Dimensionality Reduction with Generalized Linear Models

Mo Chen,[1] **Wei Li,**[1] **Wei Zhang,**[2] **Xiaogang Wang**[1]
[1]Department of Electronic Engineering
[2]Department of Information Engineering
The Chinese University of Hong Kong
[1]*{mchen,liwei,xgwang}@ee.cuhk.edu.hk*, [2]*wzhang009@gmail.com*

Abstract

In this paper, we propose a general dimensionality reduction method for data generated from a very broad family of distributions and nonlinear functions based on the generalized linear model, called Generalized Linear Principal Component Analysis (GLPCA). Data of different domains often have very different structures. These data can be modeled by different distributions and reconstruction functions. For example, real valued data can be modeled by the Gaussian distribution with a linear reconstruction function, whereas binary valued data may be more appropriately modeled by the Bernoulli distribution with a logit or probit function. Based on general linear models, we propose a unified framework for extracting features from data of different domains. A general optimization algorithm based on natural gradient ascent on distribution manifold is proposed for obtaining the maximum likelihood solutions. We also present some specific algorithms derived from this framework to deal with specific data modeling problems such as document modeling. Experimental results of these algorithms on several data sets are shown for the validation of GLPCA.

1 Introduction

Principal component analysis (PCA) is one of the most widely used methods for dimensionality reduction of multivariate data. Given a set of sample points $\mathbf{y}_i \in \mathbb{R}^d$ for $i = 1, \ldots, n$, PCA finds a lower dimensional subspace that minimizes the sum of the square distances from the data points $\mathbf{y}_i$ to their projections $\boldsymbol{\mu}_i$ in the subspace. In other words, PCA finds a linear reconstruction of the data that is optimal in the least square sense. PCA also can be interpreted from a probabilistic viewpoint that each point $\mathbf{y}_i$ is thought as a random draw from an isotropic Gaussian with mean $\boldsymbol{\mu}_i$ and variance ψI where ψ is shared among all samples. In other words, each $\mathbf{y}_i$ is considered as a noise-corrupted sample of the true data point $\boldsymbol{\mu}_i$ which lies in a lower dimensional subspace.

However, the assumption of the Gaussian noise model and the linear reconstruction is not always appropriate for data of different domains. For example, binary data are more suitable to be modeled by Bernoulli with a logit or probit reconstruction (link) function. In order to easily obtain a proper representation for data using appropriate distributions and reconstruction functions, we want to have a general framework for data modeling, so that we can easily incorporate specific distribution and reconstruction function which are suitable for the data. The first step toward this goal was made by [Collins *et al.*, 2002], in which the authors extended PCA to exponential family distributions (called exponential family PCA or EPCA). EPCA is based on the minimization of a generalized criterion in terms of the Bregman divergence, which has a duality relationship with exponential family distributions.

In this paper, we propose a probabilistic model called generalized linear principal component analysis (GLPCA), which is based on the generalized linear model to solve dimensionality reduction problems. Different from the algorithm for the EPCA model, which is based on Bregman divergence minimization, we propose a algorithm which directly maximizes the likelihood using the so called natural gradient ascent on the Riemannian manifold of the parameter space. The algorithm makes incorporating domain specific distributions and reconstruction functions a easy task which simply reduces to input the sufficient statistics to the algorithm. We demonstrate the logit and probit GLPCA algorithms for modeling documents.

It is worth mentioning that the paper [Gordon, 2003] also proposed a very general framework for both regression and factor analysis problems called $(\mathrm{GL})^2\mathrm{M}$. This framework in a sense, is more general than the generalized linear model. It uses three link functions instead of one in regular GLM. However, this formulation causes the problem highly non-convex. The subproblem of the optimization procedure is not only non-convex, but it can have exponentially many local minima [Kivinen and Warmuth, 2001]. The author proposed a Newton typed meta algorithm for their model. However, one has to make non-trivial effort to derive the required quantities (e.g. Hessian) in order to incorporate specific distributions and link functions.

