Generalized Principal Component Analysis: Projection of Saturated Model Parameters

Andrew J. Landgraf AND Yoonkyung Lee

Abstract

Principal component analysis (PCA) is very useful for a wide variety of data analysis tasks, but its implicit connection to the Gaussian distribution can be undesirable for discrete data such as binary and multi-category responses or counts. We generalize PCA to handle various types of data using the generalized linear model framework. In contrast to the existing approach of matrix factorizations for exponential family data, our generalized PCA provides low-rank estimates of the natural parameters by projecting the saturated model parameters. This difference in formulation leads to the favorable properties that the number of parameters does not grow with the sample size and simple matrix multiplication suffices for computation of the principal component scores on new data. A practical algorithm which can incorporate missing data and case weights is developed for finding the projection matrix. Examples on simulated and real count data show the improvement of generalized PCA over standard PCA for matrix completion and collaborative filtering.

Keywords: Binary data; Count data; Dimensionality reduction; Exponential family; Low rank model

1 Introduction

Abundance of high dimensional data in many fields makes understanding and capturing the regular structures underlying the data crucial for subsequent modeling and prediction. Low dimensional projections of data are often primary tools for coping with high dimensionality along with other techniques for sparsity or structural simplicity. This paper concerns an extension of standard principal component analysis (PCA), a widely used tool for low dimensional approximation of data. The proposed extension aims to expand the scope of PCA to various types of discrete data from binary and multi-category responses to counts. For discrete data, the implicit connection of PCA to the Gaussian distribution may be undesirable.

Several approaches to dimensionality reduction for non-Gaussian data have been proposed when all of the variables are of one particular type. For example, probabilistic PCA (Tipping and Bishop, 1999) has been extended to binary data (Tipping, 1998) as well as data from

a multinomial distribution (Buntine, 2002), which is related to latent Dirichlet allocation (Blei et al., 2003). Logistic PCA (Schein et al., 2003) fits a low-rank estimate of the natural parameter matrix for Bernoulli data. Correspondence analysis (Greenacre, 1984) reduces the dimension of categorical data by finding a low-rank representation of the contingency table. Non-negative matrix factorization (Lee and Seung, 1999) finds a low-rank estimate of the mean parameters assuming that the data come from a Poisson distribution and under the constraints that the factors are non-negative.

Other techniques have been developed for the dimensionality reduction of many possible types of non-Gaussian data. Collins et al. (2001) introduced a general framework for low-rank matrix factorizations of the natural parameters of exponential family data by maximizing the constrained likelihood of exponential family distributions. This matrix factorization formulation has further been generalized (Gordon, 2002; Singh and Gordon, 2008; Udell et al., 2014), incorporating more loss functions and data types, as well as penalties on the row and column factors. Many of the formulations above can be shown to be special cases of these generalizations.

We propose a new formulation of generalized PCA which extends Pearson (1901)'s motivation for the optimal data representation with minimum mean squared error to data from members of the exponential family. A special case of the proposed generalization for binary data (called logistic PCA) is presented in Landgraf and Lee (2015). In contrast to the existing approach of matrix factorizations for exponential family data, our generalized PCA provides low-rank estimates of the natural parameters by projecting the saturated model parameters. This projection approach primarily concerns the structure underlying the joint distribution of the variables for dimensionality reduction as in standard PCA, whereas the matrix factorization approach treats both variables and cases interchangeably. Due to the structural difference, our generalized PCA has several advantages over the matrix factorization methods. Applying principal components to new data only requires matrix multiplication, and the number of parameters does not increase with the number of cases. Furthermore, the principal components scores are more interpretable, as they are a linear combination of the natural parameters from the saturated model.

We pay particular attention to dimensionality reduction of count data in this paper, either with a Poisson or multinomial assumption. Count datasets occur in many domains of interest. When analyzing text documents with a bag-of-words representation, the data consist of the counts of each word in each document. In recommendation systems, counts may indicate the number of times a user has interacted with an item. The counts could be clicks on a website, number of times listening to a song, or number of times shopping at a merchant. For the recommendation task of the Netflix competition, low-rank models are one of the most successful methods used (Feuerverger et al., 2012). If the goal is to minimize the squared error of the predicted rating, as was the case in the Netflix competition, standard PCA makes sense. However, if the goal is to predict which movies are most likely to be watched, other distributional assumptions are more appropriate.

In addition, we broaden the scope of generalized PCA by allowing for observation weights, missing data, and variable normalization. Weights can be useful for indicating a differen-

tial weight naturally associated with each observation. For example, if an observation is a proportion, it should get higher weight for being associated with more trials. With missing data, the case weight can be set to either zero or one, depending on whether it is missing or observed. Missing data are common in collaborative filtering problems, where the goal is often to fill in the missing values of a matrix as accurately as possible using the observed values. The normalization allows variables of different types and magnitudes to be on the same scale, similar to how PCA can be applied to the correlation matrix instead of the covariance matrix.

For implementation, a majorization-minimization (MM) algorithm is derived to determine the principal component loadings. By using the convex hull of projection matrices, an alternative algorithm for solutions to a convex relaxation of the original optimization problem is derived as well. Both of these algorithms work in the presence of weights and missing data. Numerically, we examine the relative performance of the generalized PCA in matrix completion and recommendation tasks with simulated count datasets, varying the sparsity and the magnitude of the positive counts. Further, using a count dataset of users' song listening history, we show the improvement in visualizing the loadings and recommending new songs to users with our generalization of PCA.

The rest of the paper is organized as follows. The generalized PCA formulation is introduced in Section 2, along with first-order optimality conditions for solutions and extensions to handle differential case weights, missing data, and variable normalization. Section 3 presents the two computational algorithms for generalized PCA. The simulated and real data examples are provided in Sections 4 and 5, respectively, which show the usefulness of the proposed formulation for visualization, recommendation, and matrix completion. Finally, Section 6 offers conclusions and future research directions.

2 Generalized PCA

2.1 Preliminaries

Generalized PCA presented in this paper is based on generalized linear models and exponential family distributions. The probability density or mass function of a random variable X with an exponential family distribution can be written as

$$f(x|\theta,\phi) = \exp\left(\frac{x\theta - b(\theta)}{a(\phi)} + c(x,\phi)\right), \tag{2.1}$$

where θ is the canonical natural parameter and ϕ is the scale parameter, which is assumed to be known. The mean and variance of the distribution are easily obtained from

$$E_\theta(X) = b'(\theta) = \frac{\partial}{\partial\theta} b(\theta) \text{ and}$$
$$\text{var}_\theta(X) = a(\phi)b''(\theta) = a(\phi)\frac{\partial^2}{\partial\theta^2} b(\theta).$$

For simplicity of exposition, we will further assume that $a(\phi) = 1$, which is the case for the binomial and Poisson distributions and for the Gaussian distribution with unit variance. We will revisit the scale function $a(\cdot)$ when discussing normalization in Section 2.5.3.

The canonical link function $g(\cdot)$ is defined such that $g(b'(\theta)) = \theta$. The best possible fit to the data is referred to as the saturated model, where the mean $b'(\tilde{\theta})$ is set equal to the data x, implying that the saturated model parameter $\tilde{\theta}$ is $g(x)$. For instance, for the Gaussian distribution, $b(\theta) = \theta^2/2$ and $\tilde{\theta} = x$, for Bernoulli, $b(\theta) = \log(1 + \exp(\theta))$ and $\tilde{\theta} = \text{logit } x$, and for Poisson, $b(\theta) = \exp(\theta)$ and $\tilde{\theta} = \log x$. This data-driven saturated model will play an important role in defining a low-rank data approximation for generalized PCA. Note also that, for some distributions and values of x, $g(x)$ may not be finite. For example, if the distribution is Bernoulli, $g(1) = \infty$ and $g(0) = -\infty$, and if the distribution is Poisson, $g(0) = -\infty$.

Finally, the deviance for θ based on observation x is defined as

$$D(x; \theta) = -2 \left(\log f(x|\theta, \phi) - \log f(x|\tilde{\theta}, \phi) \right).$$

Suppose that there are n independent observations on d variables from the same distribution family. Then, the deviance of an $n \times d$ matrix of natural parameters, $\mathbf{\Theta} = [\theta_{ij}]$, with a data matrix of the same size, $\mathbf{X} = [x_{ij}]$, is given by

$$D(\mathbf{X}; \mathbf{\Theta}) = \sum_{i=1}^{n} \sum_{j=1}^{d} D(x_{ij}; \theta_{ij}).$$

2.2 Basic Formulation

Our formulation of generalized PCA extends an alternative characterization of PCA that Pearson (1901) originally considered for dimensionality reduction. Pearson's formulation looks for the optimal projection of the original d-dimensional points $\mathbf{x}_i$ in a k-dimensional subspace ($k < d$) under squared error loss. That is, finding a center $\boldsymbol{\mu} \in \mathbb{R}^d$ and a rank-k matrix $\mathbf{U} \in \mathbb{R}^{d \times k}$ with $\mathbf{U}^T\mathbf{U} = \mathbf{I}_k$ that minimize

$$\sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{\mu} - \mathbf{U}\mathbf{U}^T(\mathbf{x}_i - \boldsymbol{\mu})\|^2.$$

It is shown that $\hat{\boldsymbol{\mu}} = \bar{\mathbf{x}}$ and $\hat{\mathbf{U}}$ containing the first k eigenvectors of the sample covariance matrix give the optimal representation, coinciding with the data projection with k principal components. This formulation can be interpreted as a dimensionality reduction technique for Gaussian data. If x_{ij} comes from a Gaussian distribution with known variance, the natural parameter θ_{ij} is the mean, the deviance is proportional to squared error loss, and the natural parameter from the saturated model is $\tilde{\theta}_{ij} = x_{ij}$. Hence, the approximation of $\mathbf{x}_i$ by $\boldsymbol{\mu} + \mathbf{U}\mathbf{U}^T(\mathbf{x}_i - \boldsymbol{\mu})$ can be viewed equivalently as that of approximating the saturated model parameters $\tilde{\boldsymbol{\theta}}_i$ by $\boldsymbol{\mu} + \mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})$. With these characteristics in mind, we can generalize Pearson's formulation as a method for projecting the saturated model into a lower

dimensional space by minimizing the deviance for natural parameters in a subspace based on the data.

Pearson's original formulation of a low-rank representation of data does not technically require the normality assumption, however, this dual interpretation is very conducive to extensions of standard PCA to data from other exponential family distributions. Taking this approach, Landgraf and Lee (2015) introduced an extension of PCA to binary data as a low-rank projection of the logit parameters of the saturated model. In this paper, we further generalize it to other types of data, including counts and multicategory responses. For this generalization, we allow each variable to be from a different member of the exponential family and specify the appropriate distribution for each variable.

In this framework, we want to estimate the natural parameters θ_{ij} for data x_{ij} in a k-dimensional subspace such that they minimize the deviance

$$D(\mathbf{X};\mathbf{\Theta}) = \sum_{i=1}^{n}\sum_{j=1}^{d} D_j(x_{ij};\theta_{ij}) = 2\sum_{i=1}^{n}\sum_{j=1}^{d}\{-x_{ij}\theta_{ij} + b_j(\theta_{ij})\} + \text{constant}.$$

Since each variable of the dataset may come from a different distribution, the deviance and $b(\cdot)$ functions have been subscripted by column. Letting $\tilde{\theta}_{ij}$ be the natural parameter for the saturated model of the jth variable and $\tilde{\boldsymbol{\theta}}_i$ be the vector for the ith case, the estimated natural parameters in a k-dimensional subspace are defined as

$$\hat{\theta}_{ij} = \mu_j + \left[\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})\right]_j,$$

or $\hat{\mathbf{\Theta}} = \mathbf{1}_n\boldsymbol{\mu}^T + (\tilde{\mathbf{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{U}\mathbf{U}^T$ in matrix notation.

We wish to find $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\mathbf{U} \in \mathbb{R}^{d\times k}$ with $\mathbf{U}^T\mathbf{U} = \mathbf{I}_k$ that minimize

$$\begin{aligned}
&\sum_{i,j}\left\{-x_{ij}\left(\mu_j + \left[\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})\right]_j\right) + b_j\left(\mu_j + \left[\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})\right]_j\right)\right\} \\
&= -\langle \mathbf{X}, \mathbf{1}_n\boldsymbol{\mu}^T + (\tilde{\mathbf{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{U}\mathbf{U}^T\rangle + \sum_{i,j} b_j\left(\mu_j + \left[\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})\right]_j\right),
\end{aligned}$$

where $\langle \mathbf{A}, \mathbf{B}\rangle = tr(\mathbf{A}^T\mathbf{B})$ is the trace inner product. Note that for some distributions and values of x_{ij}, $\tilde{\theta}_{ij}$ may be ∞ or $-\infty$. For instance, $\tilde{\theta}_{ij} = -\infty$ for $x_{ij} = 0$ under the Poisson distribution. Numerically, we will approximate ∞ with a large positive constant m, treating it as a tuning parameter. It can be chosen data-adaptively by cross-validation, as was done in Landgraf and Lee (2015) for the Bernoulli distribution.

2.3 First-Order Optimality Conditions

To examine the relation between the optimization problems that standard PCA and the proposed generalization involve, we derive necessary conditions for the generalized PCA solutions. The orthonormality constraints on $\mathbf{U}$ naturally lead to the application of the

method of Lagrange multipliers. In order to incorporate the orthonormal constraints $\mathbf{U}^T\mathbf{U} = \mathbf{I}_k$, we consider the Lagrangian,

$$L(\mathbf{U}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \sum_{i,j} D_j \left(x_{ij}; \mu_j + \left[\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu}) \right]_j \right) + tr(\boldsymbol{\Lambda}(\mathbf{U}^T\mathbf{U} - \mathbf{I}_k)),$$

where $\boldsymbol{\Lambda}$ is a $k \times k$ symmetric matrix with the Lagrange multipliers. See Wen and Yin (2013) for more general discussions of optimization with orthogonal constraints.

Taking the gradient of the Lagrangian with respect to the $\mathbf{U}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Lambda}$ and setting them equal to $\mathbf{0}$ gives the first-order optimality conditions for generalized PCA solutions:

$$\left[\left(\mathbf{X} - b'(\hat{\boldsymbol{\Theta}}) \right)^T \left(\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n \boldsymbol{\mu}^T \right) + \left(\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n \boldsymbol{\mu}^T \right)^T \left(\mathbf{X} - b'(\hat{\boldsymbol{\Theta}}) \right) \right] \mathbf{U} = \mathbf{U}\boldsymbol{\Lambda} \tag{2.2}$$

$$\left(\mathbf{I}_d - \mathbf{U}\mathbf{U}^T \right) \left(\mathbf{X} - b'(\hat{\boldsymbol{\Theta}}) \right)^T \mathbf{1}_n = \mathbf{0}_d, \text{ and} \tag{2.3}$$

$$\mathbf{U}^T\mathbf{U} = \mathbf{I}_k. \tag{2.4}$$

The details of the calculation can be found in Appendix A.1. $b'(\hat{\boldsymbol{\Theta}})$ is a matrix given by taking $b'(\cdot)$ elementwise. Each element is the expected value of x_{ij} at the optimal parameter $\hat{\theta}_{ij}$. That is, the ijth element equals $b'_j \left(\mu_j + \left[\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu}) \right]_j \right)$. For the Gaussian distribution, $b'(\theta) = \theta$ and $\tilde{\boldsymbol{\Theta}} = \mathbf{X}$, which simplify the condition (2.2) to the eigen equation for a covariance matrix. Thus, (2.2) is analogous to the eigen equation for standard PCA. (2.3) can be interpreted as the condition that the residuals projected onto the null space of $\mathbf{U}$ sum to zero.

