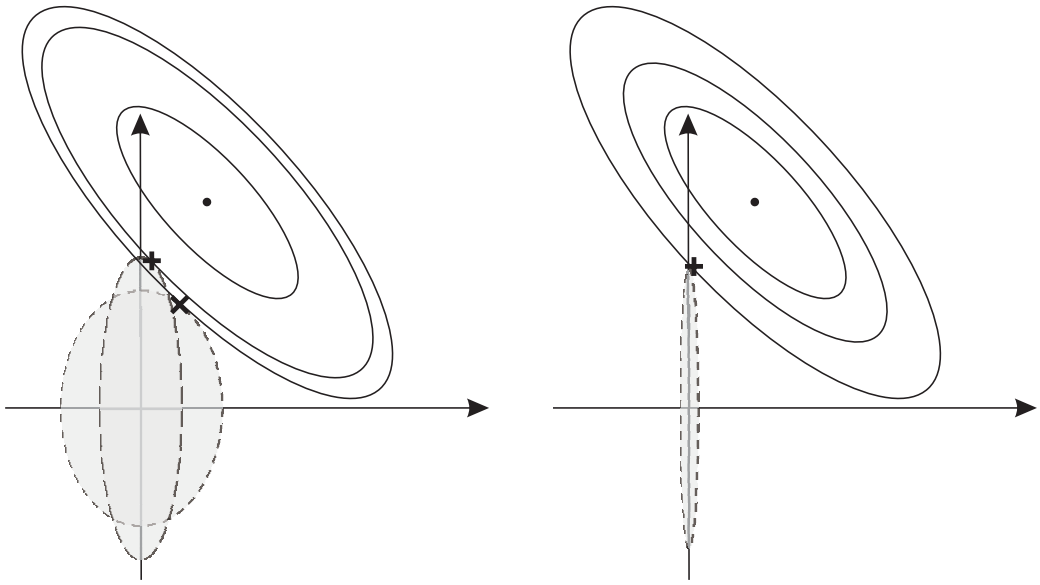


Figure 6: Estimation picture for MM algorithm. Left panel shows how the constraint region changes adaptively based on the previous solution. Right panel illustrates the case that the sparse solution is attained.

2.4 Implementation Issues

In this section we discuss the methods for selecting the tuning parameters in the sparse logistic PCA algorithm. Sections 2.4.1 and 2.4.2 treat the usual $n \gg d$ case. Section 2.4.3 handles the case when $d \gg n$ or d is comparable to n.

2.4.1 Choosing the penalty parameters

In the situation of $n \gg d$, leave-row-out cross-validation (CV) can be used to choose the regularization parameter $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_k)^T$. We propose to use the 5-fold version of the cross-validation. To this end, we randomly divide the rows of the data matrix to form 5

submatrices with approximately equal number of rows. Denote these submatrices as $\mathbf{Y}_{(i)}$, $i = 1, \cdots, 5$. Let $\mathbf{Y}_{(-i)}$ denote the submatrix of $\mathbf{Y}$ after removing $\mathbf{Y}_{(i)}$. For each i, we use $\mathbf{Y}_{(-i)}$ as a training set and use $\mathbf{Y}_{(i)}$ as a test set. The training set is used for extraction of the principal component loadings, the test set is projected to the loading vectors, and a goodness-of-fit measured using the negative log likelihood on the test set is calculated. The sum of the five goodness-of-fit measures is used as the crossvalidation score. We select the optimal $\boldsymbol{\lambda}$ which minimizes the crossvalidation score.

Alternatively, we can develop a GCV-type criterion based on the regression like calculation of the loading matrix. By (2.13) of Section 2.2.2, we see that the jth row of $\mathbf{B}$ can be obtained by a weighted ridge regression with the responses $\tilde{\mathbf{x}}_j^*$ and the predicted values of the responses are given by

$$\mathbf{A}\big(\mathbf{A}^T\mathbf{W}_j\mathbf{A} + n\mathbf{D}_{\boldsymbol{\lambda},j}\big)^{-1}\mathbf{A}^T\mathbf{W}_j\tilde{\mathbf{x}}_j^* \;=\; \mathbf{R}_{\boldsymbol{\lambda},j}\tilde{\mathbf{x}}_j^*,$$

where $\mathbf{R}_{\boldsymbol{\lambda},j} = \mathbf{A}\big(\mathbf{A}^T\mathbf{W}_j\mathbf{A} + n\mathbf{D}_{\boldsymbol{\lambda},j}\big)^{-1}\mathbf{A}^T\mathbf{W}_j$ is the hat matrix. Following the usual development of GCV (Hastie and Tibshirani, 1990), we define the GCV score for sparse logistic PCA as

$$GCV(\boldsymbol{\lambda}) \;=\; \frac{1}{d}\sum_{j=1}^{d}\frac{\|\tilde{\mathbf{x}}_j^* - \mathbf{R}_{\boldsymbol{\lambda},j}\tilde{\mathbf{x}}_j^*\|^2}{n\{1 - \mathrm{Tr}(\mathbf{R}_{\boldsymbol{\lambda},j})/n\}^2}.$$

Our simulation study, not presented here, shows that both CV and GCV work well when $n \gg d$. But when d is larger than or even comparable to n we observed that CV and GCV fail to find good regularization parameters. A new method is proposed in Section 2.4.3 below to deal with this difficult case.

2.4.2 Determining the dimensionality of the subspace

In the standard PCA, the percentage of total variance explained by the principal components can be defined and is frequently used for choosing the appropriate number of principal components with the aid of a "screeplot". Zou et al. (2006) and Shen and Huang (2008)

extended this approach to sparse PCA by modifying the definition of variance explained by the PCs. We propose to use a similar strategy for sparse logistic PCA but using the Bernoulli likelihood instead of the variance to measure the goodness-of-fit. Specifically, we draw the plot of the negative log likelihood as a function of k. The plot usually starts with a quick drop and after a "knee" or "ankle" point, the drop is much slower. The "k" corresponding to this "knee" point is chosen as a suitable dimension to project the data for logistic PCA. Another approach for selecting "k" is to use the model selection criteria such as the AIC or BIC. Our simulation study (not shown) reveals that both approaches work well when $n \gg d$. However, when $d \gg n$, our experience shows that the screeplot method and the AIC criterion tend to select k conservatively (large k) and BIC tends to choose the anti-conservative k (small k). In the next subsection, we develop a method to determine k for the case that $d \gg n$ or d is comparable to n.

2.4.3 High-dimensional low-sample-size settings

When the number of variables d is large, we suggest to use a single regularization parameter λ for all PC loadings to reduce the computation time, unless there is a need to consider the different regularization. We use the following strategy to decide the two tuning parameters. We first fix k at a reasonable large value and select a good λ, then using this λ we refine the choice of k.

Since the AIC criterion usually selects a k that is bigger than what is needed, we first fix k at the AIC selected value when focusing on the selection of λ. Note that a larger value of λ will lead to a smaller number of nonzeros in the loading matrix $\mathbf{B}$ and reduced model complexity, the reduced model complexity is usually associated with less good fit of the model. To compromise the goodness-of-fit and model complexity, we use the corrected BIC criterion defined by

$$\text{CBIC}(\lambda) = -2\ell(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) + \log n \times m(\lambda)$$

where $m(\lambda)$ is the number of nonzero parameters. Let $\mathcal{B}(\lambda)$ denote the index set of the nonzero loadings in $\mathbf{B}$ with the regularization parameter λ and $|\cdot|$ denote the cardinality function of the set argument. Then $|\mathcal{B}(\lambda)|$ is the number of total nonzero loadings in $\mathbf{B}$ obtained by the regularized logistic PCA at λ, and thus $m(\lambda) = d+nk+|\mathcal{B}(\lambda)|$. The corrected BIC is studied in Zou et al. (2007), where it is shown that the number of nonzero coefficients is an unbiased estimate for the degrees of freedom for the LASSO regression. We select the optimal λ which minimizes the corrected BIC criterion. Fixing the selected λ we choose the optimal "k" again by minimizing the corrected BIC. The screeplot as discussed in the previous section can also be used to decide on the value of "k". The effectiveness of the above selection procedure in the high dimensional scenario will be demonstrated in the simulation study and the real data applications in the following sections.

2.5 Handling Missing Data

Missing data are commonly encountered in real applications. In this section, we extend our sparse logistic PCA method to cases when missing data are present.

Let $\mathcal{N} = \{(i,j)|y_{ij} \text{ is not observed}\}$ denote the index set for missing values. The sparse logistic PCA minimizes the following criterion function

$$T(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) = -\ell_{obs}(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) + nP_{\boldsymbol{\lambda}}(\mathbf{B}), \tag{2.16}$$

where

$$\ell_{obs}(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) = \sum_{(i,j)\notin\mathcal{N}} \sum \log \pi\{q_{ij}(\mu_j + \mathbf{a}_i^T \mathbf{b}_j)\}$$

can be interpreted as the observed data log likelihood. Similar to the non-missing data case, direct minimization of (2.16) is not straightforward because the log likelihood term is not quadratic and the penalty term is non-differentiable. Direct minimization of (2.16) is also complicated by the fact that the summation in the definition of the observed data

log likelihood is not over a rectangular region. Again, we develop an MM algorithm to iteratively solve the optimization problem.

Define the working variables

$$z_{ij}^{(m)} = \begin{cases} x_{ij}^{(m)}, & (i,j) \notin \mathcal{N} \\ \theta_{ij}^{(m)} = \mu_j^{(m)} + \mathbf{a}_i^{(m)T}\mathbf{b}_j^{(m)}, & (i,j) \in \mathcal{N}. \end{cases}$$

where $x_{ij}^{(m)}$ is defined in (2.8). Let

$$\begin{aligned} &h(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B} | \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)}) \\ &\quad = \sum_{i=1}^{n}\sum_{j=1}^{d}\Big[w_{ij}^{(m)}\big\{z_{ij}^{(m)} - (\mu_j + \mathbf{a}_i^T\mathbf{b}_j)\big\}^2 + \mathbf{b}_j^T\mathbf{D}_{\boldsymbol{\lambda},j}^{(m)}\mathbf{b}_j\Big], \end{aligned} \tag{2.17}$$

where $\mathbf{D}_{\boldsymbol{\lambda},j}^{(m)}$ are diagonal matrices with diagonal elements $\lambda_l/2|b_{jl}^{(m)}|$ for $l = 1, \cdots, k$. The following result extends Theorem II.1 to the missing data case.

THEOREM **II.2**. (i) *Up to a constant that depends on* $\boldsymbol{\mu}^{(m)}$, $\mathbf{A}^{(m)}$, *and* $\mathbf{B}^{(m)}$ *but not on* $\boldsymbol{\mu}$, $\mathbf{A}$, *and* $\mathbf{B}$, *the function* $h(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B} | \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$ *defined in* (2.17) *majorizes* $T(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$ *at* $(\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$.

(ii) *Let* $(\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$, $m = 1, 2, \ldots$, *be a sequence obtained by iteratively minimizing the majorizing function. Then* $T(\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$ *increases with* m *and it converges to a local minimum of* $T(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$ *as* m *goes to infinity.*

Proof. Note that the objective function to be minimized is the summation of two terms – the log likelihood term and the penalty term. Because the majorization property is closed under function summation, we deal with the two terms separately. We can find a majorization function of the penalty term as in Theorem II.1. To find a majorization function of the log likelihood term, we apply the argument in the standard EM algorithm for handling missing data (Dempster et al., 1977). The complete data log likelihood is

$$\ell_{com}(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) = \sum_{(i,j)\notin\mathcal{N}}\sum \log \pi(q_{ij}\theta_{ij}) + \sum_{(i,j)\in\mathcal{N}}\sum \log \pi(q_{ij}\theta_{ij}).$$

Its conditional expectation given the observed data and the current guess of the parameter values is

$$\begin{aligned} & Q(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B} | \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)}) \\ = & \sum_{(i,j) \notin \mathcal{N}} \sum \log \pi(q_{ij}\theta_{ij}) \\ & + \sum_{(i,j) \in \mathcal{N}} \sum E[\log \pi(q_{ij}\theta_{ij}) | \mathbf{Y}_o, \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)}], \end{aligned} \tag{2.18}$$

where $\mathbf{Y}_o$ denote the observed data. By the standard EM theory,

$$\begin{aligned} & -\tilde{\ell}_{obs}(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) \\ & = -Q(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B} | \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)}) - \ell_{obs}(\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)}) \\ & \quad + Q(\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)} | \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)}) \end{aligned} \tag{2.19}$$

majorizes $-\ell_{obs}(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$ at $(\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$, that is, $-\tilde{\ell}_{obs}(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) \geq -\ell_{obs}(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$, and the equality holds when $(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}) = (\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$.

Now we find a quadratic majorizing function of $-\tilde{\ell}_{obs}(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$, which in turn majorizes $-\ell_{obs}(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$ because of the transitivity of the majorization relation. We need only to find a quadratic majorization function of $-Q(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B} | \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$ since it is the only term in the definition (2.19) of $-\tilde{\ell}_{obs}(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B})$ that depends on the unknown parameters. According to (2.18), $-Q(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B} | \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$ can be decomposed into two terms, one corresponding to observed data, the other corresponding to the missing data. The former term can been treated as in the proof of Theorem II.1. When $(i,j) \notin \mathcal{N}$, $-\log \pi(q_{ij}\theta_{ij})$ is majorized by $w_{ij}^{(m)}(\theta_{ij} - x_{ij}^{(m)})^2$, up to a constant. To treat the latter term, note that, when $(i,j) \in \mathcal{N}$,

$$\begin{aligned} & E\big[\log \pi(q_{ij}\theta_{ij}) | \mathbf{Y}_o, \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)}\big] \\ & = \pi(\theta_{ij}^{(m)}) \log \pi(\theta_{ij}) + \{1 - \pi(\theta_{ij}^{(m)})\} \log\{1 - \pi(\theta_{ij})\} \\ & = \sum_{q_{ij} = \pm 1} \pi(q_{ij}\theta_{ij}^{(m)}) \log \pi(q_{ij}\theta_{ij}), \end{aligned}$$

using the fact that the missing data is independent of the observed data, and that $1-\pi(\theta) = \pi(-\theta)$. Then, by applying the inequalities (2.5) and (2.6) and using the definition of $w_{ij}^{(m)}$, we obtain that

$$\begin{aligned}
&- E\big[\log \pi(q_{ij}\theta_{ij})|\mathbf{Y}_o, \boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)}\big] \\
&\leq \sum_{q_{ij}=\pm 1} \pi(q_{ij}\theta_{ij}^{(m)})\big[-\log \pi(\theta_{ij}^{(m)}) \\
&\qquad - \{1-\pi(q_{ij}\theta_{ij}^{(m)})\}\{q_{ij}(\theta_{ij}-\theta_{ij}^{(m)})\} + w_{ij}^{(m)}\{(\theta_{ij}-\theta_{ij}^{(m)})\}^2\big] \\
&\leq C_m + w_{ij}^{(m)}\{(\theta_{ij}-\theta_{ij}^{(m)})\}^2,
\end{aligned}$$

where C_m is a constant independent of $\boldsymbol{\mu}$, $\mathbf{A}$, and $\mathbf{B}$. Combining the above results, we see that $-Q(\boldsymbol{\mu}, \mathbf{A}, \mathbf{B}|\boldsymbol{\mu}^{(m)}, \mathbf{A}^{(m)}, \mathbf{B}^{(m)})$ is up to a constant majorized by $\sum_{ij} w_{ij}^{(m)}\{(\theta_{ij} - z_{ij}^{(m)})\}^2$, where $z_{ij}^{(m)}$ equals $x_{ij}^{(m)}$ if $(i,j) \notin \mathcal{N}$, and $\theta_{ij}^{(m)}$ if $(i,j) \in \mathcal{N}$. The proof of Part (i) is thus complete. Part (ii) of the theorem follows from the general result of the MM algorithm. $\diamondsuit$

Note that the majorizing functions given in (2.17) have the same form as those given in (2.10) except that $x_{ij}^{(m)}$ in (2.10) is changed to $z_{ij}^{(m)}$ in (2.17). Thus the computation algorithm developed in Section 2.2.2 is readily applicable in the missing data case with a simple replacement of $x_{ij}^{(m)}$ by $z_{ij}^{(m)}$. The working variable $z_{ij}^{(m)}$ in (2.17) is easily understood: It is the same as the non-missing data case if $y_{i,j}$ is observable; otherwise, it is an imputed θ_{ij} value based on the reduced rank model (2.2) and the current guess of $\boldsymbol{\mu}$, $\mathbf{A}$, and $\mathbf{B}$.

2.6 Simulation Study

In this section we demonstrate our sparse logistic PCA method using a simulation study. The method worked well in various settings that we tested, but here we only report results in a challenging case that the number of variables d is bigger than the sample size n.