In this paper, we denote scalars with lowercase letters (x), vectors with bold lowercase letters ($\mathbf{x}$), matrices with bold uppercase letters ($\mathbf{X}$) and high order tensors with calligraphy uppercase letters ($\mathcal{T}$).

2 Generalized Linear Principal Component Analysis Model

The classical PCA is a linear model which assumes that an observation $\mathbf{y}_i$ is generated from a linear transformation of a latent low dimensional vector $\mathbf{x}_i$ plus a bias term $\mathbf{m}$ and a Gaussian noise term $\boldsymbol{\epsilon}$,

$$\mathbf{y}_i = \mathbf{W}^T\mathbf{x}_i + \mathbf{m} + \boldsymbol{\epsilon}. \tag{1}$$

The least square estimator is adapted to this assumption. However, the restriction of linearity is too strict for a variety of practical situations. For example, the linear model fails, when dealing with binary data (Bernoulli) or with count data (Poisson).

The generalized linear model (GLM) [Nelder and Wedderburn, 1972] was proposed to relax the assumption to a broad family of nonlinear models for regression problems. In [Rish *et al.*, 2008], the authors utilized the GLM for supervised dimensionality reduction problems. In this paper, we propose the GLPCA model to extend PCA with GLM for dimensionality reduction problems. The essential feature of GLPCA is that the observation $\mathbf{y}_i$ is a sample from certain distribution $p(\mathbf{y})$. The expectation $\boldsymbol{\mu}_i = E[\mathbf{y}_i]$ is a monotonic function of $\boldsymbol{\eta}_i$ such that

$$E[\mathbf{y}_i] = \boldsymbol{f}(\boldsymbol{\eta}_i),\ \boldsymbol{\eta}_i = \mathbf{W}^T\mathbf{x}_i + \mathbf{m}. \tag{2}$$

Here, we overload the notation $\boldsymbol{f}(\boldsymbol{\eta})$ as a vector function that each elements is obtained by applying f over each elements of vector $\boldsymbol{\eta}$. The conditional distribution $p(\boldsymbol{y}|\mathbf{W}, \mathbf{x}, \mathbf{m})$ is a distribution of natural exponential family, which will be discussed later. The reconstruction function $\boldsymbol{f}$ is also called the link function in GLM literature. Given the i.i.d. samples $\{\mathbf{y}_i\} \in \mathbb{R}^d$, our goal is to find the optimal $\mathbf{W} \in \mathbb{R}^{d\times q}$, $\mathbf{m} \in \mathbb{R}^d$, and $\{\mathbf{x}_i\} \in \mathbb{R}^q$ in the sense that the likelihood of this model is maximized.

2.1 Exponential Family

We first introduce the natural exponential family, from which we derive our GLPCA model. Some quantities of the natural exponential family are derived and will be used latter in the algorithm.

In the GLPCA model, we assume that the distribution of the observation $\mathbf{y}_i$ is a member of the multivariate natural exponential family [Wainwright and Jordan, 2008]. The natural exponential family is a class of probability distributions where the first order statistics are of primary interest. It covers a broad range of distributions, such as Gaussian, Gamma, Bernoulli and Poisson distributions. The density functions of the natural exponential family distributions have a general form

$$p(\mathbf{y}|\boldsymbol{\theta}, \psi) = \exp\left(\frac{\mathbf{y}^T\boldsymbol{\theta} - A(\boldsymbol{\theta})}{\psi} + C(\mathbf{y}, \psi)\right). \tag{3}$$