2.4 Geometry of the Projection

To illustrate the difference in the low-rank representation of data using standard PCA and generalized PCA, we simulated a two-dimensional count dataset with 100 rows where the mean matrix has a rank of one. The data are plotted in Figure 1. The area of the points is proportional to the number of observations with that pair of counts. For example, there were 12 observations with the counts $(0, 0)$ and most of the larger counts only have one observation. The one-dimensional representations for standard PCA and Poisson PCA are displayed in the figure. The two solid lines indicate the one-dimensional mean space given by standard PCA (red) and Poisson PCA (blue).

The most obvious difference between the representations is that the Poisson PCA representation is non-linear. It is also constrained to be non-negative unlike the standard PCA projections. For instance, the standard PCA projection of $(0, 0)$ is slightly negative for the second variable. The dashed lines in the figure show the correspondence between selected observations and their projections onto the lower dimensional space. Standard PCA gives projections in the count space, while Poisson PCA gives projections in the natural parameter space. When transforming back to the count space, the Poisson PCA fits are no longer projections in the Euclidean sense.

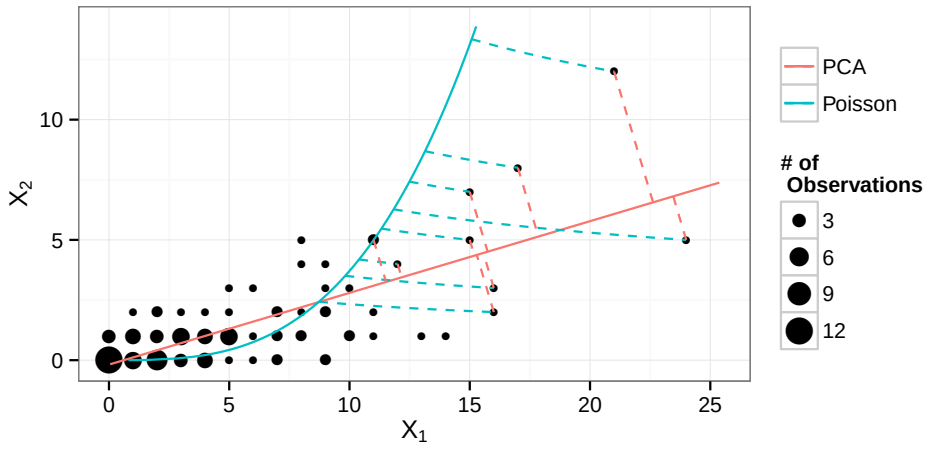

Figure 1: Projection of two-dimensional count data using standard PCA and Poisson PCA

The Poisson PCA estimates seem to fit those counts close to zero more tightly than the larger counts. This is due to the properties of the Poisson distribution, where the mean is proportional to the variance. As shown in Lemma 1 of Appendix A.2, if $(x - \lambda)$ is held constant, where λ is the mean, the deviance monotonically decreases to zero as either λ or x increases to ∞. A large difference of the Poisson PCA fit from an observation in the count space may not correspond to a large deviance when the estimate or observation is relatively large. This is why some of the highlighted projections are more distant from the observation in the count space for Poisson PCA than for standard PCA and also why the Poisson PCA fits the values close to 0 more tightly.

2.5 Further Extensions

Generalized PCA is flexible enough to allow for a variety of extensions, incorporating data weights, missing data and normalization of variables of different types or scales.

2.5.1 Weights

When each observation x_{ij} has a weight w_{ij}, it can easily be incorporated into the deviance as $\sum_{i,j} w_{ij} D_j(x_{ij}; \theta_{ij})$ analogous to the weighted least squares criterion. One example where weights can naturally occur is when a given observation is associated with multiple replications. For example, if an observation is a proportion of "successes" in w number of Bernoulli trials, then w is the natural weight for the observation.

2.5.2 Missing Data

In many practical applications, data may be missing and filling in the missing values is often an important task. For example, in collaborative filtering, the goal is to use observed users'

feedback on items to estimate their unobserved or missing feedback. One of the desirable features of dimensionality reduction techniques is the ability to not only accommodate missing data, but also fill in values that are missing. Techniques that can account for constraints in the domain have an added advantage of only filling in feasible values.

Let $\Omega \subset \{1, \ldots, n\} \times \{1, \ldots, d\}$ be the set of indices for which x_{ij} is observed. Instead of defining the deviance over all entries in the matrix, we could define the deviance over non-missing elements only and solve for the parameters that minimize the weighted deviance,

$$\sum_{(i,j)\in\Omega} w_{ij} D_j(x_{ij}; \theta_{ij}).$$

This is equivalent to setting $w_{ij} = 0$ for all $(i, j) \notin \Omega$. However, the form of the estimated natural parameters for $(i, j) \in \Omega$, $\hat{\theta}_{ij} = \mu_j + [\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})]_j$, requires $\tilde{\boldsymbol{\theta}}_i$, the saturated model parameters for both observed and missing data of the ith case. Therefore, we must extend the definition of saturated model parameters to account for missing data.

We may extend the definition of the natural parameters from the saturated model as

$$\tilde{\theta}_{ij} = \begin{cases} g_j(x_{ij}) & \text{if } (i, j) \in \Omega \\ \mu_j & \text{if } (i, j) \notin \Omega \end{cases}.$$

The saturated model has the best possible fit to the data. When x_{ij} is known, the data can be fit exactly, but when x_{ij} is unknown, our best guess for the observation is the main effect of the jth variable.

One consequence of this definition is that the estimated natural parameters only depend on the saturated model parameters of the non-missing data. The estimated natural parameter is

$$\hat{\theta}_{ij} = \mu_j + [\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})]_j = \mu_j + \sum_{l:(i,l)\in\Omega} [\mathbf{U}\mathbf{U}^T]_{jl}(\tilde{\theta}_{il} - \mu_l).$$

Further, μ_j being in lieu of $\tilde{\theta}_{ij}$ does not complicate its estimation because it cancels out in the expression of $(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})$ when x_{ij} is missing.

2.5.3 Normalization

If data are on different scales in standard PCA, it is common practice to normalize the variables by dividing them by their standard deviations first and apply PCA to the standardized variables, which is equivalent to performing PCA on the correlation matrix. Difficulties in normalization can occur when the variables are of different types. How do we compare a binary variable with mean 1/4 to a count variable with mean 10 to a continuous variable with variance 5? Standardization makes sense only for location-scale families. For example, if we standardize a Bernoulli random variable, it is no longer Bernoulli.

We would like to derive a notion of variable normalization for generalized PCA similar to standardization for PCA. For example, as a result of normalizing variables before PCA, all of

the standard basis vectors $\mathbf{e}_j, j = 1, \ldots, d$, as loadings explain the same amount of variance, which is proportional to the reduction in Gaussian deviance with those loadings. We extend this criterion to generalized PCA by weighting the variables such that the reduction in deviance (or equivalently the deviance) is the same with each of the standard basis vectors as loadings.

Letting $\tau_j, j = 1, \ldots, d$, be the normalizing factors we assign to the variables, $\mathbf{W} = [w_{ij}]$, and $D_j(X_j, W_j; \Theta_j) := \sum_{i=1}^n w_{ij} D_j(x_{ij}; \theta_{ij})$, we want the lth "partial" deviance after normalization

$$D_{\mathbf{e}_l} := \min_{\boldsymbol{\mu}} \sum_{j=1}^{d} \frac{1}{\tau_j} D_j(X_j, W_j; \mathbf{1}_n \mu_j + [(\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n \boldsymbol{\mu}^T)\mathbf{e}_l \mathbf{e}_l^T]_j)$$

to be the same for all l. Note that τ_j can also be interpreted as an estimate for the scale parameter function, $a_j(\phi_j)$ in equation (2.1) up to a proportionality constant.

When $\mathbf{U} = \mathbf{e}_l$, the deviance is minimized with respect to the main effects $\boldsymbol{\mu}$ if $\hat{\mu}_j = g_j(\bar{X}_j^w)$ for $j \neq l$, where $\bar{X}_j^w := X_j^T W_j / (\mathbf{1}_n^T W_j)$ is the weighted mean of the jth variable and $g_j(\cdot)$ is the canonical link function for the jth variable. Therefore,

$$\hat{\theta}_{ij} = \begin{cases} g_j(\bar{X}_j^w) & \text{if } j \neq l \\ g_j(x_{ij}) & \text{if } j = l \end{cases}.$$

This implies that

$$D_{\mathbf{e}_l} = \sum_{j=1, j\neq l}^{d} \frac{1}{\tau_j} D_j(X_j, W_j; \mathbf{1}_n g_j(\bar{X}_j^w)) + \frac{1}{\tau_l} D_l(X_l, W_l; g_l(X_l)).$$

$D_l(X_l, W_l; g_l(X_l)) = 0$ by definition because $g_l(X_l)$ gives the fit of the saturated model.

The goal is to find $\tau_j > 0, j = 1, \ldots, d$, such that $D_{\mathbf{e}_l}$ is the same for all l, i.e.

$$D_{\mathbf{e}_1} = D_{\mathbf{e}_2} = \cdots = D_{\mathbf{e}_d},$$

which implies that

$$\frac{1}{\tau_1} D_1(X_1, W_1; \mathbf{1}_n g_1(\bar{X}_1^w)) = \cdots = \frac{1}{\tau_d} D_d(X_d, W_d; \mathbf{1}_n g_d(\bar{X}_d^w)).$$

These are equal if $\tau_j \propto D_j(X_j, W_j; \mathbf{1}_n g_j(\bar{X}_j^w))$. We set

$$\tau_j = \frac{1}{n} D_j(X_j, W_j; \mathbf{1}_n g_j(\bar{X}_j^w))$$

to make the normalizing factors more interpretable. Note that τ_j will be greater than zero as long as the variance of the jth variable is greater than zero and that $D_j(X_j, W_j; \mathbf{1}_n g_j(\bar{X}_j^w))$ is the deviance of the null (intercept-only) model for the jth variable and τ_j is the average null deviance.

The expression of the normalization factor τ is examined from some exponential family distributions with no missing data and constant weights.

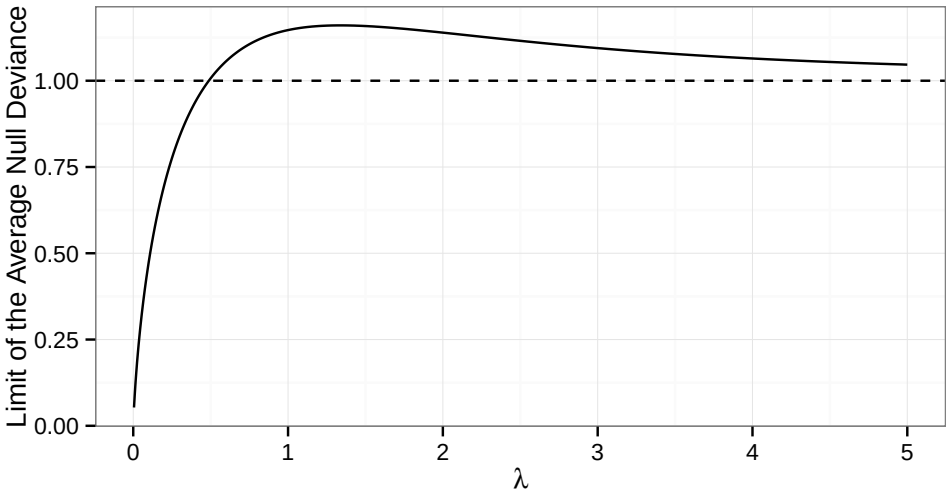

Figure 2: Numerical estimates of the limit of the average null deviance as a function of the mean parameter for Poisson(λ)

Gaussian Distribution For Gaussian variables, $\tau_j = \text{var}(X_j)$. This normalization is very similar, but not exactly equivalent to performing PCA on the correlation matrix. Let $\boldsymbol{\tau} := \text{diag}(\tau_1, \ldots, \tau_d)$ be a d-dimensional diagonal matrix with the normalizing factors on the diagonals. PCA on the correlation matrix minimizes

$$\|(\mathbf{X} - \mathbf{1}_n \boldsymbol{\mu}^T)\boldsymbol{\tau}^{-1/2} - (\mathbf{X} - \mathbf{1}_n \boldsymbol{\mu}^T)\boldsymbol{\tau}^{-1/2}\mathbf{U}\mathbf{U}^T\|_F^2$$

whereas our proposed normalization minimizes

$$\|(\mathbf{X} - \mathbf{1}_n \boldsymbol{\mu}^T)\boldsymbol{\tau}^{-1/2} - (\mathbf{X} - \mathbf{1}_n \boldsymbol{\mu}^T)\mathbf{U}\mathbf{U}^T\boldsymbol{\tau}^{-1/2}\|_F^2.$$

Bernoulli Distribution For Bernoulli variables, $\tau_j = -2(\bar{X}_j \log(\bar{X}_j) + (1 - \bar{X}_j)\log(1 - \bar{X}_j))$. This normalization factor is largest when $\bar{X}_j = 1/2$ and it decreases as $\bar{X}_j$ moves away from $1/2$. This is similar to the behavior of the variance estimate of a Bernoulli variable, $\bar{X}_j(1 - \bar{X}_j)$. Therefore, variables with means close to 0 or 1 will get more weight than those with means close to $1/2$.

Poisson Distribution For Poisson variables,

$$\tau_j = -2\left(\bar{X}_j \log(\bar{X}_j) - \frac{1}{n}\sum_i x_{ij}\log(x_{ij})\right),$$

which is not a simple function of the mean, like for Gaussian or Bernoulli variables. To see the relation between the normalization factor and the Poisson mean parameter λ, we obtained numerical estimates of the limit of the average null deviance as n goes to ∞ for the

Poisson distribution, which equals $-2\left[\lambda\log(\lambda) - E(X\log(X))\right]$. Figure 2 shows this limit as a function of λ. When the sample size is large, τ_j is expected to be smallest when λ tends towards 0 and to flatten out when λ is greater than 1. This implies that variables with means smaller than 1 get the most weight and that two variables with means much larger than 1 will get approximately the same weight. In contrast to the Bernoulli distribution, the normalization factor is not monotonically related to the variance of a Poisson variable, which equals the mean.