2.6.1 The signal-to-noise ratio

We first introduce a notion of signal-to-noise ratio for logistic PCA. In our logistic PCA model, the entries of the $n \times d$ data matrix are independent Bernoulli random variables with success probability $\pi_{ij} = \{1 + \exp(-\theta_{ij})\}^{-1}$ for the (i, j) cell. The matrix of canonical parameters $\mathbf{\Theta} = (\theta_{ij})$ has a reduced rank representation $\mathbf{\Theta} = \boldsymbol{\mu} + \mathbf{A}\mathbf{B}^T$, where $\mathbf{A}$ is a $n \times k$ matrix of PC scores and $\mathbf{B}$ is a sparse $d \times k$ PC loading matrix. In our simulation study, each column of $\mathbf{A}$ is generated from a zero-mean Gaussian distribution. The variances of these Gaussian distributions measure the signal levels of the PCs. We set up these PC variances relative to a suitably defined baseline noise level.

We define a baseline noise level as follows. First we generate $n \times d$ independent binary variables from Bernoulli distribution with the success probability $1/2$. These binary variables are understood to come from the pure noise since they are generated without having any structure on the success probabilities. Using these binary variables, we would like to determine a noise level in the canonical parameter space. To this end, we conduct a k-component logistic PCA without regularization and then compute the average of variances for the obtained k PC scores, which is denoted as σ_b^2. This average variance can serve as a measure of the baseline noise level. To get a more stable measure of the baseline noise level, we generate a large number of (for example, 100) "pure noise" binary data matrices and take the median of σ_b^2 computed from these matrices as our baseline noise level. The baseline noise level depends on n, d, and k.

With the notion of baseline noise level, we define the signal-to-noise ratio (SNR) for a PC as

$$\text{SNR} \;=\; \frac{\text{variance of PC scores}}{\text{baseline noise level}}.$$

In our simulation study, we first compute the baseline noise level for a given combination of n, d, and k, then use the above formula to specify the variances of PC scores based on

the fixed values of SNR. Table 1 reports SNRs under some scenarios with the sample size $n = 100$, which will be used for the following simulation study.

Table 1: Simulations on the baseline noise level with the sample size $n = 100$ in the standard deviation scale. The averages of k PC score standard deviations are computed over 100 simulated datasets. Table shows median and MAD (median of absolute deviation) of 100 averages. The squared value of them is used as the baseline noise level.

k	$d = 200$	$d = 500$	$d = 1000$
1	36.63 (1.54)	55.89 (1.06)	77.87 (1.08)
2	37.37 (3.89)	56.73 (4.23)	78.73 (4.38)
10	47.30 (4.92)	67.30 (5.46)	90.17 (4.79)
20	54.65 (3.91)	75.20 (4.17)	99.16 (4.53)
100	14.08 (2.66)	22.31 (1.74)	90.17 (4.79)

2.6.2 Simulation setup

We set the intrinsic dimension to be $k = 2$ and the number of rows of the data matrix to be $n = 100$. We vary the number of the variables d and the signal-to-noise ratio SNR. We construct two sparse PC loading vectors as follows: Let b_{j1} and b_{j2} denote correspondingly the components of the first and the second PC loading vectors. We let $b_{j1} = 1$ for $j = 1, \cdots, 20$, $b_{j2} = 1$ for $j = 21, \cdots, 40$, and the rest of b_{jl} are all taken to be 0. We consider three choices of d: $d = 200$, $d = 500$, and $d = 1000$. We consider two settings of SNR: $(3, 2)$ and $(5, 3)$, and the SNRs are used to determine the variances of the PC scores. For example, when the SNR is $(3, 2)$, the variance of the first PC is 3 times the baseline noise level and the variance of the second PC is 2 times the baseline noise level. The mean vector $\boldsymbol{\mu}$ is set to be a vector of zeros.

2.6.3 Simulation results

Logistic PCA with and without sparsity-inducing regularization is conducted on 100 simulated datasets for each setting. To measure the closeness of the estimated PC loading matrix

Table 2: The results of logistic PCA with and without sparsity-inducing regularization, based on 100 simulated data sets for each setting. The reported angle is the median angle. The description of results is in the text.

		$k = 2$			$k = 30$		
d	SNR Regularization	angle (°)	correct (%)	incorrect (%)	angle (°)	correct (%)	incorrect (%)
200	SNR=$(3, 2)$						
	nonregularized	12.410	100	100	35.550	100	100
	regularized	11.910	100	95.62	11.270	100	47.19
	SNR=$(5, 3)$						
	nonregularized	11.770	100	100	36.230	100	100
	regularized	11.060	100	95.62	11.060	100	44.38
500	SNR=$(3, 2)$						
	nonregularized	10.770	100	100	31.540	100	100
	regularized	6.322	100	30.43	9.730	100	19.13
	SNR=$(5, 3)$						
	nonregularized	10.240	100	100	31.490	100	100
	regularized	6.202	100	28.59	9.642	100	18.91
1000	SNR=$(3, 2)$						
	nonregularized	11.630	100	100	35.810	100	100
	regularized	5.218	88.12	8.85	12.950	100	15.99
	SNR=$(5, 3)$						
	nonregularized	11.020	100	100	35.770	100	100
	regularized	4.696	100	9.79	12.470	100	15.94

$\widehat{\mathbf{B}}$ and the true loading matrix $\mathbf{B}$, we use the principal angle between spaces spanned by $\widehat{\mathbf{B}}$ and $\mathbf{B}$. The principal angle measures the maximum angle between any two vectors on the spaces generated by the columns of $\widehat{\mathbf{B}}$ and $\mathbf{B}$. More precisely, it is defined by $\cos^{-1}(\rho) \times 180/\pi$, where ρ is the minimum eigenvalue of the matrix $\mathbf{Q}_{\widehat{\mathbf{B}}}^T \mathbf{Q}_{\mathbf{B}}$, where $\mathbf{Q}_{\widehat{\mathbf{B}}}$ and $\mathbf{Q}_{\mathbf{B}}$ are orthogonal basis matrices obtained by the QR decomposition of matrices $\widehat{\mathbf{B}}$ and $\mathbf{B}$, respectively (Golub and van Loan, 1996). The median principal angles for logistic PCA with and without regularization are presented in Table 2. We used $k = 2$ and $k = 30$ when running the logistic PCA algorithms. Since smaller principal angles indicate better estimates of the PC loading matrix, the sparsity-inducing regularization has a clear benefit — it can substantially reduce the principal angles. The benefit is even more profound when the number of PCs used in the program ($k = 30$) is different from the true number that was

used to generate the data ($k = 2$).

Table 2 also presents the percentage of the correctly and incorrectly identified nonzero loadings. In most scenarios, using the sparse logistic PCA algorithm, there is no serious risk that the true nonzeros are not selected since the percentage of the correctly selected nonzeros are 100% except for the case when $d = 1000$ and SNR=$(3, 2)$ where it still reports relatively large percentage. The percentage of the incorrectly selected nonzeros is below 30% when the number of variables are 500 and 1000. This shows that regularization can remove most zero loading variables in such cases.

Table 3: Frequencies of the selected k using the corrected BIC.

		selected k						
d	SNR	1	2	3	4	5	6	7
200	$(3, 2)$	0	95	5	0	0	0	0
	$(5, 3)$	0	96	4	0	0	0	0
500	$(3, 2)$	1	58	37	4	0	0	0
	$(5, 3)$	0	60	36	3	1	0	0
1000	$(3, 2)$	3	34	36	15	10	1	1
	$(5, 3)$	2	31	47	15	4	1	0

We then chose the number of PCs k of the sparse logistic PCA by using the corrected BIC criterion which penalizes the model fit with the number of nonzero parameters. Frequencies of the selected k from 100 simulation datasets in each settings of Table 2 are shown in Table 3. When $d = 200$, the corrected BIC finds well the true number 2 but, as d gets larger, $k = 3$ is more frequently selected. The performance for the large d cases is considered as quite good, given that the sample size is only 100.

Figure 7 shows two PC loading vectors from one simulated data set for $d = 200$ and SNR=$(5, 3)$. While the sparse logistic PCA can recover the original loading vectors well, the nonregularized logistic PCA gives more noisy results which are also subject to a rotation to get close to the original vectors.

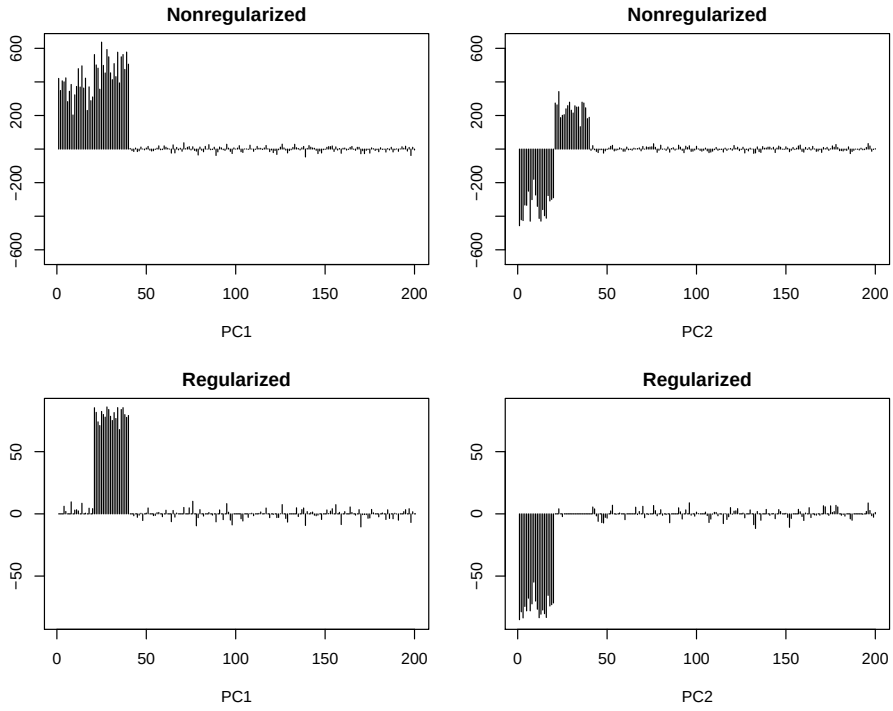


Figure 7: A simulated data set with $n = 100$, $d = 200$, and $k = 2$. Top panels shows the first and second PC loadings from the nonregularized PCA. The bottom panels are the same case of the regularized PCA.

2.7 Real Data Applications

In this section we illustrate the proposed sparse logistic PCA method to three real datasets where the dimension of data is comparable or larger than the sample size. The nonregularized logistic PCA is used for comparison.

2.7.1 Advertisement data

The advertisement data was collected to predict whether or not images obtained on Internet pages are advertisements based on a large number of their surrounding features. The feature encodes phrases occurring in the URL, the image's URL and alt text, the anchor text, and words occurring near the anchor text. The dataset and its description are available from the UCI machine learning repository (Asuncion and Newman, 2007). The dataset contains

1,558 variables with 3 continuous and 1,555 binary variables, and 3,279 observations with 459 advertisements and 2,820 non-advertisements. We focused on the binary variables and used the 3279×1555 binary data matrix. One binary variable has missing values. Although the objective of this data collection is the prediction of the advertisement webpages, we applied the sparse logistic PCA to this dataset in order to see whether PCA is able to capture the variability between two groups and also whether the sparsity-inducing regularization helps to improve the group separability. Top panels of Figure 8 present the scatterplots of the first two PC scores obtained from nonregularized and regularized logistic PCA. Clearly, the sparsity inducing regularization improved the group separability. This improvement is better seen in the boxplots of the first PC scores (bottom panels of Figure 8).

With the obtained PC scores, discrimination analysis was conducted using the linear discrimination analysis (LDA) and support vector machine (SVM) with linear, polynomial and radial kernels. To do this, we randomly select a third of data as the test set. We train the decision boundary using the remaining data (training set) and apply it to the test set. This was conducted 50 times. The regularized logistic PCA outperforms the logistic PCA without regularization especially when we use the small number of PC scores (Figure 9). This demonstrates the regularization is greatly helpful when we study the high dimensional data in the low dimensional space. It should be mentioned that advertisement dataset has been frequently used for assessment of many supervised learning algorithms, for instance C4.5 rules, yielding the high quality of prediction. However, the sparse binary PCA is the unsupervised learning without using any group information.

2.7.2 Single nucleotide polymorphism data

Association studies based on high-throughput single nucleotide polymorphism (SNP) data (Brooks, 1999; Kwok et al., 1996) have become a popular way to detect genomic regions associated with human complex disease. A SNP is a single base pair position in

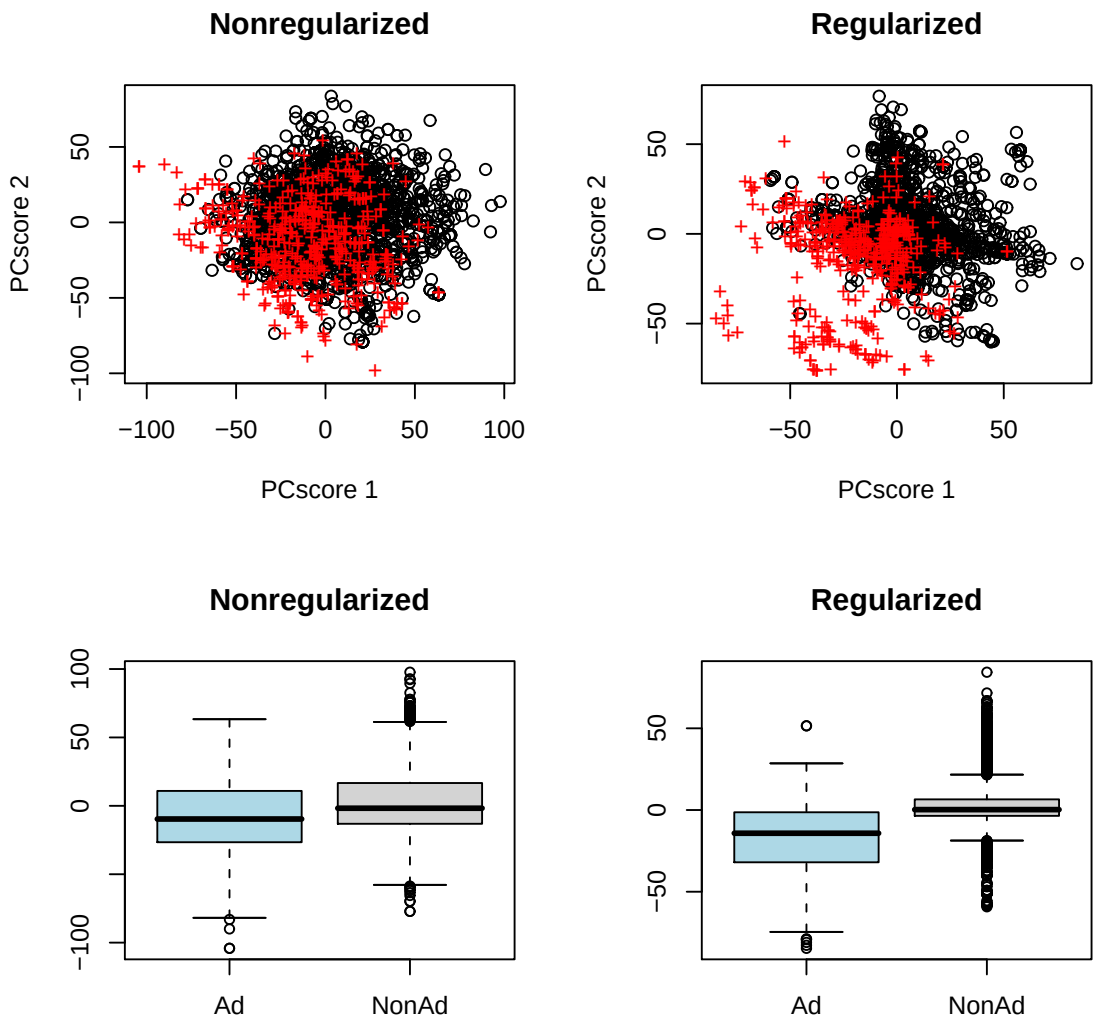


Figure 8: Advertisement data. Top panels: The scatterplots of the first two PC scores from the nonregularized (left) and regularized (right) logistic PCA. The red plus represents the advertisement case and the black circle shows the nonadvertisement case. Bottom panels: Boxplots of the first PC scores. The advertisement cases and nonadvertisement cases are labeled as “Ad” and “NonAd” respectively.

genomic DNA at which the sequence (alleles) variation occurs between members of a species, wherein the least frequent allele has an abundance of 1% or greater. A crucial issue in association studies is population stratification detection (Hao et al., 2004) which