$\mathbf{y}$ is a d dimensional observation. $\boldsymbol{\theta}$ is the natural parameter. ψ is a scalar dispersion parameter and is a scale parameter of the variance. $A(\boldsymbol{\theta})$ is called the partition function which plays an important role in our latter derivation. The expectation and covariance of $\mathbf{y}$ are

$$\begin{aligned} E[\mathbf{y}] =& \nabla A(\boldsymbol{\theta}) \\ \mathrm{Cov}[\mathbf{y}] =& \psi \nabla\nabla^T A(\boldsymbol{\theta}). \end{aligned} \tag{4}$$

For convenience, we denote

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{\theta}) =& \nabla A(\boldsymbol{\theta}), \\ \boldsymbol{\Sigma}(\boldsymbol{\mu}) =& \nabla\nabla^T A(\boldsymbol{\theta}). \end{aligned} \tag{5}$$

We can see that the expectation of $\mathbf{y}$ only depends on the natural parameter $\boldsymbol{\theta}$ whereas the variance of $\mathbf{y}$ depends on both $\boldsymbol{\theta}$ and the parameter ψ. Typically one assumes that the factor ψ is identical over all observations. One example is the isotropic Gaussian distribution which can be written in the standard form as

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\theta}, \psi) =& \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|\mathbf{y}-\boldsymbol{\mu}\|^2}{2\sigma^2}\right), \\ =& \exp\left(\frac{\mathbf{y}^T\boldsymbol{\theta} - \boldsymbol{\theta}^T\boldsymbol{\theta}/2}{\psi} - \frac{1}{2}\left(\frac{\mathbf{y}^T\mathbf{y}}{\psi} + \ln(2\pi\psi)\right)\right), \end{aligned} \tag{6}$$

where $\boldsymbol{\theta} = \boldsymbol{\mu}$,$\psi = \sigma^2$,$A(\boldsymbol{\theta}) = \boldsymbol{\theta}^T\boldsymbol{\theta}/2$, $g(\boldsymbol{\mu}) = \boldsymbol{\mu}$, $\boldsymbol{\Sigma} = \boldsymbol{I}$, and $C(\mathbf{y}, \psi) = -(\mathbf{y}^T\mathbf{y}/\psi + \ln(2\pi\psi))/2)$,

If the degree of freedom of $\mathbf{y}$ is d, then the representation is said to be minimal; otherwise it is over-complete. If the representation is minimal, the multivariate density can be factorized as

$$p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{j=1}^{d} p(y_j|\theta_j). \tag{7}$$

Then it can be easily verified that $A(\boldsymbol{\theta}) = \sum_{j=1}^{d} A(\theta_j)$ where we overload the notation $A(\theta_j)$ to be the partition function of the univariate distribution $p(y_j|\theta_j)$. As a result, the covariance matrix $\boldsymbol{\Sigma}$ becomes a diagonal matrix with diagonal elements are $\Sigma_{jj} = \psi A''(\theta_j)$. This is the case for the isotropic Gaussian. For convenience, we define

$$\nu(\mu) = A''(\theta). \tag{8}$$

For a over-complete representation, the degree of freedom of $\mathbf{y}$ is smaller than d. For example, if the sample vector represents a discrete distribution, i.e., $\sum_{j=1}^{d} y_j = 1$ and $y_j > 0$, it is natural to model the data by the Dirichlet distribution. However, if we write the Dirichlet distribution in the form of natural exponential family (3), the representation is not minimal. Another example is that if the samples are normalized such that $\|\mathbf{y}\|_2 = 1$, it is natural to model the data by the von Mises-Fisher distribution, of which the representation in the form of (3) again is not minimal. For certain distributions, transforming from the over-complete representations to the minimal ones is non-trivial. The previous EPCA algorithm is based on minimizing the Bregman divergence, and has an one-to-one map to the natural exponential family with the minimal representation [Wainwright and Jordan, 2008]. Therefore, EPCA only works with the natural exponential family with minimal representation. On the other hand, our GLPCA does not have this restriction. It works with both minimal and over-complete representations.