2.5.4 Multi-Parameter Exponential Family Data

For discussion of generalized PCA formulation so far, we have implicitly assumed that data are from a one-parameter exponential family distribution. To expound on multi-parameter exponential family data, we consider categorical data. As an extension of a Bernoulli distribution, a multinomial distribution can be posited for a categorical variable with more than two categories. Suppose that the categorical variable has K possible outcomes. This categorical variable can be represented by $K-1$ cell counts or proportion variables, where each variable represents the count or proportion of times that category is taken.

The number of trials associated with each case may differ depending on the context. In many scenarios, each case may involve only one trial. A scenario with multiple trials naturally arises in modeling text data, where a case may be a document and the different categories would be the words used in the document. Such data can be arranged into a so-called document-term frequency matrix.

Suppose that $(x_{i1}, \ldots, x_{iK})$ is the ith case with x_{ij} representing the proportion of the jth category from n_i multinomial trials with cell probabilities p_{ij}. The log-likelihood based on the ith case is given by

$$n_i \sum_{j=1}^{K-1} (x_{ij}\theta_{ij}) - n_i \log\left(1 + \sum_{j=1}^{K-1} e^{\theta_{ij}}\right),$$

where $\theta_{ij} = \log(p_{ij}/p_{iK}) \in \mathbb{R}$ and n_i, the number of trials for the ith case, acts as a weight. With the multi-parameter distribution, $b_j(\cdot)$ is not a separate function for each of the $K-1$ categories. Instead, there is one $b(\cdot)$ function for the whole categorical variable,

$$b(\boldsymbol{\theta}_i) = \log\left(1 + \sum_{l=1}^{K-1} e^{\theta_{il}}\right).$$

The natural parameter from the saturated model, $\tilde{\theta}_{ij}$, is similar to the Bernoulli case, but now depends on both x_{ij} and $x_{iK} := 1 - \sum_{j=1}^{K-1} x_{ij}$. As for other distributions, we must approximate ∞ with a large number m, and specify

$$\tilde{\theta}_{ij} = \begin{cases} -m & \text{if } x_{ij} = 0 \\ \log(x_{ij}/x_{iK}) & \text{if } x_{ij} \in (0,1) \text{ and } x_{iK} \in (0,1) \\ m + \log(x_{ij}) & \text{if } x_{ij} \in (0,1) \text{ and } x_{iK} = 0 \\ m & \text{if } x_{ij} = 1 \end{cases}. \tag{2.5}$$

Lemma 2 in Appendix A.3 shows that as m gets very large, these parameters will converge to the perfect fit:

$$\left.\frac{\partial b(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}_i} \longrightarrow \mathbf{x}_i.$$

3 Computation

Since orthonormal constraints are not convex (Wen and Yin, 2013), finding a global solution for generalized PCA is infeasible in most scenarios. To deal with the difficulties imposed by orthonormality, we use an MM algorithm (Hunter and Lange, 2004). This algorithm is somewhat of an extension to the algorithm proposed for finding the loadings of logistic PCA in Landgraf and Lee (2015). However, the deviance cannot be majorized for some members of the exponential family, for example Poisson, in contrast to the Bernoulli case. To deal with this, we first approximate the deviance with a second-order Taylor expansion and then majorize the Taylor expansion. Further, we modify the objective by relaxing its domain to the convex hull of projection matrices and propose a general algorithm for minimizing the new objective.

3.1 MM Algorithm

With a Taylor approximation, we iteratively replace a smooth objective with a second-order quadratic approximation. Considering a single scalar parameter θ, let $Q(\theta|\theta^{(t)})$ be the second-order quadratic approximation of the objective function around the most recent estimate, $\theta^{(t)}$. In many cases this quadratic approximation can be minimized with a closed-form solution. For example, in the iteratively reweighted least squares algorithm for generalized linear models (GLMs), the log-likelihood is approximated by a quadratic function, which leads to a weighted least squares problem.

A Taylor approximation alone may still result in a difficult problem to solve for the generalized principal component loadings. Therefore, we additionally majorize the quadratic approximation. Much like the Taylor approximation, MM algorithms iteratively replace the difficult objective with a simpler one. A function $M(\theta|\theta^{(t)})$ majorizes an objective $f(\theta)$ at $\theta^{(t)}$ if it satisfies two conditions: the tangency condition, $M(\theta^{(t)}|\theta^{(t)}) = f(\theta^{(t)})$, and the domination condition, $M(\theta|\theta^{(t)}) \geq f(\theta)$ for all θ.

The ijth element of the weighted deviance, $w_{ij}D(x_{ij};\theta_{ij})$, is quadratically approximated at $\theta_{ij}^{(t)}$ by

$$Q(\theta_{ij}|\theta_{ij}^{(t)}) := w_{ij}D(x_{ij};\theta_{ij}^{(t)}) + 2w_{ij}(b_j'(\theta_{ij}^{(t)}) - x_{ij})(\theta_{ij} - \theta_{ij}^{(t)}) + w_{ij}b_j''(\theta_{ij}^{(t)})(\theta_{ij} - \theta_{ij}^{(t)})^2.$$

From exponential family theory, $b_j'(\theta_{ij}^{(t)}) = E_{\theta_{ij}^{(t)}}(X_{ij})$ and $b_j''(\theta_{ij}^{(t)}) \propto \text{var}_{\theta_{ij}^{(t)}}(X_{ij})$.

Even though the Taylor approximation is quadratic, it can still be difficult to optimize with orthonormal constraints because of non-constant $b_j''(\theta_{ij}^{(t)})$. Therefore, we majorize the

ijth element of the quadratic approximation with

$$M(\theta_{ij}|\theta_{ij}^{(t)}) := w_{ij}D(x_{ij};\theta_{ij}^{(t)}) + 2w_{ij}(b_j'(\theta_{ij}^{(t)}) - x_{ij})(\theta_{ij} - \theta_{ij}^{(t)}) + v_i^{(t)}(\theta_{ij} - \theta_{ij}^{(t)})^2,$$

where

$$v_i^{(t)} := \max_j w_{ij}b_j''(\theta_{ij}^{(t)}). \tag{3.1}$$

Clearly, the tangency condition is satisfied for both the quadratic approximation and the deviance, $M(\theta_{ij}^{(t)}|\theta_{ij}^{(t)}) = Q(\theta_{ij}^{(t)}|\theta_{ij}^{(t)}) = w_{ij}D(x_{ij};\theta_{ij}^{(t)})$, and the domination condition is satisfied for the quadratic approximation, $M(\theta|\theta_{ij}^{(t)}) \geq Q(\theta|\theta_{ij}^{(t)})$ for all θ.

Finding the principal component loadings using this approach requires initializing $\boldsymbol{\Theta} = \mathbf{1}_n\boldsymbol{\mu}^T + (\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{U}\mathbf{U}^T$ with $\mathbf{U}^{(1)}$ and $\boldsymbol{\mu}^{(1)}$ and iteratively minimizing

$$\begin{aligned} M(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)}) &= \sum_{i,j} M(\theta_{ij}|\theta_{ij}^{(t)}) \\ &= \sum_{i,j} v_i^{(t)} \left[\theta_{ij} - \left(\theta_{ij}^{(t)} + \frac{w_{ij}}{v_i^{(t)}}\left(x_{ij} - b_j'(\theta_{ij}^{(t)})\right)\right)\right]^2 + \text{constant}, \end{aligned}$$

with respect to $\boldsymbol{\mu}$ and $\mathbf{U}$.

Let

$$z_{ij}^{(t)} := \theta_{ij}^{(t)} + \frac{w_{ij}}{v_i^{(t)}}\left(x_{ij} - b_j'(\theta_{ij}^{(t)})\right) \tag{3.2}$$

be the adjusted response variables, $\mathbf{Z}^{(t)} = [z_{ij}^{(t)}]$, and $\mathbf{V}^{(t)}$ be an $n \times n$ diagonal matrix with ith diagonal equal to $v_i^{(t)}$. The majorizing function can be rewritten as

$$M(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)}) = \left\|\left(\mathbf{V}^{(t)}\right)^{1/2}\left(\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T\right)\mathbf{U}\mathbf{U}^T - \left(\mathbf{V}^{(t)}\right)^{1/2}\left(\mathbf{Z}^{(t)} - \mathbf{1}_n\boldsymbol{\mu}^T\right)\right\|_F^2 + \text{constant}.$$

Holding $\mathbf{U}$ fixed, $M(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)})$ is minimized with respect to $\boldsymbol{\mu}$ when

$$\boldsymbol{\mu}^{(t+1)} = \left(\mathbf{1}_n^T\mathbf{V}^{(t)}\mathbf{1}_n\right)^{-1}\left(\mathbf{Z}^{(t)} - \tilde{\boldsymbol{\Theta}}\mathbf{U}^{(t)}(\mathbf{U}^{(t)})^T\right)^T\mathbf{V}^{(t)}\mathbf{1}_n.$$

Let $\tilde{\boldsymbol{\Theta}}_c^{(t)} := \tilde{\boldsymbol{\Theta}} - \mathbf{1}_n(\boldsymbol{\mu}^{(t+1)})^T$ and $\mathbf{Z}_c^{(t)} := \mathbf{Z}^{(t)} - \mathbf{1}_n(\boldsymbol{\mu}^{(t+1)})^T$ be the column-centered saturated parameters and adjusted responses. Similar to the result in Landgraf and Lee (2015) for logistic PCA, when holding $\boldsymbol{\mu}$ fixed, $M(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)})$ is minimized with respect to orthonormal $\mathbf{U}$ by the first k eigenvectors of

$$\left(\tilde{\boldsymbol{\Theta}}_c^{(t)}\right)^T\mathbf{V}^{(t)}\mathbf{Z}_c^{(t)} + \left(\mathbf{Z}_c^{(t)}\right)^T\mathbf{V}^{(t)}\tilde{\boldsymbol{\Theta}}_c^{(t)} - \left(\tilde{\boldsymbol{\Theta}}_c^{(t)}\right)^T\mathbf{V}^{(t)}\tilde{\boldsymbol{\Theta}}_c^{(t)}. \tag{3.3}$$

The complete proof of the minimizers of $M(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)})$ is in Appendix A.4.

Algorithm 1 summarizes the steps of the MM algorithm for finding the loadings and main effects of generalized PCA. This algorithm is similar to iteratively reweighted least squares

input : Data matrix ($\mathbf{X}$), m, number of principal components (k), weight matrix ($\mathbf{W}$)

output: $d \times k$ orthonormal matrix of principal component loadings ($\mathbf{U}$) and column main effects ($\boldsymbol{\mu}$)

Set $t = 1$ and initialize $\boldsymbol{\mu}^{(1)}$ and $\mathbf{U}^{(1)}$. We recommend setting $\boldsymbol{\mu}^{(1)}$ to the column means of $\tilde{\boldsymbol{\Theta}}$ and $\mathbf{U}^{(1)}$ to the first k right singular vectors of column-centered $\tilde{\boldsymbol{\Theta}}$

repeat

1. Set the weights and the adjusted response variables using equations (3.1) and (3.2), where $\theta_{ij}^{(t)} = \mu_j^{(t)} + [\mathbf{U}^{(t)}(\mathbf{U}^{(t)})^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu}^{(t)})]_j$
2. $\boldsymbol{\mu}^{(t+1)} \leftarrow \left(\mathbf{1}_n^T \mathbf{V}^{(t)} \mathbf{1}_n\right)^{-1} \left(\mathbf{Z}^{(t)} - \tilde{\boldsymbol{\Theta}}\mathbf{U}^{(t)}(\mathbf{U}^{(t)})^T\right)^T \mathbf{V}^{(t)} \mathbf{1}_n$
3. Set $\mathbf{U}^{(t+1)}$ to the first k eigenvectors of (3.3)
4. $t \leftarrow t + 1$

until *Deviance converges*;

Algorithm 1: Majorization-minimization algorithm for generalized PCA

for GLMs. At each iteration, an adjusted response variable $z_{ij}^{(t)}$ is generated, which is equal to the previous estimate plus the residual weighted by the inverse of the estimated variance. In our algorithm, a case-wise upper bound on the estimated variance is used instead. This makes minimization of $M(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)})$ a weighted least squares problem, which can be solved by an eigen-decomposition.

Other options are available for majorizing $Q(\theta_{ij}|\theta_{ij}^{(t)})$. With the terminology of Lee et al. (2010), the majorization above uses a "tight" bound. We could instead use a "uniform" bound by defining $v_i^{(t)} := \max_{i,j} w_{ij} b_j''(\theta_{ij}^{(t)})$, which is the same for all i. There is a trade-off between the two bounds. It is more computationally efficient to calculate each iterate with the uniform bound since each iteration involves an un-weighted least squares problem, but the tight bound takes fewer iterations to converge. We use the tight bound in the data analyses presented in this paper.

3.1.1 Comparison of Uniform and Tight Majorizations

To illustrate this trade-off, we performed logistic PCA with $k = 2$ on the binary dataset of disease indicators from patients admitted to the intensive care unit, which is described in Landgraf and Lee (2015). For binary data, proper bounds of the estimated variances depend on the fitted probabilities. We consider two settings with different ranges of fitted probabilities. First, we used only the 318 variables that had column means closest to 0 out of the 584 variables. For the Bernoulli distribution, $b''(\hat{\theta}_{ij}) = \hat{p}_{ij}(1-\hat{p}_{ij})$ is smallest when the fitted probability is close to 0 or 1 and largest when the fitted probability is 0.5. If a small fraction of fitted probabilities in the whole matrix are close to 0.5 and the rest are close to

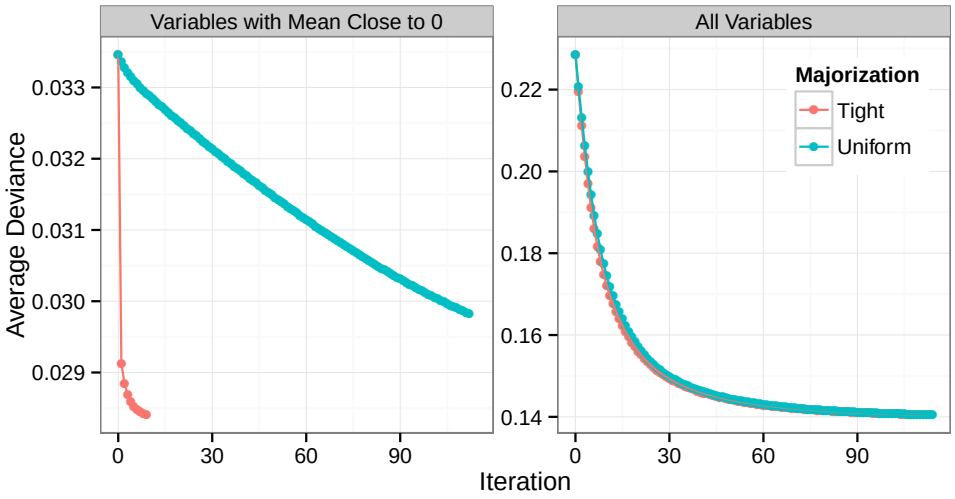

Figure 3: Comparison of the tight and uniform majorizations in the MM algorithm applied to binary datasets. Most of the variables in the data used for the left panel have means close to 0 and the full dataset used in the right panel has some variables with means close to 0.5

0 or 1, the uniform majorization will be very loose compared to the tight majorization.