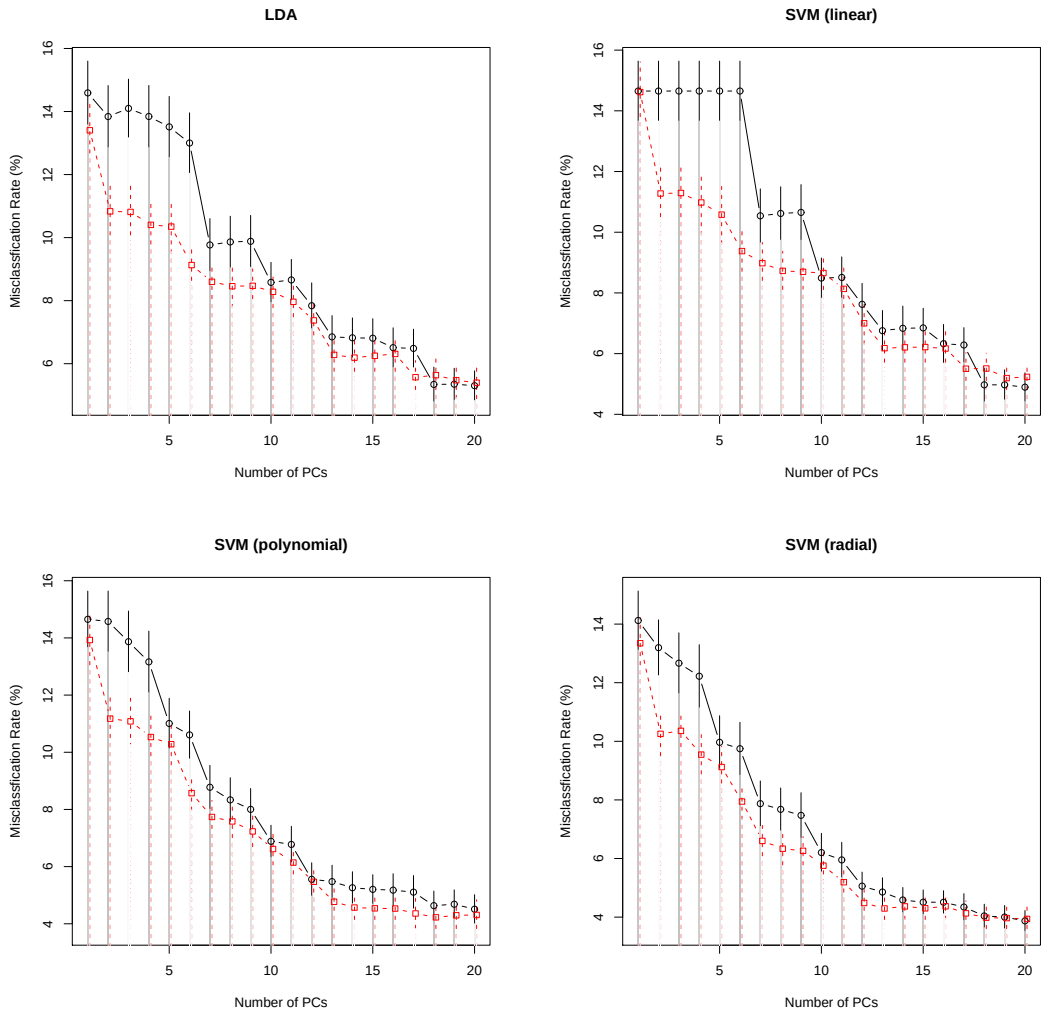


Figure 9: Discrimination analysis using LDA and SVM. Black circle and red rectangle show the misclassification rates using the nonregularizes and regularized PC scores respectively. Vertical bar stands margin of one standard deviation of 50 misclassification rates.

is to determine whether a population is homogeneous or has hidden structures within it. With the presence of population stratification, the naive case-control approach not accounting for this factor would yield biased results (Ewens and Spielman, 1995) and, therefore, draw inaccurate scientific conclusions. Also the additional analysis challenge arises from high dimensionality of the SNP data. Liang and Kelemen (2008) discusses extensively the statistical development and difficulties for SNP data analysis.

The proposed sparse logistic PCA method can be used for population stratification detection. For the purpose of demonstration, we use the SNP data set available in the International HapMap project (The International HapMap Consortium, 2005). It consists of 3 different ethnic populations of 90 Caucasians, 90 Africans and 90 Asians. Our task is to detect this three-subpopulation structure using the SNP data on the 270 subjects. At many SNP locations, heterozygosity distribution and allele frequency are known to be different among populations and could confound the effect of the risk of disease. To account for this factor, Serre et al. (2008) selected 1,536 SNPs with the similar heterozygosity distribution and allele frequency. The locations of these SNPs cover all the chromosomes except for the sex-determining chromosome. Among these 1,536 SNPs, 1,392 are shared by three ethnic groups, which are used in our analysis. Their distribution over chromosomes is presented in Table 4. We coded 0 for the most prevalent homogeneous base pair (wild-type) and 1 for others (mutant), resulting in a 270×1392 binary matrix. This data matrix has 2.37% missing entries.

Table 4: 1,392 SNP distribution over 22 chromosomes.

chromosome	1	2	3	4	5	6	7	8	9	10	11
number of SNPs	152	49	63	46	92	129	100	63	106	20	35
chromosome	12	13	14	15	16	17	18	19	20	21	22
number of SNPs	34	39	13	67	31	102	42	45	23	54	88

Figure 10 provides the scatterplots of first 2 PC scores with and without regularization. The clear splitting pattern among the three ethnic groups is shown in the regularized PCA case but not in the nonregularized PCA case. In addition, the proposed sparse method allows identifying directly the SNPs that contribute to this subpopulation pattern. The selected model yields 816 and 685 nonzero variable loadings (representing the SNPs) on the first 2 PC directions, among which 508 are commonly shared. Therefore, 993 SNPs

in the first 2 PC directions are claimed to be associated with the ethnic group effect. It suggests that the population stratification factor should be taken into consideration at these 993 SNP locations in the following study of the association between SNPs and the disease phenotype to avoid biased conclusion.

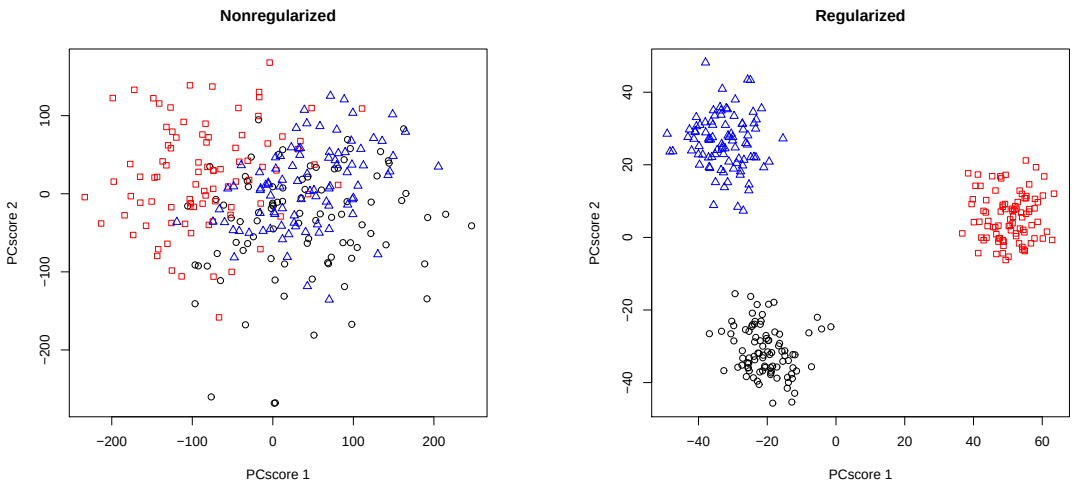


Figure 10: The scatterplots of the first two PC scores from the nonregularized (left) and regularized (right) logistic PCA. Black circles, red rectangles and blue triangles represent Caucasian, African and Asian population respectively.

2.7.3 Handwritten digits data

The handwritten digits data come from the ZIP code on envelopes from U.S. postal mail (Hastie et al., 2001). Each image is a segment from a five digit ZIP code, isolating a single digit. The images are 16×16 eight-bit grayscale maps. After deslanting and size-normalizing, 16×16 matrices of pixel intensities are obtained with scales ranging from -1 to 1. To illustrate the logistic PCA methods, the pixel intensity values less than 0 were coded as 1's and otherwise as 0's. In the original dateset, there are 500$\sim$1,200 images for each of the 10 digits. For each digit, we randomly selected 100 images to get a dataset whose sample size is smaller than the dimension $d = 16 \times 16 = 256$. Both regularized and

nonregularized logistic PCA were applied to these smaller datasets. We only present here results for the digit “5”.

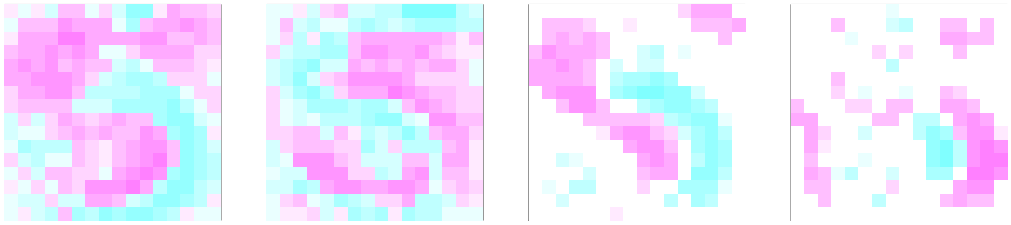


Figure 11: The first two panels from the left are the first 2 PC loadings from the nonregularized logistic PCA. The right two panels are the first 2 PC loadings from the regularized logistic PCA. The blue and red colors represent the positive and negative loading. The density of colors is proportional to their magnitude of loadings. Zero loadings are colored by white.

Figure 11 presents PC loadings from the nonregularized and regularized logistic PCA. The sparse PCA generates many spots with zero loadings and thus enhances the interpretability of the extracted PCs. For example, the first PC loading reflects the contrast between the strong “head” and “tail”, while the second PC loading explains the variability coming from the “width” of digits. The similar interpretation may be given for PC loadings obtained from the nonregularized logistic PCA, but the message is much less apparent because of many nonzero loadings. The enhanced interpretation by sparsity can be more easily appreciated by examining the images having the highest and lowest PC scores as shown in Figure 12. In particular, the five images with the highest first PC loading by the sparse PCA all have big round tail part and weak head while the five with the lowest first PC loadings show the opposite pattern (third row of Figure 12); the images with high and low second PC loadings show strong contrast in the size of the width (fourth row of Figure 12). As comparison, no clear patterns appeared using nonregularized logistic PCA. This example illustrates that regularization can help find interesting features or structures in binary data sets.

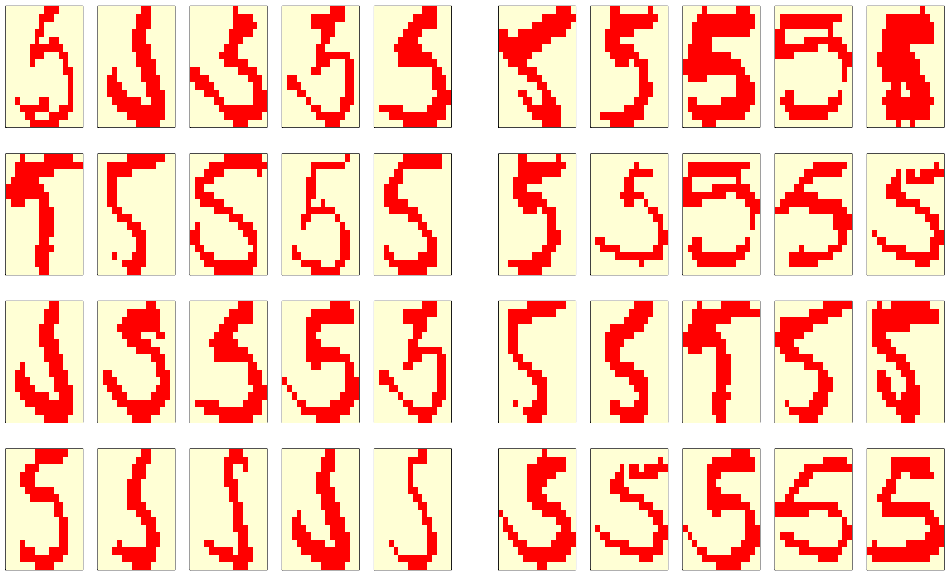


Figure 12: The sample images with the five highest (left) and lowest (right) PC scores. The first and second rows correspond to the first and second PCs of the nonregularized logistic PCA. The third and fourth rows correspond to the first and second PCs of the regularized logistic PCA.

CHAPTER III

LATENT VARIABLE MODEL FOR BINARY PRINCIPAL COMPONENTS ANALYSIS

In this chapter, we develop principal components analysis for binary variable data with latent variable models. The sparse solutions of principal component loadings are sought with the regularized maximum marginal likelihood estimation. The benefit from the regularization method in binary principal components analysis is that the derived principal component loadings have an easy interpretation and lead to better feature extraction. Since the EM formulation of latent variable model is intractable, we develop its variational approximation. Possible missing cases are considered and we provide their treatment. We also incorporate the situation where binomial and normal variables appear simultaneously with binary variables in the data and provide the unified algorithm in such case. The performance of regularization is tested using synthetic and a real-world dataset and compared with results without regularization.

3.1 Introduction

Principal components analysis is the best known and widely used technique for multivariate analysis. The central idea of principal components analysis is to reduce the dimensionality of a dataset in which there are a large number of interrelated variables, while retaining as much as possible of the variation present in the dataset (Jolliffe, 2004). Its applications include exploratory data analysis, visualization, denoising and feature selections (Hastie et al., 2001; Bishop, 2006).

In the real-valued variables, the derivations and properties of principal components are based on the eigen-structure of the covariance matrix. Principal components are com-

monly defined as $\mathbf{x}_i = \mathbf{W}^T(\mathbf{y}_i - \boldsymbol{\mu})$ where $\mathbf{W} = (\tilde{\mathbf{w}}_1, \cdots, \tilde{\mathbf{w}}_k)$ has k columns as the first k eigenvectors of the covariance matrix, called principal component loadings or directions, and $\boldsymbol{\mu}$ is mean vector as an intercept term. If we use the first k eigenvectors for $\mathbf{W}$, then the new expression $\boldsymbol{\theta}_i = \boldsymbol{\mu} + \mathbf{W}\mathbf{x}_i = \boldsymbol{\mu} + x_{i1}\tilde{\mathbf{w}}_1 + \cdots + x_{ik}\tilde{\mathbf{w}}_k$ can be viewed as the orthogonal projection of $\mathbf{y}_i$ onto the k-dimensional subspace in which the projected points retain the maximal variability of the data points in the original space. This implies that variabilities along the orthogonal direction to this subspace are minimized. Therefore, such principal subspace can be found, without relying on the eigen-structure of covariance matrix, by directly looking for the subspace spanned by $\mathbf{W}$ and translated by the intercept $\boldsymbol{\mu}$. Components $\mathbf{W}$ and $\boldsymbol{\mu}$ may be derived by minimizing $\sum_{i=1}^{n} ||\mathbf{y}_i - (\boldsymbol{\mu} + \mathbf{W}\mathbf{x}_i)||^2$. Such minimization criterion is equivalent to maximizing Gaussian likelihood with an isotropic covariance (identity covariance matrix) and mean $\boldsymbol{\theta}_i$ lying on the k-dimensional subspace. This probabilistic interpretation motivates the model-based principal components analysis (Bishop and Tipping, 1998; Tipping and Bishop, 1999). This model-based approach for PCA, however, has a limitation. While estimate of $\mathbf{W}$ from the maximum Gaussian likelihood correctly find the principal subspace, its columns are not identical to the first k eigenvectors of covariance matrix because the estimate is subject to a rotation, as commonly appeared in factor analysis.

This model-based approach of PCA can be generalized to special types of data other than real-valued variables. Considering data types, we can deploy the distribution conforming such variables. For example, one may use Bernoulli distribution for binary variables and binomial or Poisson for count data. Generally, any exponential family distribution can be substituted instead of Gaussian distribution, and corresponding canonical parameters, mean parameters in Gaussian distribution case, are assumed to reside in the k-dimensional subspace embedded in the original d-dimensional space. Collins et al. (2001) studied a generalization of PCA to the exponential distribution in this direction and, in their approach,

principal components are treated as a fixed parameters as well as principal component loadings. This approach has a drawback that the number of parameters to be estimated becomes large so the parameter estimation suffers from over-fitting in the modeling sense.

In this chapter, we develop PCA for binary variables in the latent variable model approach using Bernoulli distribution. In order to reduce the number of parameters we treat the principal components as random variables, serving latent variables in the model. Therefore, the intercept term and principal component loadings are estimated as unknown parameters, and principal component scores are predicted as the conditional expectation given binary variable data. The resulting model becomes a generalized linear mixed effect model, which has been widely studied in statistics. It is well known that the marginal likelihood, by integrating out the latent variables, does not have a closed-form expression in a generalized linear mixed effect model, so the approximation technique is necessary for its implementation. We employ the variational method to approximate the marginal likelihood in which the estimation procedure gives a closed-form solution in EM framework and its resulting form becomes a weighted least squares solution.

