A Generalized Linear Model for Principal Component Analysis of Binary Data

Andrew I. Schein   Lawrence K. Saul   Lyle H. Ungar
Department of Computer and Information Science
University of Pennsylvania
Moore School Building
200 South 33rd Street
Philadelphia, PA 19104-6389
{ais,lsaul,ungar}@cis.upenn.edu

Abstract

We investigate a generalized linear model for dimensionality reduction of binary data. The model is related to principal component analysis (PCA) in the same way that logistic regression is related to linear regression. Thus we refer to the model as logistic PCA. In this paper, we derive an alternating least squares method to estimate the basis vectors and generalized linear coefficients of the logistic PCA model. The resulting updates have a simple closed form and are guaranteed at each iteration to improve the model's likelihood. We evaluate the performance of logistic PCA—as measured by reconstruction error rates—on data sets drawn from four real world applications. In general, we find that logistic PCA is much better suited to modeling binary data than conventional PCA.

1 Introduction

Principal component analysis (PCA) is a canonical and widely used method for dimensionality reduction of multivariate data. Applications include the exploratory analysis[9] and visualization of large data sets, as well as the denoising and decorrelation of inputs for algorithms in statistical learning[2, 6]. PCA discovers the linear projections of the data with maximum variance, or equivalently, the lower dimensional subspace that yields the minimum squared reconstruction error. In practice, model fitting occurs either by computing the top eigenvectors of the sample covariance matrix or by performing a singular value decomposition on the matrix of mean-centered data.

While the centering operations and least squares criteria of PCA are naturally suited to real-valued data, they are not generally appropriate for other data types. Recently, Collins et al.[5] derived generalized criteria for dimensionality reduction by appealing to properties of distributions in the exponential family. In their framework, the conventional PCA of real-valued data emerges naturally from assuming a Gaussian distribution over a set of observations, while generalized versions of PCA for binary and nonnegative data emerge respectively by substituting the Bernoulli and Poisson distributions for the Gaussian. For binary data, the generalized model's relationship to PCA is analogous to the relationship between logistic and linear regression[12]. In particular, the model exploits the log-odds as the natural parameter of the Bernoulli distribution and the logistic function as its canonical link. In this paper we will refer to the PCA model for binary data as logistic PCA, and to its counterpart for real-valued data as linear (or conventional) PCA.

Collins et al.[5] proposed an iterative algorithm for all of these generalizations of PCA, but the optimizations required at each iteration of their algorithm do not have a simple closed form for the logistic PCA case. In this paper, we derive an alternating least squares method to estimate the basis vectors and generalized linear coefficients of logistic PCA. Our method adapts an algorithm, originally proposed by Tipping[15], for fitting the parameters of a closely related generative model. Like Tipping's algorithm, it also relies on a particular convexity inequality introduced by Jaakkola and Jordan[10] for the logistic function. Our algorithm is easy to implement and guaranteed at each iteration to improve the logistic PCA log-likelihood. Thus, it provides a simple but powerful tool for dimensionality reduction of binary data.

We evaluate the performance of our algorithm on data sets drawn from four real world applications. Performance is measured by the ability to compress and reconstruct large binary matrices with minimum error.

Our experimental results add significantly to those of Collins et al[5], who looked only at simulated data, and Tipping[15], who mainly considered the application to visualization. In the majority of experiments, we find that logistic PCA is much better suited to binary data than conventional PCA.

2 Model

Logistic PCA is based on a multivariate generalization of the Bernoulli distribution. The Bernoulli distribution for a univariate binary random variable $x \in \{0, 1\}$ with mean p is given by:

$$\mathsf{P}(x|p) = p^x (1-p)^{1-x}. \tag{1}$$

We can equivalently write this distribution in terms of the log-odds parameter $\theta = \log(\frac{p}{1-p})$ and the logistic function $\sigma(\theta) = [1 + e^{-\theta}]^{-1}$. In these terms, the Bernoulli distribution is given by:

$$\mathsf{P}(x|\theta) = \sigma(\theta)^x \sigma(-\theta)^{1-x}. \tag{2}$$

The log-odds and logistic function are, respectively, the natural parameter and canonical link function of the Bernoulli distribution expressed as a member of the exponential family.