The left panel of Figure 3 shows the average Bernoulli deviance as a function of the number of iterations for the two types of majorization. The tight majorization entails many fewer iterations, and achieves a smaller deviance after only one iteration. At convergence, around 98% of the rows had maximum fitted probabilities less than 0.05, but a few rows had fitted probabilities close to 0.5, which causes the large discrepancy between the tight and uniform majorizations.

Second, we included all of the 584 variables, a few of which have column means close to 0.5. This time, most of the rows had a variable with a fitted probability close to 0.5. This resulted in little difference between the tight and uniform majorizations, as can be seen in the right panel of Figure 3. Finally, note that the MM algorithm presented in Landgraf and Lee (2015) for logistic PCA is similar to the algorithm for generalized PCA with the uniform bound.

3.1.2 MM Algorithm with Missing Data

So far, we have assumed that we have a complete $n \times d$ matrix for dimensionality reduction. Elements may be missing from the dataset for a variety of reasons. Often, as in collaborative filtering, the goal of dimensionality reduction is to fill in, or complete, the missing elements of the matrix.

To find the loadings and main effects with missing data, we use an MM algorithm which is very similar to the expectation maximization (EM) algorithm (Dempster et al., 1977).

With missing data, a new majorization function is derived as

$$\begin{aligned}
M(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)}) &= \sum_{(i,j)\in\Omega} M(\theta_{ij}|\theta_{ij}^{(t)}) \\
&= \sum_{(i,j)\in\Omega} v_i^{(t)} \left[\theta_{ij} - \left(\theta_{ij}^{(t)} + \frac{w_{ij}}{v_i^{(t)}}\left(x_{ij} - b_j'(\theta_{ij}^{(t)})\right)\right)\right]^2 + \text{constant} \\
&\leq \sum_{(i,j)\in\Omega} v_i^{(t)} \left[\theta_{ij} - \left(\theta_{ij}^{(t)} + \frac{w_{ij}}{v_i^{(t)}}\left(x_{ij} - b_j'(\theta_{ij}^{(t)})\right)\right)\right]^2 + \sum_{(i,j)\notin\Omega} v_i^{(t)}\left[\theta_{ij} - \theta_{ij}^{(t)}\right]^2 + \text{constant} \\
&:= N(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)}).
\end{aligned}$$

This is effectively the same majorization used in Algorithm 1, with $w_{ij} = 0$ when $(i,j) \notin \Omega$.

As in many EM algorithms, this is equivalent to filling in the missing elements of the natural parameter matrix with the latest estimated value. This strategy has been used by others for matrix factorization in the presence of missing data (e.g. Mazumder et al., 2010; Lee et al., 2010). Therefore, a solution to generalized PCA with missing data can be found by iteratively minimizing

$$N(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)}) = \left\| \left(\mathbf{V}^{(t)}\right)^{1/2} \left(\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T\right)\mathbf{U}\mathbf{U}^T - \left(\mathbf{V}^{(t)}\right)^{1/2}\left(\mathbf{Z}^{(t)} - \mathbf{1}_n\boldsymbol{\mu}^T\right)\right\|_F^2 + \text{constant}.$$

where the definitions of $v_i^{(t)}$ and $z_{ij}^{(t)}$ are the same as before with augmented weights of 0 for the missing data. The same operations as in Algorithm 1 can be used to minimize $N(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)})$.

3.2 Convex Relaxation

The MM algorithm makes the non-convex problem tractable by reducing it to a series of problems where the solution is known. Another strategy is to relax the problem. Instead of projection matrices $\mathbf{U}\mathbf{U}^T$, we may look for matrices in the convex hull of projection matrices, called the Fantope (Dattorro, 2005), which is defined as

$$\mathcal{F}^k = \text{conv}\{\mathbf{U}\mathbf{U}^T \mid \mathbf{U} \in \mathbb{R}^{d\times k},\ \mathbf{U}^T\mathbf{U} = \mathbf{I}_k\} = \{\mathbf{H} \mid \mathbf{0} \preceq \mathbf{H} \preceq \mathbf{I}_d,\ tr(\mathbf{H}) = k\}.$$

This convex relaxation reduces the generalized dimensionality reduction task to

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in\Omega} w_{ij} D_j(x_{ij}; \mu_j + [\mathbf{H}(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})]_j) \\
\text{subject to} \quad & \boldsymbol{\mu} \in \mathbb{R}^d \text{ and } \mathbf{H} \in \mathcal{F}^k.
\end{aligned}$$

For fixed $\boldsymbol{\mu}$ the problem is convex in $\mathbf{H}$, and for fixed $\mathbf{H}$ the problem is convex in $\boldsymbol{\mu}$, i.e. it is a bi-convex problem. With either $\boldsymbol{\mu}$ or $\mathbf{H}$ fixed, there are theoretical guarantees to finding the other parameter with minimum deviance. We propose to solve for $\boldsymbol{\mu}$ first and

keep it fixed while solving for $\mathbf{H}$. With $\mathbf{H} = \mathbf{0}$, the deviance is minimized with respect to $\boldsymbol{\mu}$ if $\mu_j = g_j\left(X_j^T W_j / (\mathbf{1}_n^T W_j)\right)$, for $j = 1, \ldots, d$.

To solve the relaxed problem with respect to $\mathbf{H}$ with fixed $\boldsymbol{\mu}$, we use the projected gradient method (Boyd et al., 2003), which involves updating $\mathbf{H}$ at each iteration by

$$\mathbf{H}^{(t+1)} = \Pi_{\mathcal{F}^k}\left(\mathbf{H}^{(t)} - \alpha_t \mathbf{G}^{(t)}\right).$$

$\mathbf{G}^{(t)}$ is the gradient of the deviance with respect to $\mathbf{H}$ evaluated at $\mathbf{H}^{(t)}$, α_t is the step size for the tth iteration, and $\Pi_{\mathcal{F}^k}(\cdot)$ is the Euclidean projection operator onto the convex set $\mathcal{F}^k$.

This method is guaranteed to produce a solution arbitrarily close to the global minimum as the number of iterations goes to ∞ as long as the step size, α_t, decreases in an appropriate manner. One way to ensure convergence is to have the step sizes be square summable, but not summable (Boyd et al., 2003), which can be achieved with $\alpha_t = a_0/t$, for instance, where a_0 is the starting step size.

Without taking into account the symmetry of $\mathbf{H}$, the gradient of the deviance with respect to $\mathbf{H}$ can be derived in a similar way to (2.2) and is given by

$$\frac{\partial D}{\partial \mathbf{H}} = 2\left(\mathbf{W} \circ (b'(\hat{\boldsymbol{\Theta}}) - \mathbf{X})\right)^T \left(\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n \boldsymbol{\mu}^T\right), \tag{3.4}$$

where $\circ$ is the Hadamard (entrywise) product operator. The proof is given in Appendix A.5. When x_{ij} is missing, $w_{ij} = 0$, which causes it to have no influence on the gradients. Since $\mathbf{H}$ is symmetric, the gradient needs to be symmetrized as follows (see Petersen and Pedersen (2012)):

$$\mathbf{G} = \left[\frac{\partial D}{\partial \mathbf{H}}\right] + \left[\frac{\partial D}{\partial \mathbf{H}}\right]^T - \operatorname{diag}\left[\frac{\partial D}{\partial \mathbf{H}}\right]. \tag{3.5}$$

To update $\mathbf{H}$ in the negative gradient direction, the projection operator $\Pi_{\mathcal{F}^k}$ has to be defined explicitly. Vu et al. (2013) showed that the Euclidean projection of a positive definite matrix $\mathbf{A}$ with a spectral decomposition $\mathbf{A} = \sum_{j=1}^d \lambda_j \mathbf{u}_j \mathbf{u}_j^T$ onto $\mathcal{F}^k$ is given by

$$\Pi_{\mathcal{F}^k}(\mathbf{A}) = \sum_{j=1}^d \lambda_j^+(\nu) \mathbf{u}_j \mathbf{u}_j^T, \tag{3.6}$$

where $\lambda_j^+(\nu) = \min(\max(\lambda_j - \nu, 0), 1)$ with ν that satisfies $\sum_j \lambda_j^+(\nu) = k$. A quick and easy way to find ν is bisection search, with an initial lower bound of $\lambda_d - k/d$ and an initial upper bound of λ_1, the largest eigenvalue. Algorithm 2 presents the complete algorithm for the Fantope solution.

Although there are guarantees for iterative solutions from the projected gradient method to converge to the global minimum, it could potentially take many iterations. In practice, we have found that having a good initial starting gradient descent step size, a_0, can be very important. One solution that works fairly well is to take the inverse of the second

input : Data matrix ($\mathbf{X}$), m, rank (k), weight matrix ($\mathbf{W}$), starting step size (a_0)
output: $\boldsymbol{\mu}$ and $d \times d$ rank k Fantope matrix ($\mathbf{H}$)

Set $t = 0$ and $\mu_j = g_j\left(X_j^T W_j / (\mathbf{1}_n^T W_j)\right)$, $j = 1, \ldots, d$
Set $\mathbf{U}$ to the first k right singular vectors of $\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n \boldsymbol{\mu}^T$
Initialize $\mathbf{H}^{(0)} = \mathbf{U}\mathbf{U}^T$
repeat

1. $t \leftarrow t + 1$. Set $\alpha_t = a_0 \frac{1}{t}$
2. Calculate the derivative, $\mathbf{G}^{(t-1)}$, of the deviance at $\mathbf{H}^{(t-1)}$ using equations (3.4) and (3.5)
3. Move in the negative direction of the derivative and project onto the rank-k Fantope using equation (3.6), $\mathbf{H}^{(t)} = \Pi_{\mathcal{F}^k}\left(\mathbf{H}^{(t-1)} - \alpha_t \mathbf{G}^{(t-1)}\right)$

until *Deviance converges*;

Algorithm 2: Fantope algorithm for the convex relaxation of generalized PCA

derivative with respect to each component of $\mathbf{H}$, evaluated at the initial value, and average them together, i.e.

$$a_0 = \frac{1}{d^2} \sum_{j,l} \left[\left. \frac{\partial^2 D}{\partial \mathbf{H}_{jl}^2} \right|_{\mathbf{H}_{jl}^{(0)}} \right]^{-1} = \frac{1}{d^2} \sum_{j,l} \left[2 \sum_{i=1}^{n} w_{ij} b_j'' \left(\hat{\theta}_{ij}^{(0)} \right) \left(\tilde{\theta}_{il} - \mu_l \right)^2 \right]^{-1}.$$

As stated in Landgraf and Lee (2015), there are practical differences to the solutions from the original formulation and the convex relaxation. The Fantope solutions may not be low-rank, even if k is small, and therefore are not as useful if data compression or visualization are of interest. If the goal is approximation of the natural parameter matrix itself, the Fantope solution may be more efficient to calculate due to convexity, and may give better predictions. We give an example in Section 5.2 where the convex relaxation gives more accurate predictions for a collaborative filtering problem.

An R (R Core Team, 2015) implementation of these algorithms can be found at `github.com/andland/generalizedPCA`.

4 Simulation Studies

We present two data analyses of simulated count data in this section. First, we simulate datasets with missing data and compare how well standard PCA and Poisson PCA fill in the missing values on both training and test data. Second, we examine how the characteristics of the count data affect the quality of recommendations, and which distributional assumption of generalized PCA has the biggest advantage in various scenarios. For both examples, we

simulate count datasets, controlling two specified characteristics: the level of sparsity – the proportion of zero counts – and the average of the non-zero counts.

4.1 Simulation Setup

In simulating count data, we assumed that there is a latent cluster variable C_i for each row and that $C_i \sim$ Discrete Uniform$(1, \ldots, 5)$. Next we assumed that $X_{ij}|C_i = k \sim$ Poisson(λ_{jk}), where $\mathbf{\Lambda} = [\lambda_{jk}]$ is a $d \times 5$ matrix of positive means. The elements of $\mathbf{\Lambda}$ were simulated from a gamma distribution. This latent structure implies that both the mean matrix and the natural parameter (logarithm of mean) matrix have a rank of 5.

When $P(X|\lambda)$ is the Poisson probability mass function and $P(\lambda)$ is the gamma probability density function, the marginal distribution of X is negative binomial. We chose the shape and scale parameters of the gamma distribution in order to have counts with a specified proportion of zeros and mean of the non-zero counts.

If the gamma distribution has a shape parameter r and scale parameter $\phi = p/(1-p)$, then the negative binomial distribution has number of failures r and probability of success p. Under these assumptions, let

$$\gamma_1 := P(X = 0) = (1-p)^r \text{ and}$$
$$\gamma_2 := E(X|X > 0) = \frac{1}{1-(1-p)^r}\frac{pr}{(1-p)}.$$

Using the relations, we can determine the values of probability p and shape parameter r that match the specified level of sparsity and the expected non-zero counts. For given γ_1 and γ_2, the shape parameter is the solution to the following equation

$$\log(r) - \frac{1}{r}\log(\gamma_1) - \log(\gamma_2 - \gamma_1\gamma_2 + r) = 0$$

and $p = 1 - \gamma_1^{1/r}$.

In our simulations, we varied $\gamma_1 = P(X = 0)$ from 0.25 to 0.9 and $\gamma_2 = E(X|X > 0)$ from 3 to 15. For each setting, we simulated a training dataset with 100 rows ($n = 100$) and 50 columns ($d = 50$). We also simulated a test dataset using the same $\mathbf{\Lambda}$, but with 1000 rows. For each scenario, we simulated ten training and test datasets, and the summary presented below is the results averaged over the ten simulations.

When applying generalized PCA with distributions where it is necessary to provide m, we used five-fold cross validation on a range of 1 to 10 to select the optimal value with respect to predictive deviance. In estimating the principal component loadings of the simulated datasets, we did not include main effects for any of the simulations in order to highlight the performance peeking at the true rank.

4.2 Matrix Completion

Low-rank models are commonly used for matrix completion problems with missing or incomplete data (Feuerverger et al., 2012). Using the simulated count data, we examine whether

and under what conditions it is possible to improve the estimates of missing values. To emulate the situation with missing data, we randomly set 20% of the entries of the training matrix to be missing. We then applied both standard PCA and Poisson PCA to the simulated data with $k = 1, \ldots, 10$. We did not include multinomial PCA in this matrix completion setting because filling in the missing entries would require knowing the total number of counts for each row, which would give it an unfair advantage. Note that typical implementations of PCA cannot estimate the loadings in the presence of missing values, although there are some options available (e.g. Tipping and Bishop (1999)). We used the algorithm introduced in Section 3.1.2 for both standard PCA and Poisson PCA to handle missing data.