Although our interest is mainly on binary data, we also consider other types of variables, binomial and normal variables, in the PCA model and we provide a unified estimation procedure in the case where binary, binomial and normal variables appear together in a single dataset. Incorporating various type variables in the latent variable model has been studied in generalized latent trait models (Moustaki and Knott, 2000; Huber et al., 2004). However, existing methodologies to estimate parameters are not satisfactory when we analyze a high dimensional dataset because their techniques to approximating likelihood function are not computationally feasible in the high dimensional situation. Comparing such methodologies, our proposed algorithm using variational method can be successfully applied to analyzing high dimensional dataset.

While principal components analysis has been proved to be useful in many appli-

cations, its interpretation of principal component loadings is often difficult since lots of nonzero loadings are involved. So we deploy L_1 regularization to force negligible nonzero loadings to be zero so that the derived principal component loadings have simple structure. It is well known that bias introduced by the regularization will reduce the variance of estimates so that the performance of prediction is improved and estimates becomes more stable. A model-based approach is not free from the model identifiability due to loading rotation, as in the model-based standard PCA and factor model. Another benefit from L_1 regularization is that the model does not suffer from loading rotation so that the estimated principal component loadings are close to the true principal component loadings, as will be shown in simulation study.

3.2 PCA Model for Binary Variables with Regularization

3.2.1 Latent variable model

Suppose we have d-dimensional binary response vector $\mathbf{y} = (y_1, \cdots, y_d)^T$. Natural distribution assumption of binary variables is Bernoulli distribution with success probabilities, π_j ($j = 1, \ldots, d$). We model π_js in the logit scale, which are often called canonical parameters, denoted by the d-dimensional vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)^T$ with $\theta_j = \log\{\pi_j/(1-\pi_j)\}$. This canonical parameter $\boldsymbol{\theta}$ is modeled as a linear combination of basis vectors, $\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_k$ and the intercept term $\boldsymbol{\mu}$, giving

$$\boldsymbol{\theta} \;=\; \boldsymbol{\mu} + x_1\tilde{\mathbf{w}}_1 + \cdots + x_k\tilde{\mathbf{w}}_k \;=\; \boldsymbol{\mu} + \mathbf{W}\mathbf{x} \tag{3.1}$$

with $\mathbf{x} = (x_1, \ldots, x_k)^T$ and $\mathbf{W} = (\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_k)$. A set of k basis vectors $\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_k$ are called principal components. These basis vectors are commonly assumed to be orthogonal in standard principal components analysis, however, we will relax the orthogonal constraint later since regularization on principal component loadings makes the orthogonal constraint

inappropriate. The coefficients $x_1, \ldots, x_k$ in the model (3.1) are called principal component scores, which are treated as random variables in this model. The latent variable $\mathbf{x}$ is assumed to be normally distributed with the zero mean and the identity covariance matrix, as in probabilistic principal components analysis for continuous variables (Tipping and Bishop, 1999). Therefore, variabilities of binary variable $\mathbf{y}$ are modeled in the canonical parameter space and k principal components represent the mode of variabilities. With the Gaussianity assumption on $\mathbf{x}$, the model (3.1) is known as generalized linear mixed effect model, which is widely considered and extensively studied in statistics area (McCulloch and Searle, 2001).

Tipping (1999) used the same probabilistic model for visualization of binary data only in the 2-dimensional representation for visualization purpose. In this study we generalize it to k-component representation and discuss the selection of the number of components in the subsequent arguments. Most model-based approaches for principal components analysis have a limitation that the proposed model is not identifiable due to the rotation of principal component loading matrix $\mathbf{W}$ and, thus, resulting solution under the model suffers from such rotational indeterminacy. In order to look at this aspect, consider any orthogonal or rotation matrix $\mathbf{H}$ satisfying $\mathbf{H}^T\mathbf{H} = \mathbf{H}\mathbf{H}^T = \mathbf{I}_k$. Then from model (3.1),

$$\boldsymbol{\mu} + \mathbf{W}\mathbf{x} \;=\; \boldsymbol{\mu} + \mathbf{W}\mathbf{H}^T\mathbf{H}\mathbf{x} \;=\; \boldsymbol{\mu} + \mathbf{W}^*\mathbf{x}^*$$

with $\mathbf{W}^* = \mathbf{W}\mathbf{H}^T$ and $\mathbf{x}^* = \mathbf{H}\mathbf{x}$. From the assumption $\mathbf{x} \sim N(\mathbf{0}, \mathbf{I}_k)$, it follows that $\mathbf{x}^* = \mathbf{H}\mathbf{x} \sim N(\mathbf{0}, \mathbf{I}_k)$. Therefore, two different model parameters $\mathbf{W}$ and $\mathbf{W}^*$ lead to the same model so the proposed model (3.1) is not identifiable. The same problem also commonly appears in factor model. In order to make the model identifiable, it is necessary to impose some restriction on the form of estimate of $\mathbf{W}$. For the principal components analysis, the orthogonality constraint on principal components is desirable. In factor analysis, the "best" of these rotated solutions is chosen according to some particular criterion, such as varimax

or oblique rotation. These selection procedures are usually conducted after parameters are estimated. In this sense, orthogonalization and factor rotation form a post-processing, which is done outside the estimation procedure. Later, we introduce L_1 regularization in the estimation procedure and it turns out that the estimate of parameters is uniquely determined up to sign change during the estimation step.

Since the given model involves the latent variables, log likelihood is obtained by integrating out the joint distribution over the latent variables. Suppose we have n independent d-dimensional binary vectors, $\mathbf{y}_1, \ldots, \mathbf{y}_n$. The log likelihood is written as

$$\begin{aligned} \ell(\boldsymbol{\Theta}) &= \sum_{i=1}^{n} \log \int P(\mathbf{y}_i, \mathbf{x}_i; \boldsymbol{\mu}, \mathbf{W}) d\mathbf{x}_i \\ &= \sum_{i=1}^{n} \log \int \prod_{j=1}^{d} P(y_{ij}|\mathbf{x}_i; \boldsymbol{\mu}, \mathbf{W}) P(\mathbf{x}_i) d\mathbf{x}_i \end{aligned} \tag{3.2}$$

where $\boldsymbol{\Theta}$ denotes all parameters, $\boldsymbol{\mu}$ and $\mathbf{W}$, collectively, $P(y_{ij}|\mathbf{x}_i; \boldsymbol{\mu}, \mathbf{W})$ is the probability mass function of Bernoulli distribution with the success probability $\exp(\boldsymbol{\mu} + \mathbf{W}\mathbf{x}_i)/\{1 + \exp(\boldsymbol{\mu} + \mathbf{W}\mathbf{x}_i)\}$ and $P(\mathbf{x}_i)$ is the density of k-variate standard Gaussian distribution. This log likelihood does not have a closed-form expression, which motivates to use the approximation techniques for estimation procedure.

3.2.2 L_1 regularization

The interpretation of principal component loadings is not an easy task because there are usually lots of nonzero loadings involved. In the standard principal components analysis, there have been several attempts to make the principal component loadings have the sparse structure by regularization for the simple interpretation (Jolliffe et al., 2003; Zou et al., 2006; Shen and Huang, 2008). To this end, for binary data we propose to impose L_1 penalty on the principal component loading estimation. L_1 regularization technique has been widely studied and used, especially in regression-type problems, not only for the simple structure of parameter estimation, also for better prediction by reducing the variance

of estimates. In binary PCA model, L_1 penalty is employed to the log likelihood function, whose form is

$$\begin{aligned} P(\mathbf{W}) &= \eta_1||\tilde{\mathbf{w}}_1||_1 + \cdots + \eta_k||\tilde{\mathbf{w}}_k||_1 \\ &= \eta_1 \sum_{j=1}^{d} |w_{j1}| + \cdots + \eta_k \sum_{j=1}^{d} |w_{jk}| \end{aligned} \tag{3.3}$$

where positive values $\eta_1, \ldots, \eta_k$ are the regularization parameters controlling the model complexity. L_1 penalty is applied to columns of $\mathbf{W}$, principal components, one by one. If regularization parameter gets larger the model becomes simpler by giving the small number of nonzero loadings, while fit to the data becomes worse due to the bias from the rigid model structure. If regularization parameter is set to zero, parameter estimates are in a free form so that the typical maximum likelihood estimators are retained.

Since L_1 penalty applies to each principal component loading, we should optimize all regularization parameters $\eta_1, \ldots, \eta_k$ for model selection. This is an unattractive aspect in implementation because the grid search requires considerable computing time. Instead of considering separate regularization parameters, we propose to use a single regularization parameter $\eta_1 = \cdots = \eta_k = \eta$ for the computational efficiency. Therefore, k principal component loadings are regulated by a single parameter η. Beside the computational economy, another benefit from using a single regularization parameter is that this lenders an automatic procedure to select the number of principal components. Since the same amount of penalization is applied to all principal component loadings, all loading values of negligible principal component are shut down to zero but important component still remain to have nonzero loadings.

By invoking L_1 regularization, therefore, the objective function to be maximized is the penalized log likelihood function given as

$$\ell_p(\mathbf{\Theta}) \;=\; \ell(\mathbf{\Theta}) - nP(\mathbf{W}). \tag{3.4}$$

Therefore maximum penalized likelihood estimates will be derived under the balance between the maximal model fit to data and the simple structure on principal component loadings.

Another important feature of L_1 regularization is that the penalty function (3.3) is not invariant under rotation. In other words, $P(\mathbf{W}\mathbf{H}^T) \neq P(\mathbf{W})$ for any rotation matrix $\mathbf{H}$ except for a permutation matrix. This explains the solution of the model (3.1) with L_1 regularization is unique without indeterminacy from the rotation. The penalty function (3.3) can be rearranged as

$$P(\mathbf{W}) \;=\; \eta \sum_{m=1}^{k} |w_{1m}| + \cdots + \eta \sum_{m=1}^{k} |w_{dm}|$$

using a single regularization parameter. Each component $\sum_{m=1}^{k} |w_{jm}|$ in the right-hand side is the sum of the absolute values of k principal component loadings of the jth variable, which is corresponding to the jth row of $\mathbf{W}$. In geometrical sense, when k principal component loadings for the jth variable is depicted as a point in the k dimensional space, this can be interpreted a sum of the distances of k axes from that point. Since, among many rotated candidates, L_1 regularization prefers one that gives the minimum distances from axes, lots of loadings of such solution are close to axes and small number of loadings have large values, as illustrated in Figure 13. This is the similar strategy as varimax rotation criterion in factor analysis (Kaiser, 1958). Varimax rotation chooses a principal component loading matrix $\mathbf{W}$ which maximizes

$$Q \;=\; \sum_{m=1}^{k} \Bigg[\sum_{j=1}^{d} w_{jm}^4 - \frac{1}{d} \Bigg(\sum_{j=1}^{d} w_{jm}^2 \Bigg)^2 \Bigg]. \tag{3.5}$$

This provides axes with a few large loadings and as many near-zero loadings as possible. Although varimax uses a different criterion that the solution is chosen to maximize the sum of variances of squared loadings for each rotated factor, its effect is similar with minimizing L_1 penalty. Therefore, we expect that L_1 regularization leads to the estimate

similar to that of varimax criterion. However, contrast to varimax criterion, the solution from L_1 penalization has lots of exact zero loadings. And, moreover, L_1 penalized solution is automatically sought during the estimation procedure, not in post-processing step after finishing the estimation. Once $\mathbf{W}$ is estimated by maximizing the penalized log likelihood, we reorder columns of $\mathbf{W}$ by their magnitudes. This will determine the estimate of principal component loadings uniquely up to only sign change, which do not have a practical importance.

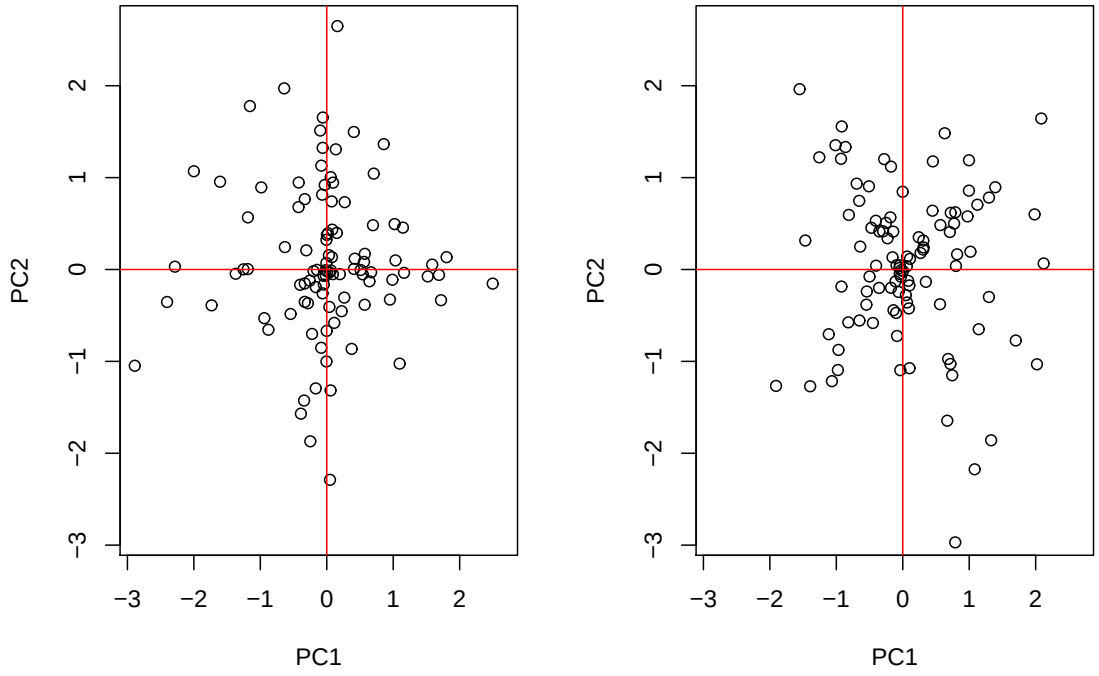


Figure 13: Illustrative example for principal component rotation. PC loadings appearing in the left panel shows the smaller L_1 penalty than those in the right panel. One of two principal component loadings can be derived by rotating the other, so that the likelihoods from two principal component loadings are the same.

3.3 Variational Learning Algorithm

In this section, we formulate the variational learning algorithm for the latent variable model for the binary principal components analysis.

3.3.1 Classical EM formulation

Since the model includes latent variables and marginal log likelihood is computationally intractable, the EM algorithm may be useful for parameter estimation. Regarding latent variables $\mathbf{x}_i$ as missing variables, the complete log likelihood becomes

$$\ell_c(\boldsymbol{\Theta}) \;=\; \sum_{i=1}^{n} \log P(\mathbf{y}_i, \mathbf{x}_i; \boldsymbol{\Theta}) \;=\; \sum_{i=1}^{n} \Big\{ \sum_{j=1}^{d} \log P(y_{ij}|\mathbf{x}_i; \boldsymbol{\mu}, \mathbf{W}) + \log P(\mathbf{x}_i) \Big\}$$

Maximizing the conditional expectation of the complete log likelihood, $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^0) = E[\ell_c(\boldsymbol{\Theta})|\mathbf{Y}; \boldsymbol{\Theta}^0]$, increases log likelihood function sequentially. Here the conditional expectation is conducted over the latent variables $\mathbf{x}_i$ conditionally on the observed data $\mathbf{Y}$ with the previous estimate $\boldsymbol{\Theta}^0$. Therefore, the maximum penalized likelihood estimator is attained by maximizing the surrogate function

$$Q_p(\boldsymbol{\Theta}|\boldsymbol{\Theta}^0) \;=\; Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^0) - nP(\mathbf{W})$$

sequentially.

Main difficulty in applying the EM algorithm is that the configuration of the conditional distribution of $\mathbf{x}_i$ given the data $\mathbf{y}_i$ is computationally infeasible, so the conditional expectation $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^0)$ is not available. To approximate the E-step, some numerical approximation techniques, such as Gauss-Hermite quadrature or Monte-Carlo EM, have been used in similar latent variable models (Samel et al., 1997; Moustaki and Knott, 2000). Such approximation approaches are computationally infeasible in the high dimensional setup. Instead we propose to use the variational method to approximate the marginal likelihood, which enables us to enjoy the closed-form expression in the EM algorithm.

3.3.2 Variational lower bound to the marginal likelihood

The motivation of the variational method is to substitute convenient surrogate for complicated marginal likelihood. Such surrogate function may not be precise, but its form is computationally convenient. Jaakkola and Jordan (1997, 2000) introduced a variational method to approximate the predictive distribution in a Bayesian logistic regression model and Tipping (1999) applied it to approximate the marginal distribution in the visualization of binary data.