A simple multivariate generalization of eq. (2) yields the logistic PCA model. Let X_{nd} denote the elements of an $N \times D$ binary matrix, each of whose N rows stores the observation of a D-dimensional binary vector. A probability distribution over matrices of this form is given by:

$$\mathsf{P}(X|\Theta) = \prod_{nd} \sigma(\Theta_{nd})^{X_{nd}} \sigma(-\Theta_{nd})^{1-X_{nd}}, \tag{3}$$

where Θ_{nd} denotes the log-odds of the binary random variable X_{nd}. The log-likelihood of binary data under this model is given by:

$$\mathcal{L} = \sum_{nd} \left[X_{nd} \log \sigma(\Theta_{nd}) + (1 - X_{nd}) \log \sigma(-\Theta_{nd}) \right]. \tag{4}$$

Low dimensional structure in the data can be discovered by assuming a compact representation for the log-odds matrix Θ and attempting to maximize this log-likelihood. A compact representation analogous to PCA is obtained by constraining the rows of Θ to lie in a latent subspace of dimensionality $L \ll D$. To this end, we parameterize the log-odds matrix Θ in terms of two smaller matrices U and V and (optionally) a bias vector Δ. In terms of these parameters, the $N \times D$ matrix Θ is represented as:

$$\Theta_{nd} = \sum_{\ell} U_{n\ell} V_{\ell d} + \Delta_d, \tag{5}$$

where U is an equally tall but narrower $N \times L$ matrix, V is a shorter but equally wide $L \times D$ matrix, Δ is a D-dimensional vector, and the sum over the subscript ℓ in eq. (5) makes explicit the matrix multiplication of U and V. Note that the parameters U, V and Δ in this model play roles analogous to the linear coefficients, basis vectors, and empirical mean computed by PCA of real-valued data. The model is summarized in Table 1. Though the bias vector Δ in this model could be absorbed by a redefinition of U and V, its presence permits a more straightforward comparison to linear PCA of mean-centered data.

Table 1: Summary of the logistic PCA model notation.

N	number of observations
D	dimensionality of binary data
L	dimensionality of latent space

X_{nd}	binary data	($N \times D$ matrix)
Θ_{nd}	log-odds	($N \times D$ matrix)
$\mathsf{P}[X_{nd} = 1 \vert \Theta_{nd}] = \sigma(\Theta_{nd})$		

$U_{n\ell}$	coefficients	($N \times L$ matrix)
$V_{\ell d}$	basis vectors	($L \times D$ matrix)
Δ_d	bias vector	($1 \times D$ vector)
$\Theta_{nd} = (UV)_{nd} + \Delta_d$		

Logistic PCA can be applied to binary data in largely the same way that conventional (or linear) PCA is applied to real-valued data. In particular, given binary data X, we compute the parameters U, V and Δ that maximize (at least locally) the log-likelihood in eq. (4). An iterative least squares method for maximizing (4) is described in the next section. While the log-likelihood in eq. (4) is not convex in the parameters U, V, and Δ, it is convex in any one of these parameters if the others are held fixed. Thus, having estimated these parameters from training data X, we can compute a low dimensional representation U' of previously unseen (or test) data X' by locating the global maximum of the corresponding test log-likelihood $\mathcal{L}'$ (with fixed V and Δ).

Logistic and linear PCA can both be viewed as special cases of the generalized framework described by Collins et al[5]. This is done by writing the log-likelihood in eq. (4) in the more general form:

$$\mathcal{L} = \sum_{nd} \Theta_{nd} X_{nd} - G(\Theta_{nd}) + \log P_0(X_{nd}), \tag{6}$$

where it applies to any member distribution of the exponential family. In this more general formulation, the function $G(\theta)$ in eq. (6) is given by the integral of the distribution's canonical link, while the term $P_0(x)$ provides a normalization but otherwise has no dependence on the distribution's natural parameter. As before, the rows of Θ are constrained to lie in a low dimensional subspace. Linear PCA for real-valued data emerges naturally in this framework from the form of the Gaussian distribution[17, 18], while logistic PCA emerges from the form of the Bernoulli distribution. Other examples and a fuller treatment are given by Collins et al[5].