Filling in the missing values of the training matrix is a typical matrix completion problem. Using the projection formulation we have developed, a fitted value for x_{ij} that is missing is given by

$$\hat{x}_{ij} = g^{-1}\left(\mu_j + \left[\hat{\mathbf{U}}\hat{\mathbf{U}}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})\right]_j\right),$$

where $\tilde{\theta}_{ij}$ is zero since x_{ij} is missing and $\boldsymbol{\mu} = \mathbf{0}$ since we did not include main effects. $g^{-1}(\cdot)$ is the identity function for standard PCA and $\exp(\cdot)$ for Poisson PCA. For standard PCA, we set any fitted values below zero to zero, since negative values as counts are infeasible.

We evaluated the mean squared error of the filled-in estimates compared to the actual values held out. When there is no missing data, PCA is known to give an optimal low-rank projection of the data with respect to mean squared error. When there is missing data, this is not necessarily guaranteed. Poisson PCA involves the projection of a non-linear function (logarithm) of the data, which might be advantageous in some scenarios.

Figure 4 shows the mean squared prediction error of PCA and Poisson PCA on the missing data. Each row represents a different degree of sparsity, ranging from 25% zeros on the top row to 90% zeros on the bottom row. The columns represent the mean of the non-zero elements, ranging from 3 to 15. Within each panel, we estimated missing counts using both standard PCA and Poisson PCA with 1 to 10 components. Finally, the dashed horizontal lines represent the predicted mean squared error when using the column means as estimates for the missing values.

For either method, the minimum mean squared error occurs at around five components, which is the true rank of the mean and natural parameter matrices. In addition, both methods achieve mean squared errors well below the prediction error with only column means. For the datasets with the least amount of sparsity (the top row), standard PCA fills in the missing values more accurately. As the degree of sparsity increases, the advantage of Poisson PCA emerges. One reason for this may be that, when the count is 0, standard PCA penalizes fitted values below 0 as much as fitted values above 0. By contrast, for Poisson PCA it is not possible to have negative fitted values because the projection in the log space is properly transformed back to the feasible range for the mean parameter. Further, as outlined in Section 2.4, the Poisson deviance gives higher weight to the errors for smaller counts than larger counts and therefore attempts to fit small counts more closely.

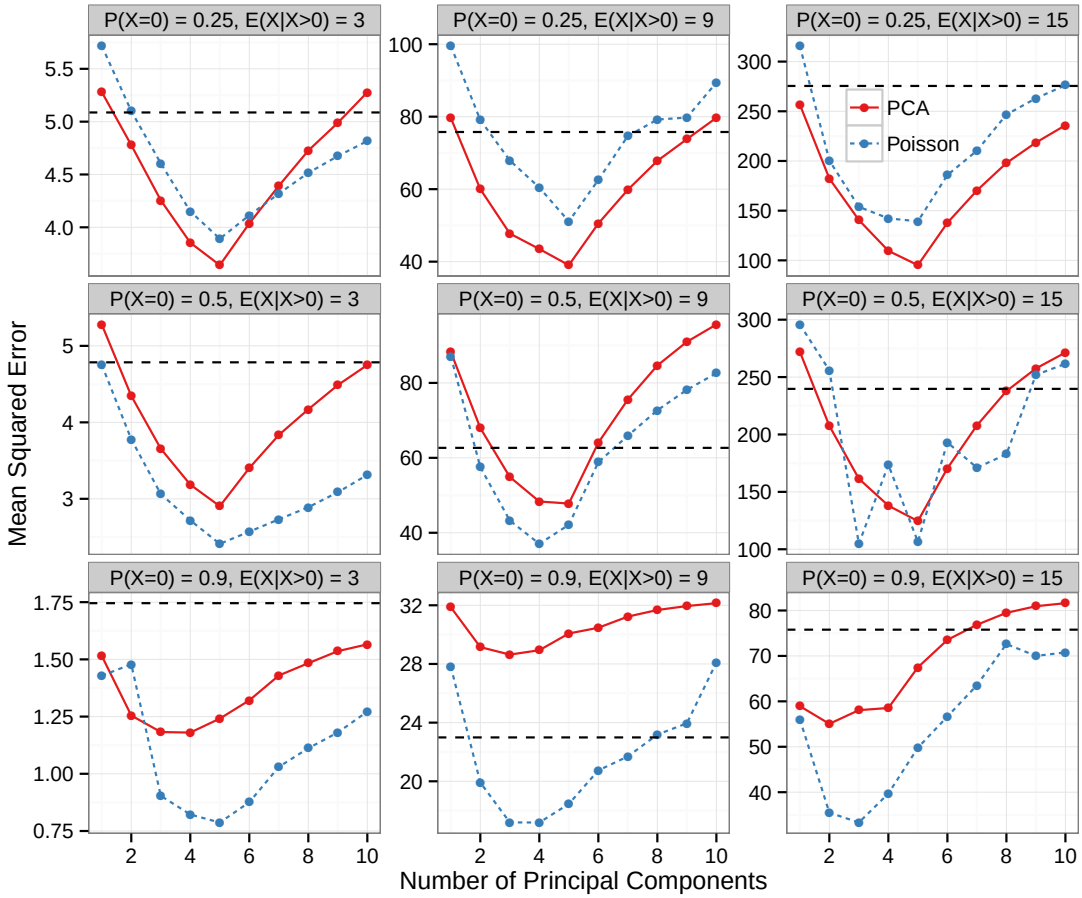

Figure 4: Mean squared error of standard PCA and Poisson PCA on withheld simulated count data in a matrix completion task. Each panel represents a different combination of the proportion of zeros and the mean of the counts greater than zero. The dashed horizontal lines represent the mean squared errors when using column means as predictions.

We also extended this simulation to fill in missing observations on the test data. For matrix factorization techniques, filling in the missing values on new test data requires additional computation. In contrast, our method can easily apply the same loadings learned on the training data to the test data. The results on the test data are essentially the same as those on the training data and are not included for brevity.

Computationally, this example first shows the capability of our MM algorithm to solve for the loadings of generalized PCA in the presence of missing data. It also shows that matrix completion can be done effectively on new observations. Finally, it shows that Poisson PCA is more effective than standard PCA at filling in the missing values for highly sparse count data, even in terms of mean squared error – the loss function PCA is optimized for.

4.3 Recommending Items to Users

As mentioned in the introduction, low-rank models have proved effective in recommender systems. In this context, a count matrix may represent the number of times that a user (row) has interacted with an item (column). Oftentimes in recommendation problems, the most important aspect of an interaction is whether a user interacted with an item, not how many times. Based on a user's interaction history, we wish to create a ranked list of items that the user has not interacted with and recommend to them the highest on the list.

Since ranking depends on the fitted values only through their ordering, this recommendation task allows us to compare the analyses of counts and their binarized version. Binarization means setting all counts greater than 0 to 1, which is a common transformation when dealing with user-item interaction data due to sparsity. From the generalized PCA toolbox, standard, Poisson, and multinomial PCA are applicable to count data while standard and logistic PCA are applicable to binary data.

To measure the ranking consistency of various methods with different distributional assumptions, we compare rankings of the items that new users have interacted with and those they have not, based on the simulated test data. To get an objective truth for testing, we randomly set five positive counts from each row in the test data to zero. Then we applied the projection matrices that were learned from the training data to the test data. To see how well we recommend items, we compare the rankings of the items where the counts were set to zero to all of the items where the counts were actually zero. Ideally, the rankings of the items where the counts were positive but set to zero will all be higher than the rankings of the items with actual zero counts. We measure the ranking accuracy using area under the ROC curve (AUC) for each user and average it across all of the users.

Under the simulation scenarios presented before, we would like to address a few questions:

- When analyzing count data, when can we expect improvement from Poisson PCA or multinomial PCA over standard PCA?
- When analyzing binarized data, when can we expect improvement from logistic PCA over standard PCA?
- When is analyzing count data more preferable to analyzing binarized data?

The analysis of the recommendations for the test data using the count data is in Figure 5, organized in the same way as the mean squared error analysis. Both Poisson PCA and multinomial PCA outperform standard PCA for most simulation scenarios. These methods are least advantageous for the top-left panel, with the least sparsity and smallest mean. Every simulation scenario has a maximum AUC at or near five principal components, the true rank. For the less sparse case (top row), it is better to over-fit with too many principal components than it is to under-fit with too few for each method. In the sparser case (bottom row), the opposite is true for standard PCA and the difference is less dramatic with the other two methods. In all of the scenarios, standard PCA is less effective than Poisson PCA or multinomial PCA. This numerical comparison suggests that it is preferable to use either Poisson PCA or multinomial PCA when analyzing count data for the purpose of ranking.

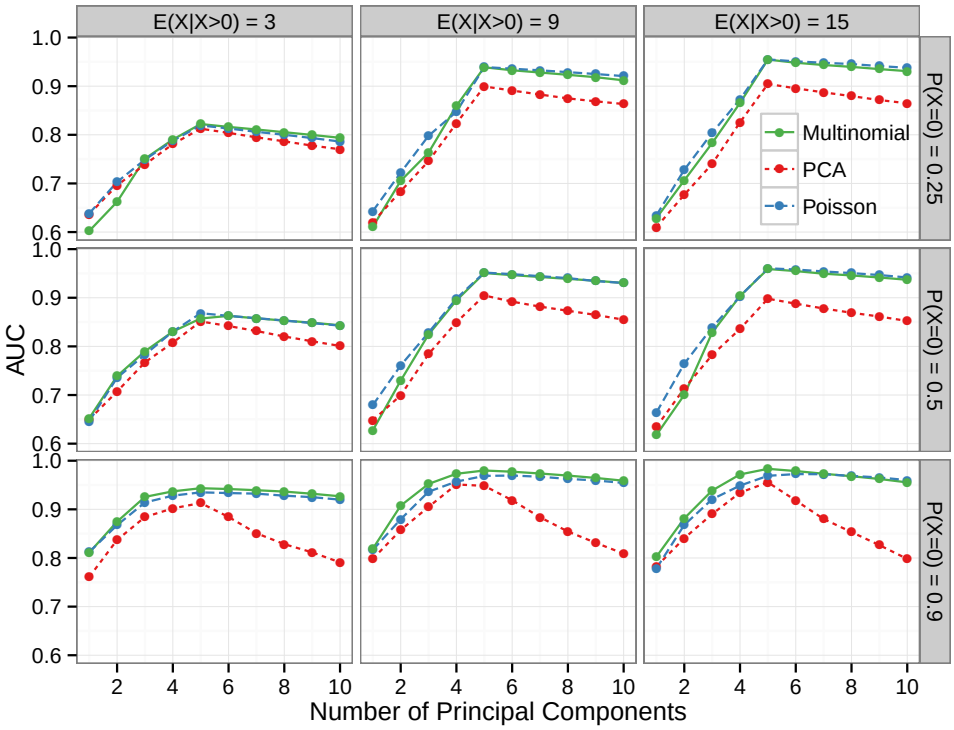

Figure 5: AUC of ranking the zero counts, both true zeros and positive counts which were set to zero, by different variations of PCA on the simulated count data

It is not surprising that Poisson PCA and multinomial PCA perform similarly. A multinomial distribution with K categories can be derived from K independent Poisson random variables, conditioning on the total count. Because of this, we expect the multinomial and Poisson formulations to perform similarly, especially when the row total counts ($N_i := \sum_j x_{ij}, i = 1, \ldots, n$) are nearly the same across the rows, as they are in this simulation. When there is substantial difference between the row totals, more difference between Poisson PCA and multinomial PCA will be observed. Say $x_{ij} = x_{ul} = 0$, but $N_i \gg N_u$. Then, in this setting, x_{ij} and x_{ul} are treated equally by the Poisson deviance whereas the multinomial deviance gives much higher weight to x_{ij} than x_{ul}. The example in Section 5.2 will illustrate this point further.

The results of the AUC analysis with the binarized version of the data are different, as shown in Figure 6. It shows that Logistic PCA and standard PCA on the binary data give similar AUCs, regardless of the simulation scenario. This is somewhat unexpected because it has been shown empirically that Collins et al. (2001)'s formulation is preferred to SVD for some recommendation problems with binary data. For example, Johnson (2014) showed that a weighted version of logistic SVD achieves the same ranking accuracy as a matrix factorization with a Gaussian assumption using a much smaller rank. Our results are possibly due to the simplicity of the simulation assumptions. For example, there is much

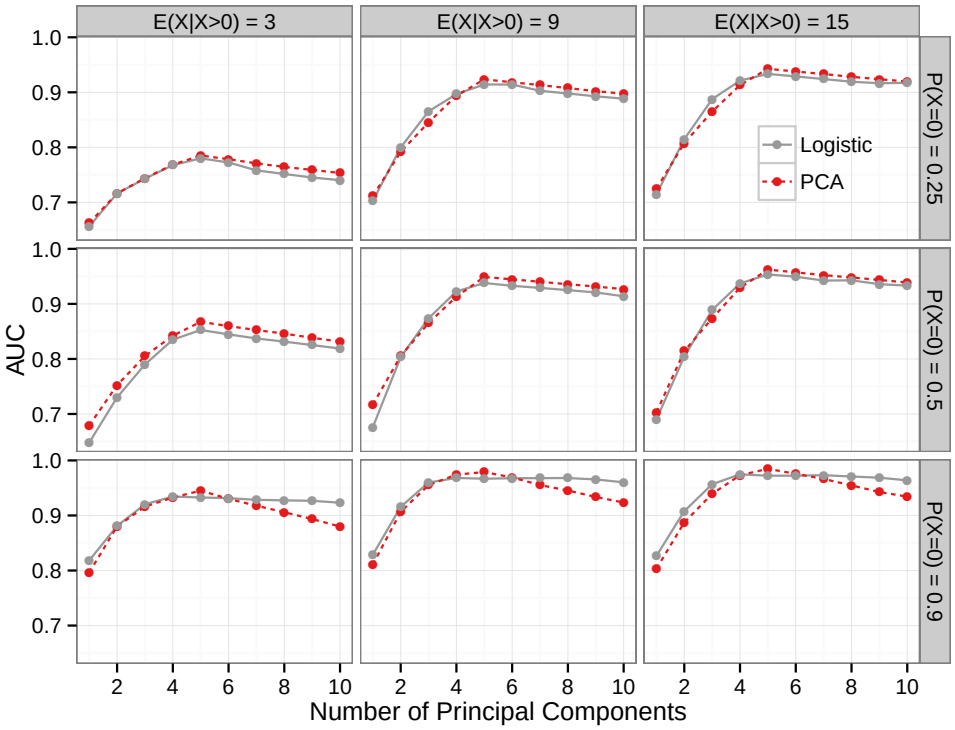

Figure 6: AUC of ranking the zero counts, both true zeros and positive counts which were set to zero, by logistic PCA and standard PCA on the simulated binarized data

larger dispersion in the number of non-zeros in the rows of the actual data in Section 5.2 than in the simulated data.