From (1.9) in Section I, the conditional distribution of y_{ij} given $\mathbf{x}_i$, $P(y_{ij}|\mathbf{x}_i; \boldsymbol{\mu}, \mathbf{W})$, can be approximated by

$$\tilde{P}(y_{ij}|\mathbf{x}_i; \boldsymbol{\mu}, \mathbf{W}, \boldsymbol{\xi}_{ij}) \;=\; \pi(\xi_{ij}) \exp\Big[\{(2y_{ij}-1)\theta_{ij} - \xi_{ij}\}/2 - \lambda(\xi_{ij})(\theta_{ij}^2 - \xi_{ij}^2)\Big] \quad (3.6)$$

where $\theta_{ij} = \mu_j + \mathbf{w}_j^T \mathbf{x}_i$ is the jth component of $\boldsymbol{\theta}_i = \boldsymbol{\mu} + \mathbf{W}\mathbf{x}_i$ and $\mathbf{w}_j$ is the jth row of $\mathbf{W}$, and $\lambda(x) = \{\pi(x) - 1/2\}/2x$ (Jaakkola and Jordan, 1997, 2000; Tipping, 1999). Extra parameters ξ_{ij}s are called variational parameters. This approximation (3.6) serves as a lower bound of the conditional distribution so that $P(y_{ij}|\mathbf{x}_i; \boldsymbol{\mu}, \mathbf{W}) \geq \tilde{P}(y_{ij}|\mathbf{x}_i; \boldsymbol{\mu}, \mathbf{W}, \xi_{ij})$. This bound is exact when $\xi_{ij} = (2y_{ij} - 1)\theta_{ij}$. When we put this variational lower bound of the conditional distribution in the likelihood, we have a lower bound for the log likelihood (3.2) by

$$\tilde{\ell}(\boldsymbol{\Theta}, \boldsymbol{\xi}) \;=\; \sum_{i=1}^{n} \log \int \prod_{j=1}^{d} \tilde{P}(y_{ij}|\mathbf{x}_i; \boldsymbol{\mu}, \mathbf{W}, \xi_{ij}) P(\mathbf{x}_i) d\mathbf{x}_i, \quad (3.7)$$

satisfying $\ell(\boldsymbol{\Theta}) \geq \tilde{\ell}(\boldsymbol{\Theta}, \boldsymbol{\xi})$. Since the exponential in (3.6) is quadratic in $\mathbf{x}_i$, the integral in (3.7), then, can be computed in the closed form. This suggests the surrogate function maximization in the iterative manner. To do this, first we optimize $\boldsymbol{\xi}$ to achieve the closest approximation of $\ell(\boldsymbol{\Theta})$ by $\tilde{\ell}(\boldsymbol{\Theta}, \hat{\boldsymbol{\xi}})$, then we maximize $\tilde{\ell}(\boldsymbol{\Theta}, \hat{\boldsymbol{\xi}})$ over model parameters $\boldsymbol{\Theta}$.

3.3.3 Variational approximation to the conditional distribution of $\mathbf{x}_i$ given $\mathbf{y}_i$

The maximization of (3.7) is still difficult since the maximization is not in the convex optimization. To relax such complexity, the EM algorithm can be applied to the maximization of (3.7), which requires to compute the conditional expectation of (3.7) given the observed data $\mathbf{Y}$. This conditional expectation of $\tilde{\ell}$, denoted by $\tilde{Q}$ here, involves only first two moment of $\mathbf{x}_i|\mathbf{y}_i$. However, their exact computations are complicated, so we approximate the conditional distribution of $\mathbf{x}_i$ given $\mathbf{y}_i$ by using (3.6). It should be noted that the lower bound (3.6) is not a proper distribution since it is not normalized. After normalization, the lower bound becomes a Gaussian distribution which we call the variational approximated conditional distribution of y_{ij} given $\mathbf{x}_i$.

From (3.6), the log of the variational approximated conditional distribution of $y_{ij}|\mathbf{x}_i$ is

$$\begin{aligned}\log \tilde{P}(y_{ij}|\mathbf{x}_i;\xi_{ij}) &= \log \pi(\xi_{ij}) + \frac{(2y_{ij}-1)(\mu_j + \mathbf{w}_j^T\mathbf{x}_i) - \xi_{ij}}{2} \\ &\quad -\lambda(\xi_{ij})\{\mu_j^2 - 2\mu_j\mathbf{w}_j^T\mathbf{x}_i + \mathbf{x}_i^T\mathbf{w}_j\mathbf{w}_j^T\mathbf{x}_i - \xi_{ij}^2\}.\end{aligned}$$

By using the conditional independence assumption, we get

$$\begin{aligned}\log \tilde{P}(\mathbf{y}_i|\mathbf{x}_i;\boldsymbol{\xi}_i) &= \sum_{j=1}^{d} \log \tilde{P}(y_{ij}|\mathbf{x}_i;\xi_{ij}) \\ &= -\frac{1}{2}\mathbf{x}_i^T\left\{2\sum_{j=1}^{d}\lambda(\xi_{ij})\mathbf{w}_j\mathbf{w}_j^T\right\}\mathbf{x}_i + \sum_{j=1}^{d}\{y_{ij} - 1/2 - 2\mu_j\lambda(\xi_{ij})\}\mathbf{w}_j^T\mathbf{x}_i \\ &\quad + \sum_{j=1}^{d}\left\{\log\pi(\xi_{ij}) + \frac{(2y_{ij}-1)-\xi_{ij}}{2} - \lambda(\xi_{ij})(\mu_j^2 0\xi_{ij}^2)\right\}.\end{aligned}$$

Thus, the log of the joint distribution of $\mathbf{y}_i$ and $\mathbf{x}_i$ is given by

$$\begin{aligned}
&\log \tilde{P}(\mathbf{y}_i, \mathbf{x}_i; \boldsymbol{\xi}_i) = \log \tilde{P}(\mathbf{y}_i|\mathbf{x}_i; \boldsymbol{\xi}_i) + \log P(\mathbf{x}_i) \\
&= -\frac{1}{2}\mathbf{x}_i^T \left\{ \mathbf{I}_k + 2\sum_{j=1}^{d} \lambda(\xi_{ij})\mathbf{w}_j\mathbf{w}_j^T \right\} \mathbf{x}_i + \sum_{j=1}^{d} \big\{ y_{ij} - 1/2 - 2\mu_j\lambda(\xi_{ij}) \big\} \mathbf{w}_j^T \mathbf{x}_i \\
&+ \sum_{j=1}^{d} (y_{ij} - 1/2)\mu_j + \sum_{j=1}^{d} \big\{ \log \pi(\xi_{ij}) - \xi_{ij}/2 - \lambda(\xi_{ij})(\mu_j^2 - \xi_{ij}^2) \big\} - \frac{k}{2}\log 2\pi \\
&= -\frac{1}{2}(\mathbf{x}_i - \mathbf{m}_i)^T \mathbf{C}_i^{-1} (\mathbf{x}_i - \mathbf{m}_i) + \frac{1}{2}\mathbf{m}_i^T \mathbf{S}_i^{-1} \mathbf{m}_i + \sum_{j=1}^{d} (y_{ij} - 1/2)\mu_j \\
&+ \sum_{j=1}^{d} \big\{ \log \pi(\xi_{ij}) - \xi_{ij}/2 - \lambda(\xi_{ij})(\mu_j^2 - \xi_{ij}^2) \big\} - \frac{k}{2}\log 2\pi.
\end{aligned}$$

Therefore, with this variational approximation and Bayes' rule, the approximation of $P(\mathbf{x}_i|\mathbf{y}_i)$ is a Gaussian distribution with mean $\mathbf{m}_i$ and covariance $\mathbf{C}_i$ where

$$\begin{aligned}
\mathbf{C}_i &= \left[\mathbf{I}_k + 2\sum_{j=1}^{d} \lambda(\xi_{ij})\mathbf{w}_j\mathbf{w}_j^T \right]^{-1} \\
\mathbf{m}_i &= \mathbf{C}_i \left[\sum_{j=1}^{d} \left\{ y_{ij} - \frac{1}{2} - 2\lambda(\xi_{ij})\mu_j \right\} \mathbf{w}_j \right].
\end{aligned}$$

Using the above, we can compute the first two moments of the conditional distribution of $\mathbf{x}_i|\mathbf{y}_i$ as

$$\begin{aligned}
\langle \mathbf{x}_i \rangle = E(\mathbf{x}_i|\mathbf{y}_i) &= \mathbf{m}_i \\
\langle \mathbf{x}_i\mathbf{x}_i^T \rangle = E(\mathbf{x}_i\mathbf{x}_i^T|\mathbf{y}_i) &= \mathbf{C}_i + \mathbf{m}_i\mathbf{m}_i^T
\end{aligned} \tag{3.8}$$

which will be used in the E-step of the EM algorithm.

3.3.4 Variational approximation to the penalty function

While L_1 regularization has good properties discussed in the previous section, its penalty function is non-differentiable so the optimization is somewhat computationally challenging. Tibshirani (1996) proposed to use quadratic programming in the seminal paper on L_1

regularization. And LARS algorithm is known to solve L_1 regularization problem in the regression setting (Efron et al., 2004). In this study, we propose an analytic algorithm for L_1 penalty function, which is compatible with the variational method.

From the inequality, $|x| \leq (x^2 + y^2)/2|y|$, the penalty function (3.3) has a quadratic upper bound as

$$\tilde{P}(\mathbf{W}, \boldsymbol{\zeta}) \;=\; \eta \sum_{j=1}^{d} \sum_{m=1}^{k} \frac{w_{jm}^2 + \zeta_{jm}^2}{2|\zeta_{jm}|} \;=\; \sum_{j=1}^{d} \Big(\mathbf{w}_j^T \boldsymbol{\Omega}_j \mathbf{w}_j + \boldsymbol{\zeta}_j^T \boldsymbol{\Omega}_j \boldsymbol{\zeta}_j \Big) \tag{3.9}$$

where $\boldsymbol{\Omega}_j = \text{diag}(\eta/2|\zeta_{jm}|)_{m=1,\ldots,k}$. Here additional parameters ζ_{jm} are variational parameters and the upper bound is exact when $\zeta_{jm} = w_{jm}$. This quadratic upper bound for penalty function can be combined nicely with the maximization of (3.7) in the estimation procedure.

3.3.5 Estimation algorithm

Using variational quadratic bounds given in (3.7) and (3.9), the variational lower bound of the penalized log likelihood (3.4) becomes

$$\tilde{\ell}_p(\boldsymbol{\Theta}, \boldsymbol{\xi}, \boldsymbol{\zeta}) \;=\; \tilde{\ell}(\boldsymbol{\Theta}, \boldsymbol{\xi}) - n\tilde{P}(\mathbf{W}, \boldsymbol{\zeta}).$$

This is maximized by employing the EM algorithm. In the E-step, the conditional expectation of $\tilde{\ell}_p(\boldsymbol{\Theta}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ becomes

$$\begin{aligned}
\tilde{Q}_p(\boldsymbol{\Theta}|\boldsymbol{\Theta}^0) \;&=\; E\big[\tilde{\ell}(\boldsymbol{\Theta}, \boldsymbol{\xi})|\mathbf{Y}, \boldsymbol{\Theta}^0\big] - n\tilde{P}(\mathbf{W}, \boldsymbol{\zeta}) \\
&=\; \sum_{i=1}^{n} \Bigg[\sum_{j=1}^{d} \Big\{ \log \pi(\xi_{ij}) + \frac{(2y_{ij} - 1)(\mathbf{w}_j^T \langle \mathbf{x}_i \rangle + \mu_j) - \xi_{ij}}{2} \\
&\quad -\lambda(\xi_{ij})(\mathbf{w}_j^T \langle \mathbf{x}_i \mathbf{x}_i^T \rangle \mathbf{w}_j + 2\mu_j \langle \mathbf{x}_i \rangle^T \mathbf{w}_j + \mu_j^2 - \xi_{ij}^2) \Big\} - \frac{k}{2} \log 2\pi - \frac{1}{2} \langle \mathbf{x}_i^T \mathbf{x}_i \rangle \Bigg] \\
&\quad -n \sum_{j=1}^{d} \Big\{ \mathbf{w}_j^T \boldsymbol{\Omega}_j \mathbf{w}_j + \boldsymbol{\zeta}_j^T \boldsymbol{\Omega}_j \boldsymbol{\zeta}_j \Big\}.
\end{aligned} \tag{3.10}$$

Before optimizing model parameters, we first optimize variational parameters $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ to make the bound tight. Taking the derivative of $\tilde{Q}_p$ with respect to ξ_{ij} and setting it to zero leads to

$$\begin{aligned}\frac{\partial \tilde{Q}_p}{\partial \xi_{ij}} &= \frac{\pi'(\xi_{ij})}{\pi(\xi_{ij})} - \frac{1}{2} - \lambda'(\xi_{ij})\big(\mathbf{w}_j^T\langle \mathbf{x}_i\mathbf{x}_i^T\rangle \mathbf{w}_j + 2\mu_j\langle \mathbf{x}_i\rangle^T\mathbf{w}_j + \mu_j^2 - \xi_{ij}^2\big) + 2\lambda(\xi_{ij})\xi_{ij} \\ &= -\lambda'(\xi_{ij})\big(\mathbf{w}_j^T\langle \mathbf{x}_i\mathbf{x}_i^T\rangle \mathbf{w}_j + 2\mu_j\langle \mathbf{x}_i\rangle^T\mathbf{w}_j + \mu_j^2 - \xi_{ij}^2\big) \\ &= 0\end{aligned}$$

where we used $\pi'(\xi_{ij}) = \pi(\xi_{ij})\{1-\pi(\xi_{ij})\}$ and $\lambda(\xi_{ij}) = \{\pi(\xi_{ij}) - 1/2\}/2\xi_{ij}$. Since $\lambda(\cdot)$ is symmetric about zero and is monotonically decreasing over the positive domain, $\lambda'(\xi_{ij})$ cannot be zero in the positive domain, so the maximum is obtained at

$$\hat{\xi}_{ij} = \sqrt{\mathbf{w}_j^T\langle \mathbf{x}_i\mathbf{x}_i^T\rangle \mathbf{w}_j + 2\mu_j\langle \mathbf{x}_i\rangle^T\mathbf{w}_j + \mu_j^2}.$$

Similarly, for another variational parameter η_{jm},

$$\frac{\partial \tilde{Q}_p}{\partial \zeta_{jm}} = \frac{\eta \cdot \text{sgn}(\zeta_{jm})}{2\zeta_{jm}^2}\big(\zeta_{im}^2 - w_{jm}^2\big) = 0$$

which gives $\hat{\zeta}_{jm} = |w_{jm}|$. Once $\hat{\xi}_{ij}$ and $\hat{\zeta}_{jm}$ are optimized, we compute conditional expectations $\langle \mathbf{x}_i\rangle$ and $\langle \mathbf{x}_i\mathbf{x}_i^T\rangle$ using the formulae in (3.8) with the previous estimates and the optimized variational parameters. Then, we update the parameters by maximizing $\tilde{Q}_p$. This gives update formulae as

$$\begin{aligned}\hat{\mu}_j &= \sum_{i=1}^n \left\{\frac{2y_{ij}-1}{4} - \lambda(\xi_{ij})\langle \mathbf{x}_i\rangle^T\mathbf{w}_j\right\}\Big/\sum_{i=1}^n \lambda(\xi_{ij}), \\ \hat{\mathbf{w}}_j &= \left[\sum_{i=1}^n \lambda(\xi_{ij})\langle \mathbf{x}_i\mathbf{x}_i^T\rangle + n\boldsymbol{\Omega}_j\right]^{-1}\cdot\sum_{i=1}^n\left\{\frac{2y_{ij}-1}{4} - \mu_j\lambda(\xi_{ij})\right\}\langle \mathbf{x}_i\rangle.\end{aligned}$$

Estimation details are almost the same as in Tipping (1999), except that the solution $\hat{\mathbf{w}}_j$ includes the ridge-type penalty term inside the matrix inverse. Thus, non-differentiable problem of L_1 regularization turns into an analytic L_2 regularization with variational method.

3.4 Implementation Issues

3.4.1 Model selection

In the proposed model, model selection procedure involves two selection problems. One is the selection of the subspace dimensionality k and the other is the selection of the regularization parameter η. As a usual model selection, typical model selection criteria, such as AIC or BIC, may be used by adding the penalty to the negative twice log likelihood. In binary variables, however, the exact evaluation of log likelihood is not readily available. We would approximate the log likelihood by Monte-Carlo sampling approximation as

$$\ell_a(\mathbf{\Theta}) \;=\; \sum_{i=1}^{N} \log\left\{ \frac{1}{B} \sum_{b=1}^{B} \prod_{j=1}^{d} P(y_{ij}|\mathbf{x}_b, \boldsymbol{\mu}, \mathbf{W}) \right\}$$

where $\mathbf{x}_b$, $b = 1, \cdots, B$, are samples from the k-variate standard Gaussian distribution. We used $B = 1000$ in the following simulation studies and real data analysis. Both of AIC and BIC work well in large sample situation, but we observed that their performance is not satisfactory when the dimension is larger than or comparable to the sample size.