Note that logistic PCA has many of the same shortcomings as linear PCA. In particular, it does not define a proper generative model that can be used to handle missing data or infer a conditional distribution $\mathsf{P}[U|X]$ over the coefficients U given data X. Factor analysis[1, 15] and related probabilistic models[14, 16] exist to address these shortcomings of linear PCA. Generalized models of factor analysis have also been proposed for binary data[1, 15]. These models typically assume that the coefficients U obey a multivariate Gaussian distribution. In such models, however, exact probabilistic inference is intractable, and approximations are required to compute the expected values for maximum likelihood estimation. Logistic PCA can be viewed as a simpler non-probabilistic alternative to these models that does not make an explicit assumption about the distribution obeyed by the coefficients U.

3 Algorithm

Logistic PCA models the structure of binary data by maximizing the log-likelihood in eq. (4), subject to the constraint that the rows of the log-odds matrix Θ lie in the linear subspace defined by the parameters U, V and Δ in eq. (5). Our method for fitting these parameters is an iterative scheme that alternates between least squares updates for U, V and Δ. Essentially, one set of parameters is updated while the others are held fixed, and this procedure is repeated—cycling through U-updates, V-updates and Δ-updates—until the log-likelihood converges to a desired degree of precision. In the appendix, an auxiliary function is used to derive the alternating least squares (ALS) updates and to prove that they lead to monotonic increases in the log-likelihood. In this section, we present the ALS updates with a minimum of extra formalism, noting similarities and differences with other approaches.

U-Update:
Holding the parameters V and Δ fixed, we obtain a simple update rule for the matrix U that stores the reconstruction coefficients of the log-odds matrix Θ. We begin by computing intermediate quantities:

$$\begin{aligned} T_{nd} &= \frac{\tanh(\Theta_{nd}/2)}{\Theta_{nd}}, & (7)\\ A_{n\ell\ell'} &= \sum_d T_{nd} V_{\ell d} V_{\ell' d}, & (8)\\ B_{n\ell} &= \sum_d \left[2X_{nd} - 1 - T_{nd}\Delta_d\right] V_{\ell d}. & (9)\end{aligned}$$

Note that T_{nd} depends on Θ_{nd} and (hence) the current estimate of U through eq. (5). In terms of the matrices in eqs. (7–9), the updates for different rows of the matrix U are conveniently decoupled. In particular, we update the nth row of the matrix U by solving the $L \times L$ set of linear equations:

$$\sum_{\ell'} A_{n\ell\ell'} U_{n\ell'} = B_{n\ell}. \qquad (10)$$

V-Update
Holding the parameters U and Δ fixed, we obtain a similar update rule for the matrix V that stores the basis vectors of the log-odds matrix Θ. Again, we compute the matrix T_{nd} as in eq. (7), as well as the intermediate quantities:

$$\begin{aligned} \mathcal{A}_{d\ell\ell'} &= \sum_n T_{nd} U_{n\ell} U_{n\ell'}, & (11)\\ \mathcal{B}_{d\ell} &= \sum_n \left[2X_{nd} - 1 - T_{nd}\Delta_d\right] U_{n\ell}. & (12)\end{aligned}$$

In terms of these quantities, the updates for different *columns* of the matrix V are conveniently decoupled. In particular, we update the dth column of the matrix V by solving the $L \times L$ set of linear equations:

$$\sum_{\ell'} \mathcal{A}_{d\ell\ell'} V_{\ell' d} = \mathcal{B}_{d\ell}. \qquad (13)$$

Δ-Update
Finally, holding the parameters U and V fixed, we obtain a simple update rule for the bias vector Δ. With T_{nd} computed as in eq. (7), we update the elements of the bias vector by:

$$\Delta_d = \frac{\sum_n \left[2X_{nd} - 1 - T_{nd}(UV)_{nd}\right]}{\sum_{n'} T_{n'd}}. \qquad (14)$$

To compute the coefficients U' for test data X' from a previously trained model of logistic PCA, one iterates just the U-updates while holding the parameters V and Δ fixed. This is done until the log-likelihood for X', based on the coefficients U', converges to a desired degree of accuracy. The optimization of U' for fixed V and Δ is convex, so that for test data X', the U-updates converge (except in degenerate cases) to the same result independent of the starting point for U'.