Finally, comparisons of the ranking accuracy using the count or binarized data are in Figure 7. We have only displayed Poisson and logistic PCA as they are representative of the two types of data. As might be expected, logistic PCA performs better when there is a higher degree of sparsity (bottom row). This may be because non-zero counts are rare, so modeling the existence of a non-zero is more important than modeling the value of the count. In contrast, when there is less sparsity, Poisson PCA can achieve a higher AUC. Finally, the data with higher means for the non-zero counts favor logistic PCA over Poisson PCA. This may be because predicting large counts to be non-zero is easier than predicting small counts to be non-zero.

5 Million Song Dataset Analysis

To show the benefits of generalized PCA, we analyze the Million Song Dataset (MSD) (Bertin-Mahieux et al., 2011). MSD contains the listening history of 110 thousand users on one million songs. For each user/song pair, the dataset lists the number of times that the user has listened to the song. Most of user/song pairs have a count of zero because

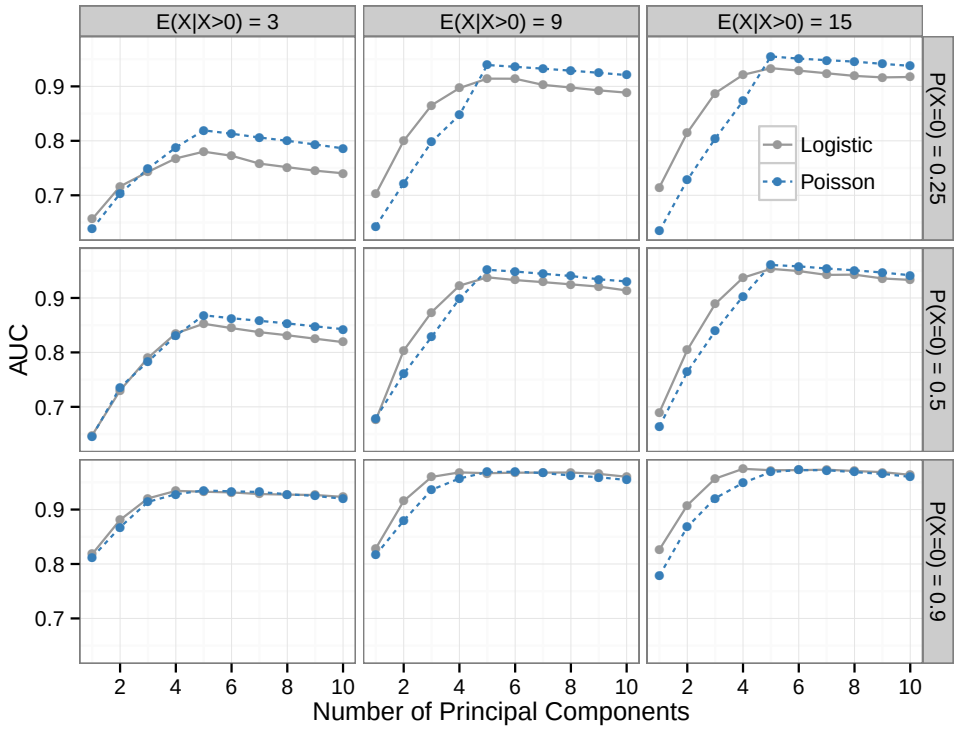

Figure 7: AUC of ranking the zero counts, both true zeros and positive counts which were set to zero, by Poisson PCA on the count data and logistic PCA on the binarized data

users only listen to a small subset of the songs. These data can be organized into a sparse "user-song frequency" matrix. Since the relationship between the songs in terms of the user's preference is of primary interest, the songs are taken as variables and the users are taken as cases. Generalized PCA is performed on two small subsets of the users and songs to show the benefits of the method for visualization and recommendation. Unlike with the simulations, we included a main effect in our analyses of MSD.

5.1 Visualizing the Loadings

The first subset of the MSD that we analyzed is the listen history on the songs from the ten most popular artists. We selected the 200 most popular songs by these artists and the 1200 most active listeners of those songs. There is quite a bit of dispersion in the count matrix created from these users and songs. 93% of the entries in the matrix are 0, the mean of the non-zero entries is 2.8, and the maximum count is 391.

For visualization of the songs in a low-dimensional latent space, we analyzed the count matrix using standard PCA on the counts and Poisson PCA. Since Poisson PCA projects the log of the counts, which are the natural parameters from the saturated model for the Poisson distribution, we also compared Poisson PCA to standard PCA on the log of the

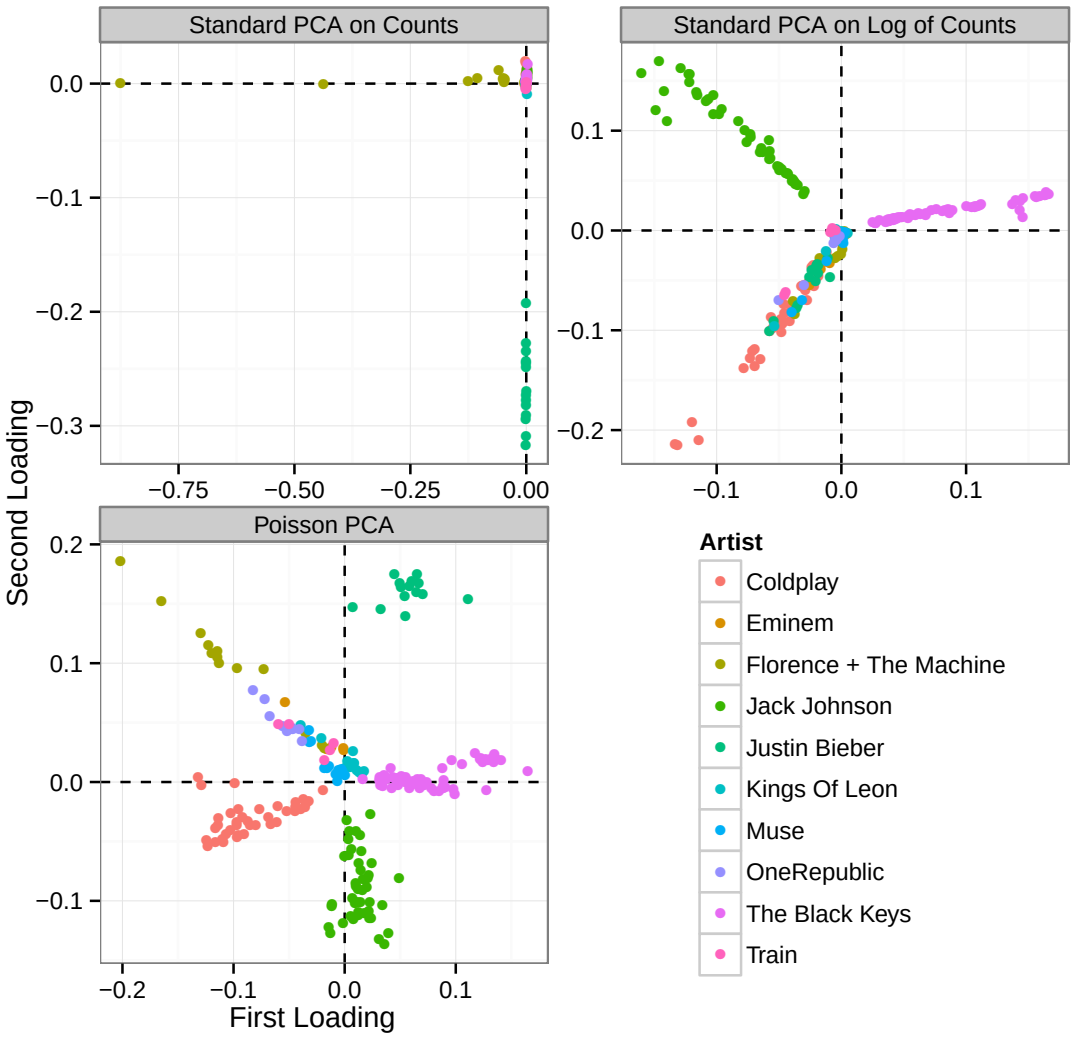

Figure 8: The first two principal component loadings of the million song dataset using standard PCA on the count data, standard PCA on the log of the count data, and Poisson PCA

counts. The log of zero is undefined, so we numerically approximate it by $-m$ with a large positive m for both Poisson PCA and standard PCA on the log transformed data. For each method, m was varied from 0.1 to 15 and was chosen by cross validation. For Poisson PCA, the optimal m was 8.0 and for standard PCA, it was 1.2.

The loadings for the three PCAs are in Figure 8. If a user listens to one song from an artist, it is likely that he or she will listen to other songs by the same artist. Therefore, we expect that songs by the same artists would be close in the latent space. For standard PCA on the count data, which is displayed on the top-left panel of Figure 8, the first component is dominated by Florence + the Machine's songs because the two largest observations in the

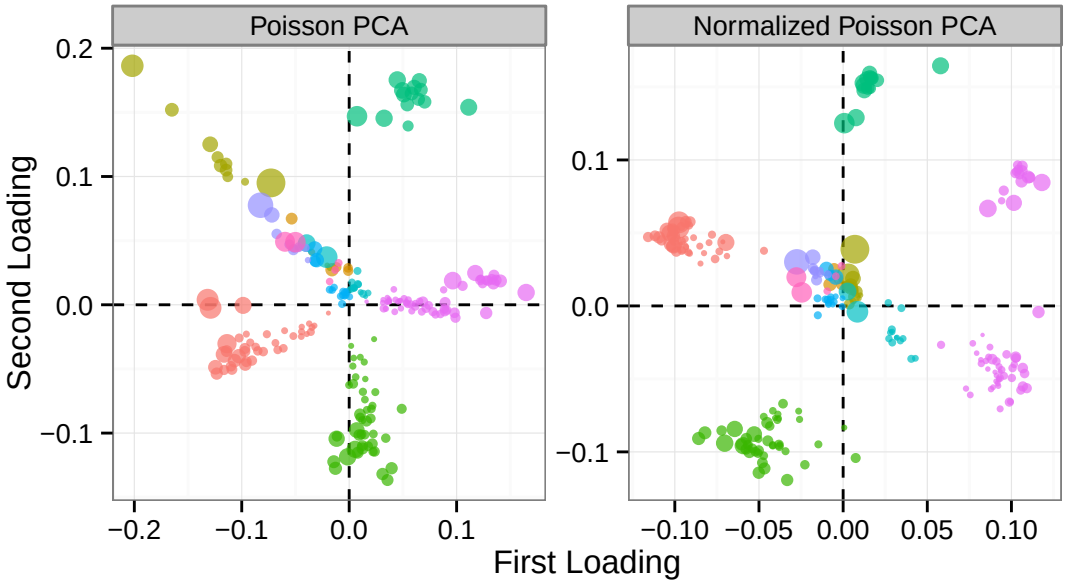

Figure 9: The first two principal component loadings of the million song dataset using Poisson PCA without normalization (left plot) and with normalization (right plot). The area of each point is proportional to the number of times the song was played

data were for songs by this artist. Songs by Justin Bieber dominate the second principal component direction because two songs by the artist were among the top ten most popular songs or variables.

In comparison to PCA on the count data, loadings from PCA on the log of the count data are easier to interpret, as displayed on the top-right panel of Figure 8. This shows the improvement due to the log transformation of the data. Songs by the Black Keys and Jack Johnson are distinguished from the rest of the artists. These two artists have the most songs in the data. However, songs by the other eight artists are clumped together.

Finally, the bottom-left panel of Figure 8 has the loadings from Poisson PCA. This plot shows the improvement from not only using the log transformation, but also optimizing the projection with respect to Poisson deviance, which is more natural for count data than mean squared error. Instead of only two dominant groups of songs corresponding to artists, there are now at least five distinguishable clusters of songs which correspond to artists. Further, for the Black Keys, there are two clusters, one of which in the first quadrant corresponds to songs from an album which had just been released.

The loadings for Poisson PCA with normalization factors proposed in Section 2.5.3 are displayed in Figure 9. The loadings of normalized PCA for the count or log counts are not included because they were qualitatively very similar to the loadings from the log of the counts. The left plot in Figure 9 shows the un-normalized Poisson PCA loadings and the right plot shows the normalized Poisson PCA loadings. The area of each point is proportional to the number of times the songs were played and the colors represent the same artists as in

Figure 8.

A notable difference between the sets of loadings is that songs by Florence + the Machine, which had very high play counts, are furthest away from the origin in the un-normalized version, but close to the origin in the normalized loadings. Also, the songs from the Black Keys new album are more distinctly clustered in the normalized version. Finally, in the un-normalized version, songs by artists cluster in ray shapes from the origin, while they cluster more tightly further away from the origin in the normalized version.

These differences can be explained by the normalization factors. The songs furthest from the origin in the un-normalized version are the songs that were played the most times. For the songs in this dataset, the Poisson null deviance was in general larger for songs with higher means. Therefore, the songs with the smallest average play counts – especially those less than 1 – were effectively given higher weights in the optimization when normalization is added. This caused the ray shapes to turn into tighter clusters and the songs by Florence + the Machine to have less influence.

5.2 Recommending New Songs to Users

In this second example, we started with the 500 most popular songs and then only included the songs that were listened to by at least 20 users and the users that listened to at least 30 of the songs. This resulted in 1040 users and 281 songs.

We consider recommending users songs that they have not listened to before. To do this, we randomly chose 80% of the users for training and 20% for testing. Similar to the simulation in Section 4.3, for each user in the test set, we randomly set five of the positive play counts to zero. We then applied the PC loadings and main effects learned from the training data to the test data to get a low-dimensional representation of each user's listens. We used this low-dimensional representation to rank the songs with zero plays, and compared how high the songs that were actually non-zero ranked using AUC. Finally we averaged the AUC from each user's prediction to get an overall AUC, which can range from 0 to 1, where 1 corresponds to perfect ranking and 0.5 corresponds to random guessing.

We have set up the experiment in a situation where typical matrix factorization methods may be computationally costly to use. For our method, computing the low-dimensional representation for new users only requires matrix multiplication, while it requires fitting a generalized linear model for each user with the matrix factorization-based methods.

First, we compare standard PCA and Poisson PCA using the count data directly. We also include multinomial PCA, where x_{ij} is the proportion of the plays of song j by the ith user and w_{ij} is the total number of times the ith user listened to a song, for all j. For all of the PCA methods, we vary the number of principal components from 2 to 50. For reference, we measured the AUC of rank order of the songs using the overall popularity of each song for every user, which ranks the songs based on the average play count of the songs. This popularity ranking is the same as the ranking that all of PCA methods would give with main effects but zero principal components. The results are in Figure 10.