For the selection of η in high dimensional dataset, thus, we propose to use the corrected BIC defined as

$$BIC(\eta) \;=\; -2\ell(\mathbf{\Theta}) + \log n \times |\mathcal{B}(\eta)|$$

where $|\mathcal{B}(\eta)|$ is the number of nonzeros in whole parameter set. Therefore, we choose the optimal η which achieves the minimum of $BIC(\eta)$. For the selection of the subspace dimensionality, we first set a tentatively large k so that important principal components are not lost. One may use standard AIC for this since it usually chooses a conservative one. But the extra AIC procedure only for a tentative k is not very attractive computationally, so we suggest to use $k \approx d/5$ but it depends on specific situation. With this tentative k, we choose η using the corrected BIC. If a small number of principal components are important and remaining are negligible, all loadings associated with negligible principal

components will be forced to be zeros so that the number of important components will be automatically chosen by giving the number of principal components having nonzero loadings. This heuristic approach has been successfully proven in the simulation study.

3.4.2 Missing treatment

For missing values, we can still use the EM algorithm for missing imputation. Suppose (i, j)th binary variable, y_{ij}, is missing or unobservable. The conditional expectation of the penalized complete log likelihood (3.10) given the observed data $\mathbf{Y}$ with current estimates $\mathbf{\Theta}$, then, involves $E[y_{ij}|\mathbf{y}_i^*, \mathbf{\Theta}]$ where $\mathbf{y}_i^*$ is the observed variables for the ith individual removing unobserved variables, since $\mathbf{y}_i$ are assumed to be independent. Then it follows

$$\begin{aligned}
\langle y_{ij} \rangle &= E[y_{ij}|\mathbf{y}_i^*, \mathbf{\Theta}] = E\big[E[y_{ij}|\mathbf{x}_i, \mathbf{y}_i^*, \mathbf{\Theta}]|\mathbf{y}_i^*, \mathbf{\Theta}\big] \\
&= E\big[E[y_{ij}|\mathbf{x}_i, \mathbf{\Theta}]|\mathbf{y}_i^*, \mathbf{\Theta}\big] = E[\pi(\theta_{ij})|\mathbf{y}_i^*, \mathbf{\Theta}] \\
&= E\left[\frac{\exp(\theta_{ij})}{1+\exp(\theta_{ij})}\middle|\mathbf{y}_i^*, \mathbf{\Theta}\right] \qquad (3.11)
\end{aligned}$$

where $\theta_{ij} = \mu_j + \mathbf{w}_j^T\mathbf{x}_i$, and the third equality comes from the fact that all components of binary vector $\mathbf{y}_i$ are independent conditionally on $\mathbf{x}_i$. Since (3.11) is not in a closed-form expression, we may approximate it by the method introduced by Mackay (1992).

Suppose $\mathcal{O}_i$ is the index set that contains the js corresponding the observed data $\mathbf{y}_i^*$. Then, using Bayes' theorem, the variational approximated conditional distribution of $\mathbf{x}_i|\mathbf{y}_i^*$ is Gaussian with mean $\mathbf{m}_i^*$ and covariance $\mathbf{C}_i^*$ as

$$\begin{aligned}
\mathbf{C}_i^* &= \left[\mathbf{I}_k + 2\sum_{j\in\mathcal{O}_i}\lambda(\xi_{ij})\mathbf{w}_j\mathbf{w}_j^T\right]^{-1} \\
\mathbf{m}_i^* &= \mathbf{C}_i^*\left[\sum_{j\in\mathcal{O}_i}\left\{y_{ij} - \frac{1}{2} - 2\lambda(\xi_{ij})\mu_j\right\}\mathbf{w}_j\right].
\end{aligned}$$

Thus, $\theta_{ij} = \mu_j + \mathbf{w}_j^T\mathbf{x}_i$ is distributed normally with mean $\nu_{ij} = \mu_j + \mathbf{w}_j^T\mathbf{m}_i^*$ and variance $\gamma_{ij} = \mathbf{w}_j^T\mathbf{C}_i^*\mathbf{w}_j$. Mackay (1992) used the approximation that $\exp(\theta_{ij})/\{1+\exp(\theta_{ij})\} \approx$

$\Phi(\theta_{ij} \times \sqrt{\pi/8})$ where $\Phi(\cdot)$ is the cumulative standard Gaussian distribution function. Therefore the expression (3.11) can be approximated by

$$E\left[\frac{\exp(\theta_{ij})}{1+\exp(\theta_{ij})}\middle|\mathbf{y}_i^*, \mathbf{\Theta}\right] \approx E\left[\Phi(\theta_{ij}\sqrt{\pi/8})|\mathbf{y}_i^*, \mathbf{\Theta}\right] = \Phi\left(\frac{\mu_j + \mathbf{w}_j^T\mathbf{m}_i^*}{\sqrt{\mathbf{w}_j^T\mathbf{C}_i^*\mathbf{w}_j + 8/\pi}}\right).$$

3.5 Simulation Study

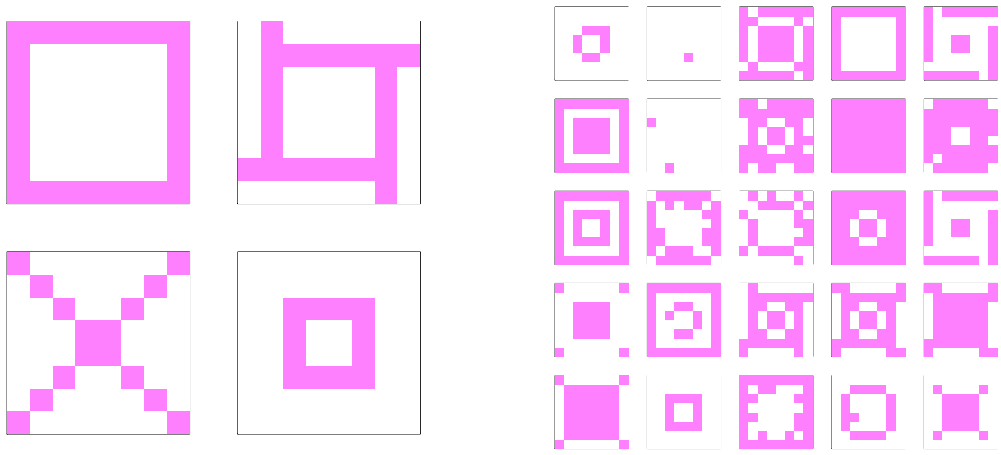
In this section, we evaluate binary PCA with latent variable model and its variational learning algorithm on two synthetic data sets constructed using latent variable model. The advantage of simulation study is that the true model as well as the true principal component loadings are known.

3.5.1 Simulation 1 : Synthetic binary images

The binary image datasets used in this experiment are generated by the latent variable model with 4 components. Each principal component loading pattern is associated with an 8×8 image pattern shown in (a) of Figure 14. All nonzero loadings are given by value 1, so that the magnitude of principal component is proportional to the number of nonzero spots. Using these components, 100 binary images are created using the latent variable model and used in analysis. Some examples of binary image data are presented in Figure 14(b).

In order to assess the performance of the regularization, we compare the results from the PCA with regularization and those without regularization. Since we know the true number of components we set the dimensionality of the subspace k by 4, the true subspace dimensionality, in this simulation.

Figure 15 shows the principal component loadings derived by the proposed algorithm with/without regularization. It is clear that the regularization greatly helps to construct loading patterns almost correctly. The derived principal loadings without regularization seem to also capture the original loading patterns but several patterns tend to appear to-



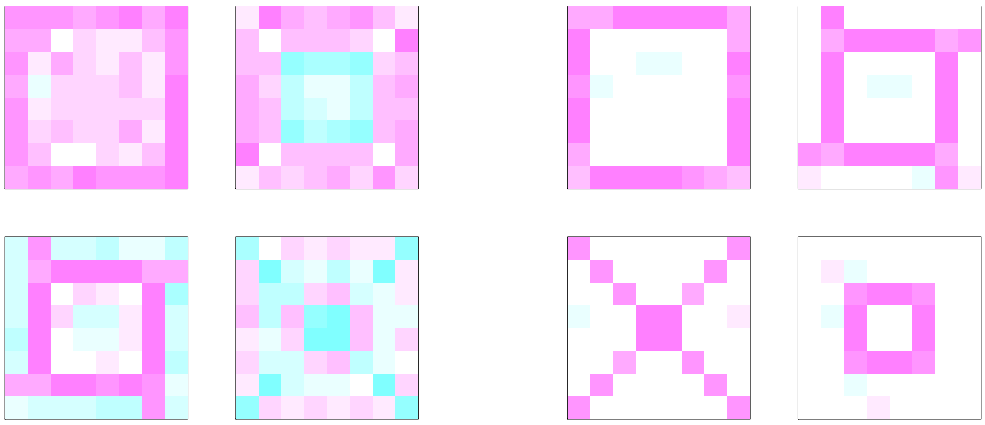
(a) (b)

Figure 14: Model reconstruction experiment 1. (a) Patterns associated with 4 principal component loadings used in the simulation. Red pixels denote nonzero loadings (b) Some binary images generated by the latent variable model with 4 components corresponding to patterns in (a) with zero background (white) and one foreground (red).

gether in a single principal component. This illustrates the estimated principal components suffer from the rotation indeterminacy. Comparing to the unregularized learning, it is apparent that each original loading pattern appears solely in a single principal component.

3.5.2 Simulation 2

In this experiment, we conduct the comparison in more systematical manner between PCA results with and without regularization. 100 binary data sets are generated from the latent variable models with 4 principal components in two different scenarios, $(n, d) = (200, 50)$ and $(100, 200)$, each of which mimics the large and low sample size situation. The original principal components are constructed in the sparse structure. Each principal component has all zero loadings except for the first 10 variables so that the first principal component has the same sized nonzero loading for the first 10 variables, and the second principal component has nonzero loading for the next 10 variables, and so on. Therefore, all loading w_{jm} are set to be zero except for $(j, m) = (1, 1), \cdots, (1, 10), (2, 11),$



(a) (b)

Figure 15: The derived principal component loading patterns (a) without regularization and (b) with regularization. Red and blue pixels stand for the positive and negative loadings respectively, and intensities are proportional to the magnitude of loadings. Zero loading is coded by white color.

$\cdots, (2, 20), (3, 21), \cdots, (3, 30), (4, 31), \cdots, (4, 40)$. Each column of $\mathbf{W}$, principal component, has the same-sized nonzero loadings and the magnitude of 4 principal components is set by $(40, 30, 20, 10)$ for $(n, d) = (200, 50)$ and $(80, 60, 40, 20)$ for $(n, d) = (100, 200)$ considering relative sample sizes. Therefore, when the dimension $d = 50$, the first $40(= 80\%)$ variables are effective to give the variability of binary variables and $10(= 20\%)$ variables do not affect the data variability. And in $d = 200$ case, $160(= 75\%)$ of 200 variables are unnecessary in explaining the variability. The intercept or shift parameter $\boldsymbol{\mu}$ is set by zero in this simulation.

In the real world, the original subspace dimensionality k is mostly unknown. So we conduct the model selection procedure to find k automatically as well as we present the result when k is known. And we also apply the proposed method to the same simulated data set with 10% randomly selected missing variables.

To assess the performance of the proposed methods, we compute and present the principal angle between spaces spanned by the original principal components and their esti-

Table 5: The results of binary PCA using 100 binary datasets consisting of 100 samples. Medians over 100 quantities are presented for each case. The description of this result is in the text.

		k is known			k is unknown		
(n, d)	Missing	angle	correct	incorrect	angle	correct	incorrect
	Regularization	(°)	(%)	(%)	(°)	(%)	(%)
$(200, 50)$	missing=0%						
	nonregularized	14.01	100.00	100.00	13.30	100.00	100.00
	regularized	6.31	100.00	90.00	6.66	100.00	86.67
	missing=10%						
	nonregularized	16.18	100.00	100.00	15.39	100.00	100.00
	regularized	6.12	100.00	90.00	7.22	100.00	80.00
$(100, 200)$	missing=0%						
	nonregularized	19.82	100.00	100.00	29.58	100.00	100.00
	regularized	4.28	100.00	29.38	4.01	100.00	15.00
	missing=10%						
	nonregularized	25.92	100.00	100.00	29.43	100.00	100.00
	regularized	5.20	100.00	32.50	5.58	100.00	8.75

mates. This principal angle is computed by $\cos^{-1}(\rho) \times 180/\pi$ where ρ is the minimum eigenvalue of matrix $\mathbf{Q}_1^T \mathbf{Q}_2$ with orthogonal matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ from the QR decomposition of the original principal component loading matrix $\mathbf{W}$ and its estimate $\widehat{\mathbf{W}}$ respectively. This quantity measures the maximum angle between any two vectors on column spaces of $\mathbf{W}$ and $\widehat{\mathbf{W}}$ (Golub and van Loan, 1996). Results are summarized in Table 5 presenting median value from 100 simulations.

It is apparent that the regularization greatly improves model assessment by finding the model that is much closer to the original model in all scenario. This result is expected because true zero loadings are usually estimated as nonzeros without regularization so that the subspace spanned by the derived non-sparse principal components becomes disturbed by such falsely detected nonzero variables. This disturbance will disappear when the original nonzero loadings are set to be zero correctly, as shown in the result with regularization. It is also interesting to note that the model with regularization shows the quite similar performances regardless of knowing the original subspace dimensionality. However, the model

assessment without regularization performs differently depending on whether k is known in advance or not, especially when the dimension is larger than the sample size. This illustrates that the regularized binary PCA model gives stable results even when we are not able to guess the true subspace dimensionality *a priori*.

We also present the percentage of the correctly and incorrectly identified nonzero loadings selected from the learning in Table 5. Without regularization, all loadings are estimated as nonzero, so that all true nonzero loadings are selected as nonzero correctly but also all zero loading variables are falsely detected as nonzero. Regularization, however, tends to force negligible loadings to zero while true nonzero variables remain in the model assessment. This aspect becomes remarkably apparent in high dimensional situation as presented in Table 5. And the performance of the proposed model and its estimation is still the same even in the situation where 10% binary variables are missing at random.

Table 6 shows the frequencies of the selected subspace dimensionality k among 100 simulation data sets when the original k is not known in advance. Most cases tend to find the original subspace dimensionality correctly, but it is noticed that some simulations select smaller k when $(n, d) = (100, 200)$ with 10% missingness. This phenomena may be explained that some missing binary variables associated with nonzero loading may seriously affect the model assessment so that corresponding important principal components become less important in the learning result.

Table 6: The frequencies of the selected subspace dimensions from 100 simulation data sets.

		Selected dimension		
(n, d)	Missing rate	3	4	5
$(200, 50)$	0%	0	98	2
	10%	0	100	0
$(100, 200)$	0%	1	99	0
	10%	14	86	0

3.6 Handwritten Digits Data Application

Our real-world example to which we apply the proposed PCA model is the handwritten digits data that come from the ZIP code on envelopes from U.S. Postal mail. Each image is a segment from a five digit ZIP code, isolating a single digit. The original scanned digits are binary and of different size and orientations. After deslanting and size-normalizeing, 16×16 images are obtained with gray scales ranging from -1 to 1 (Hastie et al., 2001). However, in order to get binary data, the values less than 0 is coded by 1 and others by 0 in this analysis. The dataset consists of $500 \sim 1,200$ images for each digit from 0 to 9, but in this analysis we use 556 images of digit 5. Therefore the sample size is $n = 556$ and the dimension is $d = 16 \times 16 = 256$. We apply the proposed algorithm to this dataset to identify the variabilities among binary images.

Figure 16 presents the first 4 principal components derived from the latent variable model for binary principal components analysis with and without regularization. To ease the interpretation and visualization, we depict the derived loadings in the original image format with color codings as blue and red representing the positive and negative loadings respectively and zero loadings are coded by white color. Their intensities of color are proportional to the magnitude of loadings. Apparently, principal components from learning with regularization show that lots of pixels (or loadings) are estimated as zero. This is contrasted to the estimated principal components without regularization, all of whose loading values are estimated nonzero. More importantly, each component from regularized PCA clearly represents a specific mode of variabilities among binary images. For example, the first component explains the variability of "roundedness" of tail part of digit so that observations with large value of the first principal component score will have "thin" tails and on the other hand observations with small value of it will show "round" tails. This is clearly observed in Figure 17(ab) where images with 5 largest and smallest values of the first prin-

cipal component score are presented. In the similar manner, other components may be easily interpreted. The second component explains the variability in the "head" part of digits, the third component presents the contrast of "tilt" of tails, and the fourth component reflects the variability from "height" of digits. Such modes of variabilities may be observed in the estimated principal components from the model without regularization, but disparate variabilities seem to simultaneously appear in the single component so that interpretation is less clear than in the regularized version. We cannot find such apparent contrasts when we look at the images with large and small principal component scores from Figure 18. This example illustrates how the regularization technique can help to improve the interpretability of estimates from learning and detect intrinsic features among binary data.