The ALS updates have a simple closed form that make them easier to implement than completely general algorithms for convex optimization. Thus, for dimensionality reduction of binary data, they provide a simpler approach than the general procedure outlined in Collins et al.[5] for models of the form in eq. (6). The ALS updates are closely related to Tipping's procedure[15] for fitting a factor analysis model of binary data[1], though in the factor analysis model, they are used to approximate a posterior *distribution* over the matrix U. The model of dimensionality reduction in this paper is simpler by comparison, as it requires only point estimates of U.

4 Experiments

We compared the performance of logistic and linear PCA on data sets of varying size, dimensionality and sparsity. To ensure a high degree of convergence we employed 300 iterations of ALS to fit the logistic PCA model. A singular value decomposition of mean-centered data served to fit the linear PCA model. For latent spaces of equal dimensionality, these two models involve the same number of fitted parameters, making a direct comparison fairly straightforward. We employed latent spaces of dimensionality $L = 1, 2, 4$ and 8 in our experiments.

We evaluated logistic and linear PCA by measuring how well their low dimensional representations could reconstruct large matrices of binary data from four real world application domains. Note that the low dimensional representations of both models yield continuous real-valued predictions for these reconstructions; these were converted to binary values $\{0, 1\}$ by simple thresholding. For each data set and for each hypothesized dimensionality L of the latent space, we report results from logistic and linear PCA in terms of two error rates: a minimum overall error rate and a balanced error rate. These error rates were obtained by choosing the models' binary decision thresholds in two different ways. In the first case, the thresholds were chosen to minimize the overall error rates, counting equally those errors due to false positives and false negatives. In the second case, the thresholds were chosen to equalize the false positive and false negative error rates; this has the effect, in sparse data sets, of giving more weight to errors from false positives. Both types of error rates are revealing, as they capture different notions of error cost. The results are summarized in Table 2. Note that logistic PCA outperformed linear PCA on the task of binary data reconstruction in nearly every one of our experiments, with exceptional performance gains under the balanced error rate criterion.

The individual data sets are described in greater detail below. The density of each data set was measured by the mean value of elements in the $N \times D$ binary data matrix X and is indicated in Table 2 by the symbol ρ.

4.1 Microsoft Web Data

The anonymous Microsoft Web Database is available from the UCI machine learning repository[1]. It has been used previously to evaluate collaborative filtering algorithms[3]. The data was generated by sampling logs at `www.microsoft.com` that recorded (anonymously) the behavior of $N = 32711$ users. The extracted portion of the logs consists of binary variables indicating whether a user visited a URL over a one week period. There are data for $D = 285$ "vroots" or URL prefixes. In our experiments, each user was represented by a row in the binary data matrix X. The data is very sparse, with density $\rho = 0.011$.

4.2 MovieLens Data

The MovieLens data set contains movie ratings by a large number of users[7]. It was collected in order to evaluate the performance of automated recommender systems. In our experiments we ignore the actual ratings except to obtain a binary matrix indicating which of $D = 1682$ movies were rated by $N = 993$ users. Each user was represented by a row in the binary data matrix X. The data is fairly sparse, with density $\rho = 0.063$.

4.3 Gene Expression Data

The gene expression data of Causton et al.[4] was generated by measuring gene expression levels on an Affymetrix[2] gene chip. The experimental design called for genome-wide expression analysis of *Saccharomyces cerevisiae* to detect transcriptional change under various environmental conditions. The measurements were converted to binary values by Affymetrix GeneChip software and provided in this form to the authors. The binary values are noisy indicators of the presence or absence of mRNA in a *Saccharomyces cerevisiae* cell. There are measurements for $N = 6015$ genes in $D = 46$ environmental conditions. Each gene was represented by a row in the binary data matrix X. The data is fairly balanced between positive and negative indicators, with density $\rho = 0.738$.

4.4 Advertisement Data

The advertisement data is available from the UCI machine learning repository. The data was collected

[1] `http://www.ics.uci.edu/~mlearn/MLRepository.html`

[2] Affymetrix, Inc. Santa Clara, CA

Table 2: Minimum and balanced error rates on four different data sets for the task of binary data reconstruction. The data sets were $N \times D$ binary matrices with ρND nonzero elements.