Standard PCA with the play count data performs very poorly – even worse than the popularity model with no personalization – regardless of the number of principal compo-

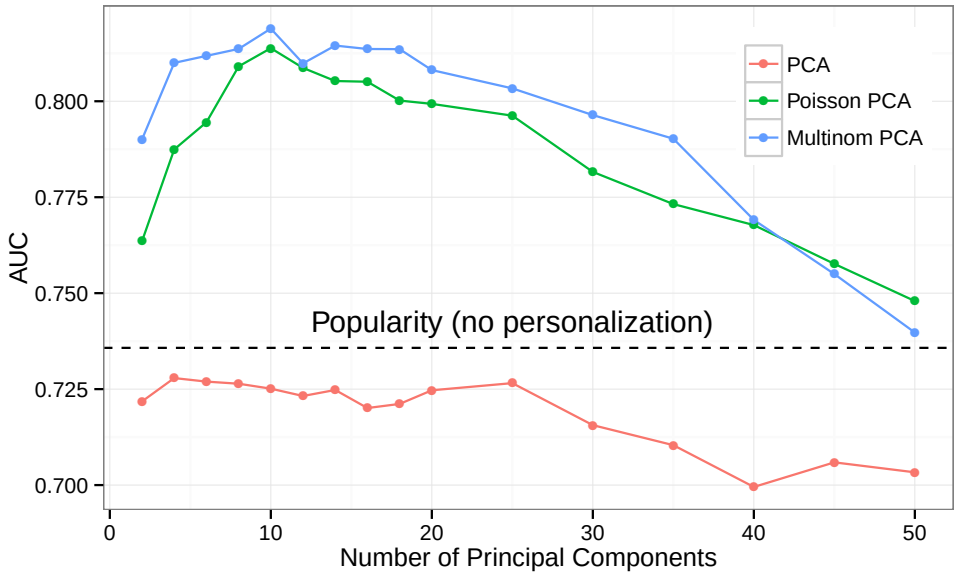

Figure 10: Comparison of different generalized PCA methods for recommending songs to users with the play count data

nents. Poisson PCA and multinomial PCA both do much better, peaking at around ten principal components. In this example, multinomial PCA has consistently higher AUC than Poisson PCA. The multinomial assumption may be more appropriate for the data because a user has a finite amount of time to listen to songs and must pick between them. Also, as discussed in Section 4.3, multinomial deviance implicitly gives higher weight to play counts of 0 coming from highly active users than from the users who had low total play counts. A similar weighting scheme explicitly used below on the binarized data is shown to improve the recommendations.

Another popular option is to binarize the count data (Hu et al., 2008) because often the most important information gained from the play count is whether or not the user listened to a song, not how many times. The simulations in Section 4.3 also show that using the binarized data is preferable when the data are very sparse, as it is here. In Figure 11, we compare AUC using PCA on the binarized data to logistic PCA and the convex relaxation of logistic PCA. We also have a popularity model for reference, which ranks the songs based on the percentage of users that listened to them.

In this scenario, PCA clearly does better than the popularity model with no personalization. PCA on the binary data does about as well as multinomial PCA on the count data. Further, both standard PCA and logistic PCA again peak at about ten principal components, with logistic PCA giving a modest improvement over standard PCA.

The convex relaxation to logistic PCA has the highest overall AUC. In contrast to the others, it peaks at $k = 18$. Even with $k = 18$, however, the rank of the optimal Fantope matrix is 72. This highlights the difference between the convex relaxation and the original

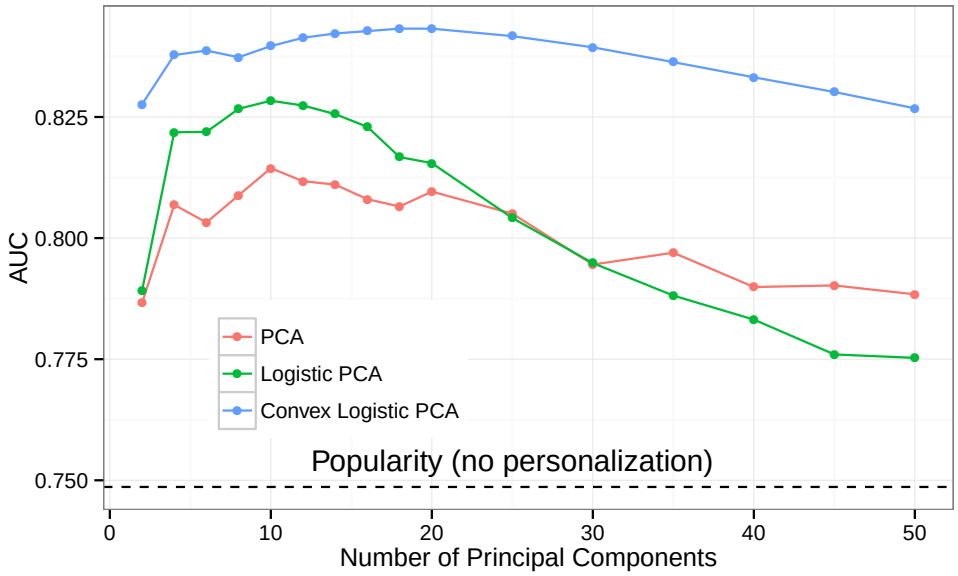

Figure 11: Comparison of different generalized PCA methods for recommending songs to users with the binarized play count data

formulation. The trace k acts as a regularizer on the symmetric matrix $\mathbf{H}$, but does not guarantee a low-rank solution. When a low-rank solution is desired for interpretation, data compression, or plotting, the convex relaxation may not be useful. On the other hand, the use of the Fantope matrix may have computational and performance advantages for prediction as we see in this example.

Finally, using the flexibility of generalized PCA, we incorporate data weights into PCA with the binarized data and examine their effects on AUC. When there are many possible songs to listen to, a user may not listen to a song either because he or she is not aware of the song or because he or she is not interested. To deal with this, Pan et al. (2008) recommended weighing the observations as follows:

$$w_{ij} = \begin{cases} 1 & \text{if } x_{ij} = 1 \\ \alpha \sum_j x_{ij} & \text{if } x_{ij} = 0 \end{cases},$$

where α controls the weight of the zeros compared to the non-zeros.

This weighting scheme is intuitive. If two users have not listened to the same song, it will give a higher weight to that song for the user that listened to many other songs. This is because it is more likely that the user who has listened to many other songs will be aware of the given song and, therefore, the zero count indicates that the user is not interested. The user who has listened to fewer songs, however, may not be aware of the song and the zero should be given a lower weight.

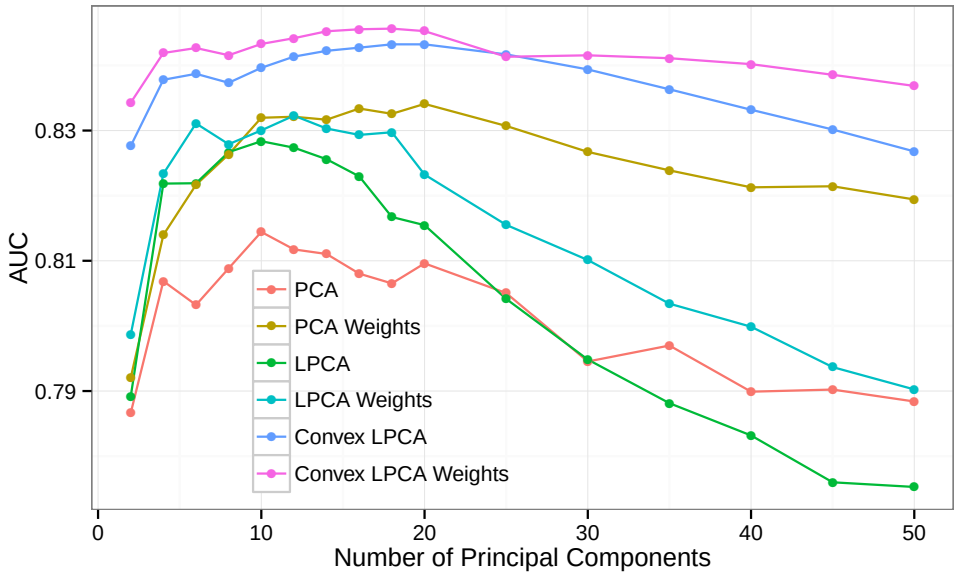

Figure 12: Comparison of different generalized PCA methods for recommending songs to users with the binarized play count data with differential weights

We chose α such that

$$\sum_{(i,j):x_{ij}=0} w_{ij} = \sum_{(i,j):x_{ij}=1} w_{ij},$$

based on the recommendation of Johnson (2014). This advice is based on the heuristic that class balance in classification can improve performance.

The results of the weighted PCA methods are in Figure 12. The weighting scheme improves the AUCs for all three methods, with standard PCA improving the most, even achieving a higher AUC than weighted logistic PCA. The convex relaxation of logistic PCA still has the highest AUC.

6 Conclusion

In this paper, we have introduced a generalization of PCA, focusing on binary and count data, which preserves many of the desirable features of standard PCA in contrast to other generalizations. Notably, principal component scores are a linear combination of the natural parameters from the saturated model, which are transformations of the data, and computing the scores or the low-rank estimate on new data does not require any extra optimization. The formulation is further extended to include weights, missing data, and variable normalization, which can improve the interpretability of the loadings and the quality of the recommendations as we have shown in numerical examples.

There are many ways that the proposed formulation can be extended further. When there are more variables than cases, regularization of the loadings may be helpful to improve estimation of a low dimensional structure or its interpretability. Further, when there is a response variable of interest and dimensionality reduction is considered for covariates as in principal component regression, our formulation can be extended for supervised dimensionality reduction with covariates of various types.

Acknowledgments

This research was supported in part by National Science Foundation grants DMS-12-09194 and DMS-15-13566.

A Appendix

A.1 Derivation of the First-Order Optimality Conditions (2.2)-(2.4)

To arrive at (2.2), we first obtain the gradient of the deviance with respect to $\mathbf{U}$ from the steps below.

$$\begin{aligned}\frac{1}{2}\frac{\partial D}{\partial \mathbf{U}} = &- \frac{\partial}{\partial \mathbf{U}} tr\left(\mathbf{X}^T\left(\mathbf{1}_n\boldsymbol{\mu}^T + (\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{U}\mathbf{U}^T\right)\right) \\ &+ \frac{\partial}{\partial \mathbf{U}}\sum_{i,j} b_j\left(\mu_j + \left[\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})\right]_j\right).\end{aligned}$$

By standard matrix derivative rules (see, for example, Petersen and Pedersen (2012)),

$$\frac{\partial}{\partial \mathbf{U}} tr(\mathbf{X}^T(\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{U}\mathbf{U}^T) = \left(\mathbf{X}^T(\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T) + (\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T)^T\mathbf{X}\right)\mathbf{U}.$$

Letting $\hat{\theta}_{ij} = \mu_j + [\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})]_j$, note that each element in the second term of the gradient with b_j is given by

$$\frac{\partial}{\partial u_{kl}}\sum_{i,j} b_j(\hat{\theta}_{ij}) = \sum_{i,j} b_j'(\hat{\theta}_{ij})\frac{\partial \hat{\theta}_{ij}}{\partial u_{kl}}.$$

Since

$$\frac{\partial[\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})]_j}{\partial u_{kl}} = \begin{cases}(\tilde{\theta}_{ik} - \mu_k)u_{jl} & \text{if } k \neq j \\ (\tilde{\theta}_{ik} - \mu_k)u_{jl} + (\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})^T U_l & \text{if } k = j,\end{cases}$$

we have

$$\begin{aligned}\frac{\partial}{\partial u_{kl}}\sum_{i,j} b_j(\hat{\theta}_{ij}) &= \sum_{i,j} b_j'(\hat{\theta}_{ij})(\tilde{\theta}_{ik}-\mu_k)u_{jl} + \sum_i b_k'(\hat{\theta}_{ik})(\tilde{\boldsymbol{\theta}}_i-\boldsymbol{\mu})^T U_l \\ &= (\tilde{\Theta}_k - \mathbf{1}_n\mu_k)^T b'(\hat{\boldsymbol{\Theta}})U_l + b_k'(\hat{\Theta}_k)^T(\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T)U_l,\end{aligned}$$

where $b'(\hat{\boldsymbol{\Theta}})$ is a matrix with its ijth element $b_j'(\hat{\theta}_{ij})$ and U_l is the lth column of $\mathbf{U}$. In matrix notation,

$$\frac{\partial}{\partial \mathbf{U}}\sum_{i,j} b_j\left(\mu_j + \left[\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i-\boldsymbol{\mu})\right]_j\right) = \left(b'(\hat{\boldsymbol{\Theta}})^T(\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T) + (\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n\boldsymbol{\mu}^T)^T b'(\hat{\boldsymbol{\Theta}})\right)\mathbf{U},$$

and the result in (2.2) follows.

The gradient of the deviance with respect to $\boldsymbol{\mu}$ in (2.3) is derived as follows.

$$\begin{aligned}\frac{1}{2}\frac{\partial D}{\partial \boldsymbol{\mu}} = &-\frac{\partial}{\partial \boldsymbol{\mu}} tr\left(\mathbf{X}^T\mathbf{1}_n\boldsymbol{\mu}^T\left(\mathbf{I}_d - \mathbf{U}\mathbf{U}^T\right)\right) \\ &+ \frac{\partial}{\partial \boldsymbol{\mu}}\sum_{i,j} b_j\left(\mu_j + \left[\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i-\boldsymbol{\mu})\right]_j\right).\end{aligned}$$

Using standard vector differentiation,

$$\frac{\partial}{\partial \boldsymbol{\mu}} tr\left(\mathbf{X}^T\mathbf{1}_n\boldsymbol{\mu}^T\left(\mathbf{I}_d - \mathbf{U}\mathbf{U}^T\right)\right) = (\mathbf{I}_d - \mathbf{U}\mathbf{U}^T)\mathbf{X}^T\mathbf{1}_n$$

and

$$\frac{\partial}{\partial \boldsymbol{\mu}}\sum_{i,j} b_j\left(\mu_j + \left[\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\theta}}_i-\boldsymbol{\mu})\right]_j\right) = \sum_{i,j} b_j'(\hat{\theta}_{ij})\left(\mathbf{e}_j - \mathbf{u}_{j\cdot}^T\mathbf{U}^T\right) = (\mathbf{I}_d - \mathbf{U}\mathbf{U}^T)b'(\hat{\boldsymbol{\Theta}})^T\mathbf{1}_n,$$

where $\mathbf{e}_j$ is a length d standard basis vector with 1 in the jth position and $\mathbf{u}_{j\cdot}$ is the jth row of $\mathbf{U}$.

A.2 Limiting Behavior of the Deviance of the Poisson Distribution with the Deviation Held Constant

Lemma 1. *Let x be an observation from a Poisson distribution with mean parameter λ. If the deviation $\Delta := (x-\lambda)$ is held constant, the deviance $D(x;\lambda)$ monotonically decreases to zero as either λ or x increases to ∞. That is, for fixed Δ*

i. $D(\Delta+\lambda;\lambda)$ decreases in λ, and $\lim_{\lambda\to\infty} D(\Delta+\lambda;\lambda) = 0$, and

ii. $D(x;x-\Delta)$ decreases in x, and $\lim_{x\to\infty} D(x;x-\Delta) = 0$.