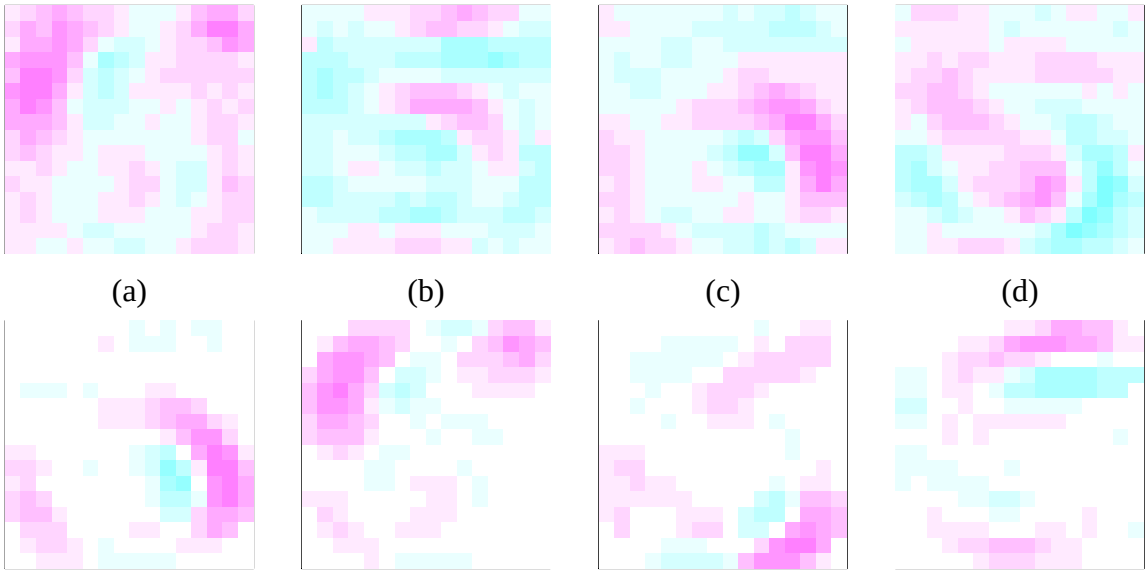


Figure 16: The derived PC loadings from handwritten digits data. (a)-(d) are the first 4 PC loadings estimated from the latent variable model for principal component analysis without regularization. (e)-(h) are those with regularization.

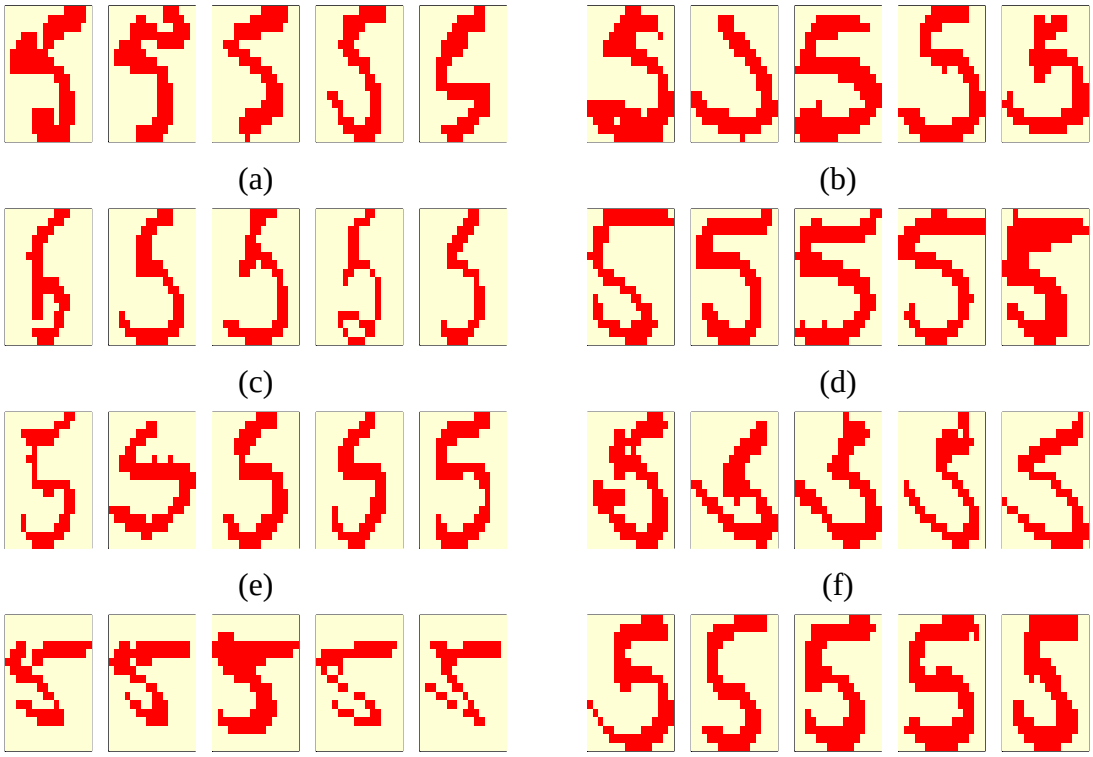


Figure 17: Binary digit images of digit 5. (a) and (b) are images that have the first 5 largest and smallest value of the first principal component score from the regularized binary PCA. Similarly, (c) and (d) corresponds for the second, (e) and (f) for the third, and (g) and (h) for the fourth principal component score.

3.7 Combining Other-Type Data

In this section, we discuss the possibility that other type data, including normal and binomial variables, can be combined with binary variables in principal components analysis. Such attempts to combine disparate variables have been extensively investigated in psychometrics area (Moustaki and Knott, 2000; Huber et al., 2004). We show normal and binomial variables can be put together coherently into the unified estimation procedure using the variational method.

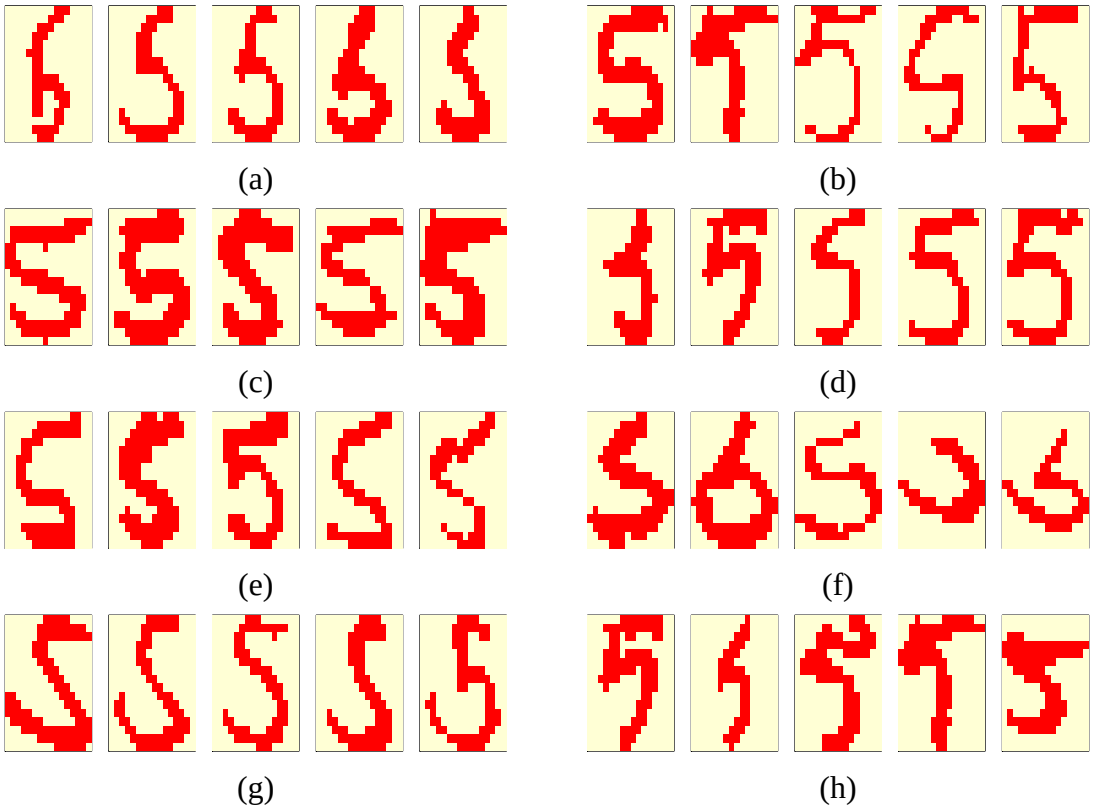


Figure 18: A counterpart of Figure 17 without regularization. Details are in Figure 17

3.7.1 Binomial variables

Suppose y_{ij} has binomial distribution with the number of binomial trials N_{ij} and the success probability $\pi_{ij} = \pi(\theta_{ij})$. Here, the canonical parameter, θ_{ij}, is defined as a logit of π_{ij}, as in binary case, and is assumed to be a linear form of the latent variable $\mathbf{x}_i$ as $\theta_{ij} = \mu_j + \mathbf{w}_j^T \mathbf{x}_i$ where $\mathbf{x}_i$ is also assumed to be normally distributed with zero mean and identity covariance as usual.

Then, the log of the probability mass function of y_{ij} given the latent variable $\mathbf{x}_i$ is written as

$$\log P(y_{ij}|\mathbf{x}_i) \quad = \quad y_{ij} \log \pi(\theta_{ij}) + (N_{ij} - y_{ij}) \log\{1 - \pi(\theta_{ij})\} + \log \binom{N_{ij}}{y_{ij}},$$

where $\theta_{ij} = \mu_j + \mathbf{w}_j^T \mathbf{x}_i$. Similarly in (3.6), using $\pi(-\theta_{ij}) = 1 - \pi(\theta_{ij})$ and the variational

lower bound

$$\begin{aligned}\log \pi(\theta_{ij}) &\geq \log \pi(\xi_{ij}) + \frac{\theta_{ij}-\xi_{ij}}{2} - \lambda(\xi_{ij})(\theta_{ij}^2 - \xi_{ij}^2),\\ \log \pi(-\theta_{ij}) &\geq \log \pi(-\xi_{ij}) + \frac{\xi_{ij}-\theta_{ij}}{2} - \lambda(-\xi_{ij})(\theta_{ij}^2 - \xi_{ij}^2),\end{aligned}$$

we get the variational lower bound to $P(y_{ij}|\mathbf{x}_i)$ as

$$\begin{aligned}\tilde{P}(y_{ij}|\mathbf{x}_i,\xi_{ij}) &= \binom{N_{ij}}{y_{ij}}\pi(\xi_{ij})^{y_{ij}}\pi(-\xi_{ij})^{N_{ij}-y_{ij}}\\ &\quad\times\exp\Big[(2y_{ij}-N_{ij})(\theta_{ij}-\xi_{ij})/2 - N_{ij}\lambda(\xi_{ij})(\theta_{ij}^2-\xi_{ij}^2)\Big].\end{aligned}$$

Here we used $\lambda(-x) = \lambda(x)$. This will reduce to binomial distribution $B(N_{ij}, \pi(\theta_{ij}))$ when $\xi_{ij} = \theta_{ij}$. It is interesting to note that (3.6) becomes a special case of the lower bound for binomial likelihood with $N_{ij} = 1$, ignoring the constant term. Now using the above and Bayes' theorem, the conditional distribution of $\mathbf{x}_i|\mathbf{y}_i$ becomes a Gaussian distribution with mean $\mathbf{m}_i$ and covariance $\mathbf{C}_i$ as

$$\begin{aligned}\mathbf{C}_i &= \left[\mathbf{I}_k + 2\sum_{j=1}^{d} N_{ij}\lambda(\xi_{ij})\mathbf{w}_j\mathbf{w}_j^T\right]^{-1}\\ \mathbf{m}_i &= \mathbf{C}_i\left[\sum_{j=1}^{d}\left\{y_{ij} - \frac{N_{ij}}{2} - 2N_{ij}\lambda(\xi_{ij})\mu_j\right\}\mathbf{w}_j\right].\end{aligned}$$

Therefore the conditional expectations $\langle\mathbf{x}_i\rangle$ and $\langle\mathbf{x}_i\mathbf{x}_i^T\rangle$ can be computed using (3.8) in the same manner. With these expression, the conditional expectation of the penalized complete log likelihood becomes

$$\begin{aligned}\tilde{Q}_p(\boldsymbol{\Theta}|\boldsymbol{\Theta}^0) &= \sum_{i=1}^{n}\Bigg[\sum_{j=1}^{d}\Bigg\{y_{ij}\log\pi(\xi_{ij}) + (N_{ij}-y_{ij})\log\pi(-\xi_{ij})\\ &\quad+\frac{(2y_{ij}-N_{ij})(\langle\mathbf{x}_i\rangle^T\mathbf{w}_j + \mu_j - \xi_{ij})}{2}\\ &\quad-N_{ij}\lambda(\xi_{ij})(\mathbf{w}_j^T\langle\mathbf{x}_i\mathbf{x}_i^T\rangle\mathbf{w}_j + 2\mu_j\langle\mathbf{x}_i\rangle^T\mathbf{w}_j + \mu_j^2 - \xi_{ij}^2)\Bigg\}\\ &\quad-\frac{k}{2}\log 2\pi - \frac{1}{2}\langle\mathbf{x}_i^T\mathbf{x}_i\rangle\Bigg] - n\sum_{j=1}^{d}\Big\{\mathbf{w}_j^T\boldsymbol{\Omega}_j\mathbf{w}_j + \boldsymbol{\zeta}_j^T\boldsymbol{\Omega}_i\boldsymbol{\zeta}_j\Big\}.\end{aligned}$$

Taking the derivative of $\tilde{Q}_p$ with respect to ξ_{ij} and setting to zero gives

$$\hat{\xi}_{ij} \;=\; \sqrt{\mathbf{w}_j^T\langle \mathbf{x}_i\mathbf{x}_i^T\rangle\mathbf{w}_j + 2\mu_j\langle\mathbf{x}_i\rangle^T\mathbf{w}_j + \mu_j^2}$$

which is the same as in binary case. And the update formulae for location and principal component parameters are given as

$$\begin{aligned}
\hat{\mu}_j &= \sum_{i=1}^n \left\{\frac{2y_{ij}-N_{ij}}{4} - N_{ij}\lambda(\xi_{ij})\langle\mathbf{x}_i\rangle^T\mathbf{w}_j\right\}\Big/\sum_{i=1}^n N_{ij}\lambda(\xi_{ij}),\\
\hat{\mathbf{w}}_j &= \left[\sum_{i=1}^n N_{ij}\lambda(\xi_{ij})\langle\mathbf{x}_i\mathbf{x}_i^T\rangle + n\boldsymbol{\Omega}_j\right]^{-1}\cdot\sum_{i=1}^n\left\{\frac{2y_{ij}-N_{ij}}{4} - N_{ij}\mu_j\lambda(\xi_{ij})\right\}\langle\mathbf{x}_i\rangle.
\end{aligned}$$

When y_{ij} is unobserved, we can address a similar missing treatment using the same approximation to the inverse logit by the probit function. With adopting the same notations as in Section 3.4.2, the conditional expectation $y_{ij}|\mathbf{y}_i^*,\boldsymbol{\Theta}$ is given by

$$\begin{aligned}
\langle y_{ij}\rangle &= E[y_{ij}|\mathbf{y}_i^*,\boldsymbol{\Theta}] \;=\; N_{ij}E[\pi(\theta_{ij})|\mathbf{y}_i^*,\boldsymbol{\Theta}] \;=\; N_{ij}E\left[\frac{\exp(\theta_{ij})}{1+\exp(\theta_{ij})}\middle|\mathbf{y}_i^*,\boldsymbol{\Theta}\right]\\
&\approx N_{ij}\Phi\left(\frac{\mu_j+\mathbf{w}_j^T\mathbf{m}_i^*}{\sqrt{\mathbf{w}_j^T\mathbf{C}_i^*\mathbf{w}_j+8/\pi}}\right),
\end{aligned}$$

where $\mathbf{C}_i^* \;=\; \big[\mathbf{I}_k + 2\sum_{j\in\mathcal{O}_i} N_{ij}\lambda(\xi_{ij})\mathbf{w}_j\mathbf{w}_j^T\big]^{-1}$ and $\mathbf{m}_i^* \;=\; \mathbf{C}_i^*\big[\sum_{j\in\mathcal{O}_i}\big\{y_{ij} - N_{ij}/2 - 2N_{ij}\lambda(\xi_{ij})\mu_j\big\}\mathbf{w}_j\big]$.