Microsoft Web Log ($N = 32711$, $D = 285$, $\rho = 0.011$)

Minimum Error Rates (%)

L	Linear PCA	Logistic PCA
1	0.0886	0.0959
2	0.0831	0.0701
4	0.0661	0.0502
8	0.0475	0.0237

Balanced Error Rates (%)

L	Linear PCA	Logistic PCA
1	1.52	1.28
2	1.41	1.15
4	1.36	0.760
8	1.11	0.355

MovieLens ($N = 943$, $D = 1682$, $\rho = 0.063$)

Minimum Error Rates (%)

L	Linear PCA	Logistic PCA
1	0.557	0.555
2	0.528	0.518
4	0.498	0.479
8	0.457	0.428

Balanced Error Rates (%)

L	Linear PCA	Logistic PCA
1	1.73	1.64
2	1.57	1.42
4	1.34	1.18
8	1.15	0.993

Gene Expression ($N = 6015$, $D = 46$, $\rho = 0.738$)

Minimum Error Rates (%)

L	Linear PCA	Logistic PCA
1	1.24	1.21
2	1.04	1.02
4	0.884	0.758
8	0.667	0.393

Balanced Error Rates (%)

L	Linear PCA	Logistic PCA
1	1.46	1.43
2	1.28	1.20
4	1.03	0.804
8	0.766	0.499

Advertising ($N = 3279$, $D = 1555$, $\rho = 0.072$)

Minimum Error Rates (%)

L	Linear PCA	Logistic PCA
1	0.0709	0.700
2	0.0682	0.0591
4	0.0616	0.0388
8	0.0576	0.0124

Balanced Error Rates (%)

L	Linear PCA	Logistic PCA
1	2.68	1.97
2	2.39	1.20
4	2.17	0.626
8	1.76	0.268

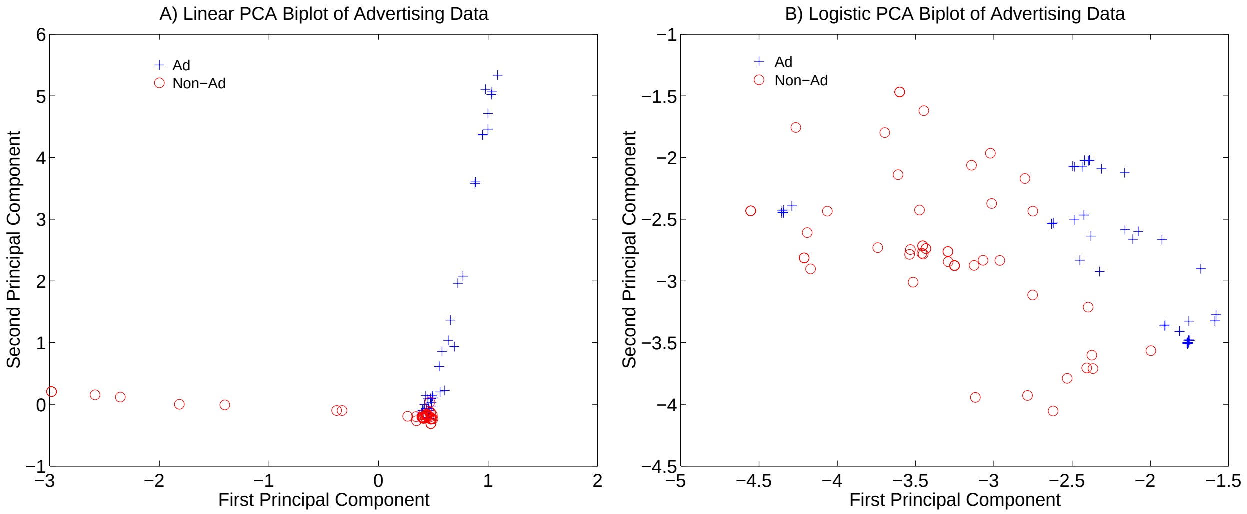

Figure 1: Separation of advertisements and non-advertisements by the first two components of (A) linear PCA and (B) logistic PCA applied to their surrounding features.

to predict whether or not images are advertisements, based on a large number of their surrounding features. To generate binary data for our experiments, we removed the three continuous features in this data set, as well as the one feature with missing values. We also removed the class labels distinguishing advertisements from non-advertisements. The data for our experiments consisted of $D = 1555$ binary features for $N = 3279$ images. Each image was represented by a row in the binary data matrix X. The data is very sparse, with density $\rho = 0.072$.