Proof. The Poisson deviance is given by $D(x;\lambda) = 2\{x\log(x/\lambda) - (x-\lambda)\}$. When $(x-\lambda)$ is fixed at Δ, the deviance is proportional to $h(\lambda|\Delta) := (\Delta+\lambda)\log[(\Delta+\lambda)/\lambda] - \Delta$. Due to L'Hôpital's rule,

$$\lim_{\lambda\to\infty}(\Delta+\lambda)\log[(\Delta+\lambda)/\lambda] = \lim_{\lambda\to\infty}\frac{\log[(\Delta+\lambda)/\lambda]}{1/\lambda} = \lim_{\lambda\to\infty}\frac{\Delta\lambda^2}{(\Delta+\lambda)\lambda} = \Delta,$$

which proves that $\lim_{\lambda\to\infty} D(\Delta+\lambda;\lambda) = 0$. The derivative of $h(\lambda|\Delta)$ with respect to λ equals $\log[(\Delta+\lambda)/\lambda] - \Delta/\lambda = \log(\Delta/\lambda+1) - \Delta/\lambda$, which is non-positive for all λ $(\geq -\Delta)$ and Δ because $e^z \geq 1+z$ for all z. Hence, the deviance decreases monotonically in λ. The second statement with respect to x can be proved similarly. □

A.3 Approximation to the Saturated Model of the Multinomial Distribution

Lemma 2. *For the approximate saturated model parameters $\tilde{\theta}_{ij}$ defined in* (2.5)*,*

$$\lim_{m\to\infty}\left.\frac{\partial b(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}_i} = \mathbf{x}_i.$$

Proof. The jth partial derivative of $b(\boldsymbol{\theta})$ $(j = 1,\ldots,K-1)$ at $\tilde{\boldsymbol{\theta}}_i$ is

$$\left.\frac{\partial b(\boldsymbol{\theta})}{\partial\theta_j}\right|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}_i} = \frac{\exp(\tilde{\theta}_{ij})}{1+\sum_{l=1}^{K-1}\exp(\tilde{\theta}_{il})}.$$

The values of this partial derivative evaluated at the approximate saturated model parameters from (2.5) and their limits are given in Table 1. As m goes to infinity, the partial derivative converges to x_{ij} in all cases.

Table 1: The value of the jth partial derivative of $b(\boldsymbol{\theta})$ evaluated at the approximate saturated natural parameters for different possible combinations of x_{ij} and x_{iK}. N_0 represents the number of $x_{ij}, j = 1,\ldots,K-1$ that are 0. The last column gives the limit as m goes to infinity.

x_{ij}	x_{iK}	$b'_j(\tilde{\boldsymbol{\theta}}_i)$	$\lim_{m\to\infty} b'_j(\tilde{\boldsymbol{\theta}}_i)$
0	$[0,1]$	$e^{-m}/(1+\sum_{l=1}^{K-1}e^{\tilde{\theta}_{il}})$	0
$(0,1)$	$(0,1)$	$(x_{ij}/x_{iK})/(1/x_{iK}+N_0e^{-m})$	x_{ij}
$(0,1)$	0	$x_{ij}e^m/(1+e^m+N_0e^{-m})$	x_{ij}
1	0	$e^m/(1+e^m+(K-2)e^{-m})$	1

The limit in the first row is true because the denominator is always greater than 1, regardless of the other values of x_{il}. The second row relies on the fact that $1-x_{iK} = \sum_{l=1}^{K-1}x_{il}$ and the third row relies on the fact that $\sum_{l=1}^{K-1}x_{il} = 1$ when $x_{iK} = 0$. The fourth row is straightforward. □

A.4 Proof of the Minimizers of the Majorizing Function

Holding $\boldsymbol{\mu}$ constant, we verify that

$$\begin{aligned}
&\arg\min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}_k} M(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)}) = \arg\min \left\| \left(\mathbf{V}^{(t)}\right)^{1/2} \tilde{\boldsymbol{\Theta}}_c^{(t)} \mathbf{U}\mathbf{U}^T - \left(\mathbf{V}^{(t)}\right)^{1/2} \mathbf{Z}_c^{(t)} \right\|_F^2 \\
&= \arg\min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}_k} tr\left(\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\Theta}}_c^{(t)})^T\mathbf{V}^{(t)}\tilde{\boldsymbol{\Theta}}_c^{(t)}\mathbf{U}\mathbf{U}^T\right) - tr\left(\mathbf{U}\mathbf{U}^T(\tilde{\boldsymbol{\Theta}}_c^{(t)})^T\mathbf{V}^{(t)}\mathbf{Z}_c^{(t)}\right) - tr\left((\mathbf{Z}_c^{(t)})^T\mathbf{V}^{(t)}\tilde{\boldsymbol{\Theta}}_c^{(t)}\mathbf{U}\mathbf{U}^T\right) \\
&= \arg\min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}_k} tr\left[\mathbf{U}^T\left((\tilde{\boldsymbol{\Theta}}_c^{(t)})^T\mathbf{V}^{(t)}\tilde{\boldsymbol{\Theta}}_c^{(t)} - (\tilde{\boldsymbol{\Theta}}_c^{(t)})^T\mathbf{V}^{(t)}\mathbf{Z}_c^{(t)} - (\mathbf{Z}_c^{(t)})^T\mathbf{V}^{(t)}\tilde{\boldsymbol{\Theta}}_c^{(t)}\right)\mathbf{U}\right] \\
&= \arg\max_{\mathbf{U}^T\mathbf{U}=\mathbf{I}_k} tr\left[\mathbf{U}^T\left(\left(\tilde{\boldsymbol{\Theta}}_c^{(t)}\right)^T\mathbf{V}^{(t)}\mathbf{Z}_c^{(t)} + \left(\mathbf{Z}_c^{(t)}\right)^T\mathbf{V}^{(t)}\tilde{\boldsymbol{\Theta}}_c^{(t)} - \left(\tilde{\boldsymbol{\Theta}}_c^{(t)}\right)^T\mathbf{V}^{(t)}\tilde{\boldsymbol{\Theta}}_c^{(t)}\right)\mathbf{U}\right].
\end{aligned}$$

The trace in the last line is maximized by the first k eigenvectors of $\left(\tilde{\boldsymbol{\Theta}}_c^{(t)}\right)^T\mathbf{V}^{(t)}\mathbf{Z}_c^{(t)} + \left(\mathbf{Z}_c^{(t)}\right)^T\mathbf{V}^{(t)}\tilde{\boldsymbol{\Theta}}_c^{(t)} - \left(\tilde{\boldsymbol{\Theta}}_c^{(t)}\right)^T\mathbf{V}^{(t)}\tilde{\boldsymbol{\Theta}}_c^{(t)}$, as shown in the famous result of Fan (1949).

Given $\mathbf{U}^{(t)}$, minimizing $M(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)})$ with respect to $\boldsymbol{\mu}$ is equivalent to minimizing

$$\begin{aligned}
&tr\left[\left(\mathbf{I}-\mathbf{U}^{(t)}(\mathbf{U}^{(t)})^T\right)\boldsymbol{\mu}\left(\mathbf{1}_n^T\mathbf{V}^{(t)}\mathbf{1}_n\right)\boldsymbol{\mu}^T\left(\mathbf{I}-\mathbf{U}^{(t)}(\mathbf{U}^{(t)})^T\right)\right] - \\
&2tr\left[\left(\mathbf{Z}^{(t)}-\tilde{\boldsymbol{\Theta}}\mathbf{U}^{(t)}(\mathbf{U}^{(t)})^T\right)^T\mathbf{V}^{(t)}\mathbf{1}_n\boldsymbol{\mu}^T\left(\mathbf{I}-\mathbf{U}^{(t)}(\mathbf{U}^{(t)})^T\right)\right].
\end{aligned}$$

The minimizer $\boldsymbol{\mu}$ can be found by differentiating $M(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)})$ with respect to $\boldsymbol{\mu}$ and setting the gradient equal to zero, which leads to

$$\left(\mathbf{I}-\mathbf{U}^{(t)}(\mathbf{U}^{(t)})^T\right)\boldsymbol{\mu}\left(\mathbf{1}_n^T\mathbf{V}^{(t)}\mathbf{1}_n\right) = \left(\mathbf{I}-\mathbf{U}^{(t)}(\mathbf{U}^{(t)})^T\right)\left(\mathbf{Z}^{(t)}-\tilde{\boldsymbol{\Theta}}\mathbf{U}^{(t)}(\mathbf{U}^{(t)})^T\right)^T\mathbf{V}^{(t)}\mathbf{1}_n.$$

The update rule produces $\boldsymbol{\mu}$ that satisfies the equation at each step since $\mathbf{I}-\mathbf{U}^{(t)}(\mathbf{U}^{(t)})^T$ is a projection matrix and $\mathbf{1}_n^T\mathbf{V}^{(t)}\mathbf{1}_n$ is a scalar.

A.5 Proof of (3.4)

The result in (3.4) follows closely from the derivation of (2.2). The weighted deviance can be written as

$$-2tr\left((\mathbf{W}\circ\mathbf{X})^T\left(\mathbf{1}_n\boldsymbol{\mu}^T + (\tilde{\boldsymbol{\Theta}}-\mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{H}\right)\right) + 2\sum_{i,j} w_{ij}b_j\left(\mu_j + \left[\mathbf{H}(\tilde{\boldsymbol{\theta}}_i-\boldsymbol{\mu})\right]_j\right).$$

The gradient of the first piece is equal to

$$\frac{\partial}{\partial\mathbf{H}}tr((\mathbf{W}\circ\mathbf{X})^T(\tilde{\boldsymbol{\Theta}}-\mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{H}) = \frac{\partial}{\partial\mathbf{H}}tr((\mathbf{W}\circ\mathbf{X})^T(\tilde{\boldsymbol{\Theta}}-\mathbf{1}_n\boldsymbol{\mu}^T)\mathbf{H}^T) = (\mathbf{W}\circ\mathbf{X})^T(\tilde{\boldsymbol{\Theta}}-\mathbf{1}_n\boldsymbol{\mu}^T),$$

where we transposed $\mathbf{H}$ in the first step to make the end result more concise.

As in Appendix A.1,

$$\begin{aligned}\frac{\partial}{\partial h_{kl}}\sum_{i,j} w_{ij} b_j(\hat{\theta}_{ij}) &= \sum_{i,j} w_{ij} b_j'(\hat{\theta}_{ij})\frac{\partial \hat{\theta}_{ij}}{\partial h_{kl}} \\ &= \sum_{i} w_{ik} b_k'(\hat{\theta}_{ik})(\tilde{\theta}_{il} - \mu_l)^T \\ &= (W_{\cdot k} \circ b_k'(\hat{\Theta}_k))^T(\tilde{\Theta}_l - \mathbf{1}_n \mu_l).\end{aligned}$$

In matrix notation,

$$\frac{\partial}{\partial \mathbf{H}}\sum_{i,j} w_{ij} b_j\left(\mu_j + \left[\mathbf{H}(\tilde{\boldsymbol{\theta}}_i - \boldsymbol{\mu})\right]_j\right) = (\mathbf{W} \circ b'(\hat{\boldsymbol{\Theta}}))^T(\tilde{\boldsymbol{\Theta}} - \mathbf{1}_n \boldsymbol{\mu}^T),$$

and the result in (3.4) follows.

References

Bertin-Mahieux, T., D. P. Ellis, B. Whitman, and P. Lamere (2011). The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research 3*, 993–1022.

Boyd, S., L. Xiao, and A. Mutapcic (2003). Subgradient methods. *Lecture notes of EE392o, Stanford University, Autumn Quarter 2004*.

Buntine, W. (2002). Variational extensions to EM and multinomial PCA. In *European Conference on Machine Learning*, pp. 23–34. Springer.

Collins, M., S. Dasgupta, and R. E. Schapire (2001). A generalization of principal components analysis to the exponential family. In T. Dietterich, S. Becker, and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14*, pp. 617–624.

Dattorro, J. (2005). *Convex optimization & Euclidean distance geometry*. Meboo Publishing USA.

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

Fan, K. (1949). On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences of the United States of America 35*(11), 652.

Feuerverger, A., Y. He, and S. Khatri (2012). Statistical significance of the Netflix challenge. *Statistical Science 27*(2), 202–231.

Gordon, G. J. (2002). Generalized^2 linear^2 models. In S. Becker, S. Thrun, and K. Obermayer (Eds.), *Advances in Neural Information Processing Systems 15*, pp. 593–600. MIT Press.

Greenacre, M. J. (1984). *Theory and applications of correspondence analysis*. Academic Press.

Hu, Y., Y. Koren, and C. Volinsky (2008). Collaborative filtering for implicit feedback datasets. In *Eighth IEEE International Conference on Data Mining (ICDM), 2008*, pp. 263–272. IEEE.

Hunter, D. R. and K. Lange (2004). A tutorial on MM algorithms. *The American Statistician 58*(1), 30–37.

Johnson, C. C. (2014). Logistic matrix factorization for implicit feedback data. In *Advances in Neural Information Processing Systems 27: Distributed Machine Learning and Matrix Computations Workshop*.

Landgraf, A. J. and Y. Lee (2015). Dimensionality reduction for binary data through the projection of natural parameters. Technical Report 890, Department of Statistics, The Ohio State University.

Lee, D. D. and H. S. Seung (1999). Learning the parts of objects by non-negative matrix factorization. *Nature 401*(6755), 788–791.

Lee, S., J. Z. Huang, and J. Hu (2010). Sparse logistic principal components analysis for binary data. *The Annals of Applied Statistics 4*(3), 1579–1601.

Mazumder, R., T. Hastie, and R. Tibshirani (2010). Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research 11*, 2287–2322.

Pan, R., Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang (2008). One-class collaborative filtering. In *Eighth IEEE International Conference on Data Mining (ICDM), 2008*, pp. 502–511. IEEE.

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 2*(11), 559–572.

Petersen, K. B. and M. S. Pedersen (2012, November). The matrix cookbook. Version 20121115.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Schein, A. I., L. K. Saul, and L. H. Ungar (2003). A generalized linear model for principal component analysis of binary data. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Volume 38.

Singh, A. P. and G. J. Gordon (2008). A unified view of matrix factorization models. In *Machine Learning and Knowledge Discovery in Databases*, pp. 358–373. Springer.

Tipping, M. E. (1998). Probabilistic visualisation of high-dimensional binary data. In M. Kearns, S. Solla, and D. Cohn (Eds.), *Advances in Neural Information Processing Systems 11*, pp. 592–598.

Tipping, M. E. and C. M. Bishop (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 61*(3), 611–622.

Udell, M., C. Horn, R. Zadeh, and S. Boyd (2014). Generalized low rank models. *arXiv preprint arXiv:1410.0342*.

Vu, V. Q., J. Cho, J. Lei, and K. Rohe (2013). Fantope projection and selection: A near-optimal convex relaxation of sparse PCA. In *Advances in Neural Information Processing Systems*, pp. 2670–2678.

Wen, Z. and W. Yin (2013). A feasible method for optimization with orthogonality constraints. *Mathematical Programming 142*(1-2), 397–434.