3.7.2 Normal variables

The standard principal components analysis for continuous type variables or normal variables is modeled by the Gaussian distribution by Tipping and Bishop (1999) in the name of the probabilistic principal components analysis. When y_{ij} are normally distributed conditionally on $\mathbf{x}_i$, e.g., $\mathbf{y}_i|\mathbf{x}_i \sim N(\boldsymbol{\mu}+\mathbf{W}\mathbf{x}_i, \sigma^2\mathbf{I}_d)$, the conditional distribution of y_{ij} given $\mathbf{x}_i$ is quadratic in exponential, so that variational approximation is not needed. Using Bayes' rule, the conditional distribution of $\mathbf{x}_i|\mathbf{y}_i$ becomes Gaussian with mean $\mathbf{m}_i$ and covariance

$\mathbf{C}_i$ as

$$\begin{aligned}\mathbf{C}_i &= \left[\mathbf{I}_k + \frac{1}{\sigma^2}\sum_{j=1}^{d}\mathbf{w}_j\mathbf{w}_j^T\right]^{-1} \\ \mathbf{m}_i &= \mathbf{C}_i\sum_{j=1}^{d}(y_{ij}-\mu_j)\mathbf{w}_j/\sigma^2.\end{aligned}$$

These are used, again, for computation $\langle\mathbf{x}_i\rangle$ and $\langle\mathbf{x}_i\mathbf{x}_i^T\rangle$ as (3.8). In the E-step, the conditional expectation of the penalized complete log likelihood follows

$$\begin{aligned}Q_p(\boldsymbol{\Theta}|\boldsymbol{\Theta}^0) &= -\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}\sum_{j=1}^{d}\Big\{\mathbf{w}_j^T\langle\mathbf{x}_i\mathbf{x}_i^T\rangle\mathbf{w}_j - 2(y_{ij}-\mu_j)\langle\mathbf{x}_i\rangle^T\mathbf{w}_j + (y_{ij}-\mu_j)^2\Big\}\right] \\ &\quad -\frac{1}{2}\sum_{i=1}^{n}\langle\mathbf{x}_i^T\mathbf{x}_i\rangle - \frac{n(d+k)}{2}\log 2\pi - \frac{nd}{2}\log\sigma^2 \\ &\quad -n\sum_{j=1}^{d}\Big\{\mathbf{w}_j^T\boldsymbol{\Omega}_j\mathbf{w}_j + \boldsymbol{\zeta}_j^T\boldsymbol{\Omega}_j\boldsymbol{\zeta}_j\Big\}.\end{aligned}$$

Contrast to binary case, there are no extra variational parameters, but we have another parameter σ^2 instead. The update formulae for parameters are given as

$$\begin{aligned}\hat{\sigma}^2 &= \frac{1}{nd}\sum_{i=1}^{n}(\mathbf{y}_i-\boldsymbol{\mu}-\mathbf{W}\langle\mathbf{x}_i\rangle)^T(\mathbf{y}_i-\boldsymbol{\mu}-\mathbf{W}\langle\mathbf{x}_i\rangle), \\ \hat{\mu}_j &= \sum_{i=1}^{n}(y_{ij}-\langle\mathbf{x}_i\rangle^T\mathbf{w}_j)/n, \\ \hat{\mathbf{w}}_j &= \left[\frac{1}{2\sigma^2}\sum_{i=1}^{n}\langle\mathbf{x}_i\mathbf{x}_i^T\rangle + n\boldsymbol{\Omega}_j\right]^{-1}\cdot\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_{ij}-\mu_j)\langle\mathbf{x}_i\rangle.\end{aligned}$$

This derivation is exactly the same as the probabilistic model for principal components analysis in Tipping and Bishop (1999). For the missing variable y_{ij}, it follows that

$$\begin{aligned}\langle y_{ij}\rangle &= E[y_{ij}|\mathbf{y}_i^*,\boldsymbol{\Theta}] = E\big[E[y_{ij}|\mathbf{x}_i,\boldsymbol{\Theta}]\big|\mathbf{y}_i^*,\boldsymbol{\Theta}\big] \\ &= E[\mu_j+\mathbf{w}_j^T\mathbf{x}_i|\mathbf{y}_i^*,\boldsymbol{\Theta}] = \mu_j+\mathbf{w}_j^T\mathbf{m}_i^*,\end{aligned}$$

and similarly,

$$\langle y_{ij}^2\rangle = \mu_j^2 + 2\mathbf{w}_j^T\mathbf{m}_i^* + \mathrm{Tr}(\mathbf{C}_i^* + \mathbf{m}_i^*\mathbf{m}_i^{*T})$$

where $\mathbf{C}_i^* = \big[\mathbf{I}_k + \sum_{j\in\mathcal{O}_i}\mathbf{w}_j\mathbf{w}_j^T/\sigma^2\big]^{-1}$ and $\mathbf{m}_i^* = \mathbf{C}_i\sum_{j\in\mathcal{O}_i}(y_{ij}-\mu_j)\mathbf{w}_j/\sigma^2$.

3.7.3 Composite case

Now we consider $\mathbf{y}$ consisting of binary, binomial and normal variables simultaneously in the same dataset. For a simple representation we define some notations in order to combine three different types of variables. Let

$$\phi_j = \begin{cases} 1 & \text{Binary or Binomial} \\ \frac{1}{2\sigma^2} & \text{Normal,} \end{cases} \tag{3.12}$$

$$t_{ij} = \begin{cases} (2y_{ij}-1)/4 & \text{Binary} \\ (2y_{ij}-N_{ij})/4 & \text{Binomial} \\ y_{ij} & \text{Normal} \end{cases} \tag{3.13}$$

and

$$\lambda_{ij} = \begin{cases} \lambda(\xi_{ij}) & \text{Binary} \\ N_{ij}\lambda(\xi_{ij}) & \text{Binomial} \\ 1 & \text{Normal.} \end{cases} \tag{3.14}$$

Then, the update formulae for μ_j and $\mathbf{w}_j$ turns into the unified forms:

$$\mu_j = \sum_{i=1}^{n}\big(t_{ij} - \lambda_{ij}\langle \mathbf{x}_i\rangle^T \mathbf{w}_j\big) \Big/ \sum_{i=1}^{n}\lambda_{ij} \tag{3.15}$$

$$\mathbf{w}_j = \big(\mathbf{A}_j + n\boldsymbol{\Omega}_j\big)^{-1}\mathbf{z}_j \tag{3.16}$$

where

$$\mathbf{A}_j = \phi_j \sum_{i=1}^{n}\lambda_{ij}\langle \mathbf{x}_i\mathbf{x}_i^T\rangle,$$

$$\mathbf{z}_j = \phi_j \sum_{i=1}^{n}\big(t_{ij} - \lambda_{ij}\mu_j\big)\langle \mathbf{x}_i\rangle.$$

And mean and covariance of the conditional distribution of $\mathbf{x}_i|\mathbf{y}_i$ can be written as

$$\begin{aligned}\mathbf{C}_i &= \left[\mathbf{I}_k + 2\sum_{j=1}^{d}\phi_j\lambda_{ij}\mathbf{w}_j\mathbf{w}_j^T\right]^{-1} \\ \mathbf{m}_i &= \mathbf{C}_i\left[2\sum_{j=1}^{d}\Big\{t_{ij} - \lambda_{ij}\mu_j\Big\}\phi_j\mathbf{w}_j\right]\end{aligned}$$

and the first two moments for the E-step of the EM algorithm can be easily obtained using them.

Now consider the case in that some elements of composite vector $\mathbf{y}_i$ are missing. Suppose y_{ij} is unobserved and it can be any type of binary, binomial or normal. Denote the missing index set of js in the ith individual by $\mathcal{O}_i$ as in the previous arguments. Then the conditional distribution of $\mathbf{x}_i|\mathbf{y}_i^*, \boldsymbol{\Theta}$ becomes a Gaussian distribution with mean $\mathbf{m}_i^*$ and covariance $\mathbf{C}_i^*$ given as

$$\begin{aligned}\mathbf{C}_i^* &= \left[\mathbf{I}_k + 2\sum_{j\in\mathcal{O}_i}\phi_j\lambda_{ij}\mathbf{w}_j\mathbf{w}_j^T\right]^{-1} \\ \mathbf{m}_i^* &= \mathbf{C}_i\left[2\sum_{j\in\mathcal{O}_i}\Big\{t_{ij} - \lambda_{ij}\mu_j\Big\}\phi_j\mathbf{w}_j\right].\end{aligned}$$

And using them, the missing value t_{ij} is imputed by the conditional expectation as

$$\langle t_{ij}\rangle = \begin{cases}(2\langle y_{ij}\rangle - 1)/4 & \text{Binary} \\ (2\langle y_{ij}\rangle - N_{ij})/4 & \text{Binomial} \\ \langle y_{ij}\rangle & \text{Normal}\end{cases} \tag{3.17}$$

where the corresponding $\langle y_{ij}\rangle$ for each variable type is given in the previous sections.

CHAPTER IV

SUMMARY

In this dissertation, we develop principal components analysis for binary data and study its performance with various scenarios, including simulation datasets and real data examples. Especially we pay an attention on the automatic variable selection in the high-dimensional situation. To this end, we focus on the minimum error formulation of principal components analysis for the normal variables and observe that minimizing the sum of errors between the data points and their projections is equivalent to maximizing the Gaussian log likelihood. This observation is generalized to the binary dataset with Bernoulli distribution. Bernoulli likelihood is maximized in the low dimensional subspace of canonical parameter space. In order to capture the features among high-dimensional variables, we introduce L_1 penalty on principal components so that only small portion of nonzero variable loadings appear in resulting principal components. This L_1 regularization turns out to improve in picking out the meaningful variabilities among high dimensional variables throughout simulations studies and real data applications, including binary image data, web advertisement data and single nucleotide polymorphsm data.

In the estimation perspective, we approach maximization problem of the penalized Bernoulli likelihood in two directions. In Chapter II, principal component scores are regarded as fixed parameters as in standard PCA problem. The maximum penalized likelihood estimator is obtained by maximizing its surrogate function iteratively. Specifically, this surrogate function is a quadratic lower bound which is easy to be optimized and gives stable estimation procedure removing possible computational instabilities such as overshooting problem. This approach is known as Majorization or MM algorithm. In Chapter II, we demonstrate L_1 penalty can also be cast into quadratic lower bound maximization

as well as log likelihood. And we prove the missing value treatment we propose here can be viewed as another layer of majorization step. As another approach, Chapter III deals with principal component scores as latent variables. One of nice features of this formulation is that the number of parameters to be estimated becomes considerably smaller than the approach in Chaper II. Latent variable model often uses EM algorithm for parameter estimation due to its latent variable nature. Problem is that E step is not in the closed-form so numerical approximations is indispensable, for example, Gauss-Hermite quadrature, Laplace approximation or Monte-Carlo EM, all of which are computationally infeasible in high-dimensional binary data. Instead of such approximations for marginal log likelihood, we propose to use variational method which gives quadratic lower bound for the marginal log likelihood and stable algorithm for parameter estimation. Since the negative L_1 penalty also has quadratic lower bound, two formulations are easily combined in the algorithm.

REFERENCES

Asuncion, A. and Newman, D. J. (2007). UCI machine learning repository.. http://www.ics.uci.edu/∼mlearn/MLRepository.html, University of California, Irvine, School of Information and Computer Sciences.

Bartholomew, D. J. (1984). Scaling binary data using a factor model. *Journal of the Royal Statistical Society, Series B* **46**, 120–123.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Singapore: Springer.

Bishop, C. M. and Tipping, M. E. (1998). A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20(3)**, 281–293.

Brooks, A. J. (1999). Review: The essence of snps. *Gene* **234**, 177–186.

Collins, M., Dasgupta, S., and Schapire, R. E. (2001). A generalization of principal component analysis to the exponential family. *Advanced in Neural Information Processing System*, In T. G. Dietterich, S. Becker, and Z. Ghahramani (eds), 617–632. British Columbia, Cananda: MIT press.

de Leeuw, J. (2006). Principal component analysis of binary data by iterated singular value decomposition. *Computational Statistics and Data Analysis* **50**, 21–39.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm (with discussion). *Journal of the Royal Statistical Society, Series B* **39**, 1–38.

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

Ewens, W. J. and Spielman, R. S. (1995). The transmission/disequilibrium test: History, subdivision, and admixture. *The American Journal of Human Genetics* **57**, 455–464.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the Americal Statistical Association* **96**, 1348–1360.

Golub, G. and van Loan, C. (1996). *Matrix Computations, 3rd ed.* Baltimore: The Johns Hopkins University Press.

Hao, K., Li, C., Rosenow, C., and Wong, W. H. (2004). Detect and adjust for population stratification in population-based association study using genomic control markers: An application of affymetrix genechip human mapping 10k array. *European Journal of Human Genetics* **12**, 1001–1006.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models*. New York: Chapman & Hall/CRC.

Hastie, T. J., Tibshirani, R. J., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* **24**, 417–441.

Huber, P., Ronchetti, E., and Victoria-Feser, M. (2004). Estimation of generalized linear latent variable models. *Journal of the Royal Statistical Society, Series B* **66**, 893–908.

Hunter, D. R. and Lange, K. (2004). A tutorial on mm algorithms. *The American Statistician* **58**, 30–37.

Hunter, D. R. and Li, R. (2005). Variable selection using mm algorithms. *Annals of Statistics* **33**, 1617–1642.

Jaakkola, T. S. and Jordan, M. I. (1997). Bayesian logistic regression: A variational approach. *The 1997 Conference on Artificial Intelligence and Statistics*, In D. Madigan and P. Smyth (eds), 65–71. Ft. Lauderdale, FL.

Jaakkola, T. S. and Jordan, M. I. (2000). Beyesian parameter estimation via variational methods. *Statistics and Computing* **10**, 25–37.

Jolliffe, I. T. (2004). *Principal Component Analysis, 2nd ed*. New York: Springer.

Jolliffe, I. T., Trendafilov, M., and Uddine, M. (2003). A modified principal component technique based on lasso. *Journal of Comuputational and Graphical Statistics* **12**, 531–547.

Jordan, M. I. (1999). *Learning in Graphical Models*. Cambridge: MIT press.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introducion to variational methods for graphical models. *Learning in Graphical Models*, In M. I. Jordan (ed), 105–162. Cambridge: MIT press.

Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika* **23**, 187–200.

Kwok, P. Y., Deng, Q., Zakeri, H., Taylor, S. L., and Nickerson, D. A. (1996). Increasing the information content of sts-based genome maps: Identifying polymorphisms in mapped stss. *Genomics* **31**, 123–126.

Lange, K., Hunter, D. R., and Yang, I. (2000). Optimization tranfer using surrogate objective functions (with discussion). *Journal of Comuputational and Graphical Statistics* **9**, 1–207.

Liang, Y. and Kelemen, A. (2008). Statistical advances and challenges for analyzing correlated high dimensional snp data in genomic study for complex diseases. *Statistics Surveys* **2**, 559–572.

Mackay, D. J. C. (1992). The evidence framework applied to classification networks. *Neural Computation* **4(5)**, 720–736.

McCulloch, C. E. and Searle, S. R. (2001). *Generalized, Lienar, and Mixed Models.* New York: Wiley & Sons.

Moustaki, I. and Knott, M. (2000). Generalized latent trait models. *Psychometrika* **65**, 391–411.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, Sixth Series* **2**, 559–572.

Rockafella, R. (1972). *Convex Analysis*. New Jersey: Princeton University Press.

Samel, M. D., Ryan, L. M., and Legler, J. M. (1997). Latent variable models for mixed discrite and continuous outcomes. *Journal of the Royal Statistical Society, Series B* **59**, 667–678.

Schein, A. I., Saul, L. K., and Ungar, L. H. (2003). A generalized linear model for principal component analysis of binary data. In *Proceeding of the Ninth International Workshop on Artificial Intelligence and Statistics*,, 14–21. Key West, FL.

Serre, D., Montpetit, A., Paré, G., Engert, J. G., Yusuf, S., Keavney, B., and Hudson, K. J. (2008). Correction of population stratification in large multi-ethnic association studies. *PLoS ONE* **2(1)**, e1382.

Shen, H. and Huang, J. Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis* **99**, 1015–1034.

Skrondal, A. and Rabe-Hesketh, S. (2004). *Generalized Latent Variable Modeling: Multilevel, Longitudinal and Structural Equation Models*. Boca Raton, FL: Chapman & Hall/CRC.

The International HapMap Consortium (2005). A haplotype map of the human genome. *Nature* **437**, 1299–1320.

Tibshirani, R. J. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* **58**, 267–288.

Tipping, M. E. (1999). Probabilistic visualization of high-dimensional binary data. *Advances in Neural Information Processing Systems*, In M. S. Kearn, S. A. Solla, and D. A. Cohn (eds), 592–598. Denver: MIT Press.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B* **21(3)**, 611–622.

Zou, H., Hastie, T. J., and Tibshirani, R. J. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics* **15**, 265–286.

Zou, H., Hastie, T. J., and Tibshirani, R. J. (2007). On the “degrees of fredom” of the lasso. *Annals of Statistics* **35**, 2173–2192.

VITA

Seokho Lee received a B.S. degree in Computer Science and Statistics from Seoul National University, Korea in 1998. He received a M.S. degree in Statistics from Seoul National University, Korea. He was admitted to the Ph.D. program in the Department of Statistics at Texas A&M University in January 2005, and he received his Ph.D. degree in May 2009. His address is Department of Statistics, Texas A&M University, TAMU 3143, College Station, TX 77843-3143, USA.