Besides the error rates in Table 2, for this data we also plotted the first two principal components for the first fifty points of each class (advertisement and non-advertisement). The results in Fig. 1 show that both linear and logistic PCA separate the classes with only a moderate number of outliers. Curiously, though, the data is much more uniformly distributed in the plot for logistic PCA. These results suggest, as in previous studies[15], that logistic PCA may be useful for continuous visualization of multivariate binary data.

5 Discussion

Our experimental results show conclusively that logistic PCA is better suited to reconstruction of binary data than linear PCA. They also establish the practical utility of the ALS updates for optimizing the log-likelihood in eq. (4). The results are noteworthy in the absence of any theoretical guarantee that the ALS updates converge to a *global* maximum of the log-likelihood. It is not currently known whether this log-likelihood has local maxima to confound optimal model fitting. Our experimental results leave room for optimism that this is not the case. Currently we know that the maximum likelihood value of the logistic PCA model will not have unique estimates for U and V as we can always permute these matrices.

There are many promising areas for future work. In many applications involving binary data, dimensionality reduction has been performed by singular value decomposition (SVD). For these applications, logistic PCA provides a natural and compelling alternative. It will also be useful to develop algorithms for dimensionality reduction of hybrid data (c.f. [1]); this could be done by combining ALS methods for binary features and SVD methods for real-valued ones. Empirical comparisons should also be made between logistic PCA and the models of nonnegative matrix factorization[11] and probabilistic latent semantic analysis[8]. These models, unlike conventional PCA, are tailored to the dimensionality reduction of nonnegative and stochastic matrices. Difficulties have been reported when some of these models were used to fit extremely sparse data from word document counts[13]. We have not encountered such difficulties in our investigations of logistic PCA. Finally, we need to develop a better understanding of the relationship between logistic PCA and factor analysis models of binary data[1, 15]. This relationship is not as well understood as the relationship between non-probabilistic and probabilistic models of PCA[14, 16] for real-valued data.

Acknowledgments

Andrew Schein was supported by NIH Training Grant in Computational Genomics, T-32-HG00046.

References

[1] D. J. Bartholomew and M. Knott. *Latent Variable Models and Factor Analysis*, volume 7 of *Kendall's Library of Satistics*. Oxford University Press, New York, 1999.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[3] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Fourteenth Conference on Uncertainty in Artificial Intelligence*, November, 1998.

[4] H. C. Causton, B. Ren, S. S. Koh, C. T. Harbison, E. Kanin, E. G. Jennings, T. I. Lee, H. L. True, E. S. Lander, and R. A. Young. Remodeling of yeast genome expression in response to environmental changes. *Molecular Biology of the Cell*, 12:323–337, 2001.

[5] M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal component analysis to the exponential family. In *Proceedings of NIPS*, 2001.

[6] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer-Verlag, 2001.

[7] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, August, 1999.

[8] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001.

[9] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[10] T. Jaakkola and M. Jordan. A variational approach to bayesian logistic regression models and their extensions. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, 1997.

[11] D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

[12] P. McCullagh and J. A. Nelder. *Generalised Linear Models.* Chapman and Hall, London, 1987.

[13] A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 17'th Conference on Uncertainty in Artificial Intelligence (UAI 2001), Seatle, WA, August*, 2001.

[14] S. Roweis. EM algorithms for PCA and SPCA. In M. I. Jordan, M. S. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 626–632, Cambridge, MA, 1998. MIT Press.

[15] M. E. Tipping. Probabilistic visualisation of high-dimensional binary data. In *Advances in Neural Information Processing Systems 11*, pages 592–598. MIT Press, 1999.

[16] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, 61:661–622, 1999.

[17] P. Whittle. On principal components and least squares method of factor analysis. *Skandinavisk Aktuarietidskrift*, 36:223–239, 1952.

[18] G. Young. Maximum likelihood estimation and factor analysis. *Psychometrika*, 6:49–53, 1940.

A Derivation

The ALS updates are derived from a lower bound on the log-likelihood $\mathcal{L}(\Theta)$ in eq. (4). To establish the bound, we begin by noting that the sigmoid function in eq. (4) obeys:

$$\log\sigma(\theta) = -\log 2 + (\theta/2) - \log\cosh(\theta/2). \tag{15}$$

Consider the rightmost term in this equation. A bound on this term is obtained by noting that $\log\cosh(\sqrt{z})$ is a concave function of z. It follows that the value of this function at $\hat{z}$ is upper bounded by its linear extrapolation from z:

$$\log\cosh(\sqrt{\hat{z}}) \leq \log\cosh(\sqrt{z}) + (\hat{z}-z)\left[\frac{\tanh(\sqrt{z})}{2\sqrt{z}}\right]. \tag{16}$$

This bound was introduced by Jaakkola and Jordan[10] for Bayesian logistic regression and subsequently applied to factor analysis of binary data by Tipping[15]. Substituting $\sqrt{z} = \frac{\theta}{2}$ and $\sqrt{\hat{z}} = \frac{\hat{\theta}}{2}$ into eq. (16) gives:

$$\log\cosh(\hat{\theta}/2) \leq \log\cosh(\theta/2) + (\hat{\theta}^2 - \theta^2)\left[\frac{\tanh(\theta/2)}{4\theta}\right]. \tag{17}$$

Finally, combining eqs. (4) and (15–17), we obtain a lower bound on the log-likelihood:

$$\mathcal{L}(\hat{\Theta}) \geq \sum_{nd} \Bigg[\log 2 - \log\cosh(\Theta_{nd}/2) + \frac{T_{nd}\Theta_{nd}^2}{4} + \frac{(2X_{nd}-1)\hat{\Theta}_{nd}}{2} - \frac{T_{nd}\hat{\Theta}_{nd}^2}{4} \Bigg], \qquad (18)$$

with T_{nd} defined as in eq. (7). Note that this bound on $\mathcal{L}(\hat{\Theta})$ is quadratic in the matrix elements of $\hat{\Theta}$ and holds for all matrices $\hat{\Theta}$ and Θ.

Let the auxiliary function $Q(\hat{\Theta}, \Theta)$ be defined by the right hand side of eq. (18). The auxiliary function satisfies the key property that $\mathcal{L}(\hat{\Theta}) \geq Q(\hat{\Theta}, \Theta)$ for all matrices $\hat{\Theta}$ and Θ, with equality holding only if $\hat{\Theta} = \Theta$. If we choose $\hat{\Theta}$ to maximize the auxiliary function $Q(\hat{\Theta}, \Theta)$, then it follows that:

$$\mathcal{L}(\hat{\Theta}) \geq Q(\hat{\Theta}, \Theta) \geq Q(\Theta, \Theta) = \mathcal{L}(\Theta). \qquad (19)$$

Thus, updates derived in this way are guaranteed to lead to monotonic increases in the log-likelihood. Assuming furthermore that the functions in eq. (19) are smooth and well-behaved, it follows that the updates converge only to stationary points of the log-likelihood.

This procedure leads to the ALS updates for logistic PCA. The U-update is obtained by setting $\hat{\Theta}_{nd} = (\hat{U}V)_{nd} + \Delta_d$ and $\Theta_{nd} = (UV)_{nd} + \Delta_d$ and maximizing the auxiliary function $Q(\hat{\Theta}, \Theta)$ with respect to $\hat{U}$. Since the auxiliary function is quadratic in the elements of the matrix $\hat{\Theta}$, it is also quadratic in the elements of the matrix $\hat{U}$. The required maximization therefore reduces to a least squares problem that can be solved in closed form. In particular, setting

$$0 = \frac{\partial Q}{\partial \hat{U}_{n\ell}} = \sum_d \frac{\partial Q}{\partial \hat{\Theta}_{nd}} \left(\frac{\partial \hat{\Theta}_{nd}}{\partial \hat{U}_{n\ell}} \right) = \sum_d \frac{\partial Q}{\partial \hat{\Theta}_{nd}} V_{\ell d} \qquad (20)$$

and solving for $\hat{U}$ leads to the U-update in eq. (10). The V-update and Δ-update are obtained in an analogous way.