2.2 Link Function

Apart from the distribution of $\mathbf{y}$, the link function is another important part of GLPCA. From previous section, we have

$$\boldsymbol{\mu} = \boldsymbol{f}(\boldsymbol{\eta}) = \nabla A(\boldsymbol{\theta}) \tag{9}$$

In the case that $\boldsymbol{\theta}(\boldsymbol{\eta}) = \boldsymbol{\eta}$, i.e., $\boldsymbol{f} = \nabla A$, the function $\boldsymbol{f}$ is called canonical link. For models with a canonical link, some theoretical and practical problems are easier to solve. If a canonical link is used and the data are modeled by the natural exponential family with the minimal representation, GLPCA is equivalent to EPCA [Collins *et al.*, 2002] (see Section 4). Table 1 summarizes the characteristics for some exponential family distributions together with natural parameters and canonical link functions. GLPCA also can work with non-canonical link functions. For a specific distribution, a family of special link functions is valid. For example, the (canonical) logit link and cumulative Gaussian link can both be used with the Bernoulli distribution, called logistic and probit model respectively. A very flexible class of link functions is the class of power functions which are also called the Box-Cox transformations [Box and Cox, 1964]. They can be defined for all models for which we have observations with positive means. This family is usually specified as

$$\eta = \begin{cases} \mu^\lambda & \text{if } \lambda \neq 0; \\ \ln \mu & \text{if } \lambda = 0. \end{cases} \tag{10}$$

We summarize some commonly used link functions in Table 2

3 Maximum Likelihood Estimation

In this section, we present the algorithm for maximum likelihood estimation based on the natural gradient ascent in the Riemannian manifold of distributions.

3.1 Natural Gradient in Riemannian Manifold of Distributions

Consider a family S of parameterized distributions $p(\mathbf{x}|\boldsymbol{\theta})$. Under some smoothness conditions, $S = \{p_{\boldsymbol{\theta}}\}$ can be considered as a differential manifold [Amari and Nagaoka, 2000]. The log-likelihood, denoted as $\ell(\mathbf{y};\boldsymbol{\theta}) = \ln p(\mathbf{y}|\boldsymbol{\theta})$ is a scalar function in this manifold. The Fisher information matrix is defined as $\mathbf{G} = [g_{ij}]$, where

$$g_{ij}(\boldsymbol{\theta}) = E_{\mathbf{y}}\left[\frac{\partial \ell}{\partial \theta_i}\frac{\partial \ell}{\partial \theta_j}\right]. \tag{11}$$

By definition, the $\mathbf{G}$ is always symmetric and positive semidefinite. Under certain regularity condition over the distribution $p(\mathbf{x}|\boldsymbol{\theta})$ (such as exponential family), the Fisher information matrix is positive definite. For positive definite Fisher information matrix, a Riemannian manifold is induced by defining the Riemannian metric over the differential manifold S as

$$\langle \partial_i, \partial_j \rangle_{\boldsymbol{\theta}} = g_{ij}(\boldsymbol{\theta}), \tag{12}$$

which is also called Fisher metric or information metric in information geometry literature. It can be proved that the only Riemannian metric is Fisher metric that the geometry is invariant under coordinate transformations of $\boldsymbol{\theta}$ and also under one-to-one transformations of random variable $\mathbf{y}$. The technical detail can be found in [Amari and Nagaoka, 2000].

The steepest ascent direction of a function $\ell(\boldsymbol{\theta})$ in a Riemannian manifold is given by

$$\tilde{\nabla}\ell(\boldsymbol{\theta}) = \mathbf{G}^{-1}(\boldsymbol{\theta})\nabla\ell(\boldsymbol{\theta}), \tag{13}$$

where $\nabla\ell(\boldsymbol{\theta})$ is the conventional gradient

$$\nabla\ell(\boldsymbol{\theta}) = \left[\frac{\partial}{\partial\theta_1}\ell(\boldsymbol{\theta}), \ldots, \frac{\partial}{\partial\theta_d}\ell(\boldsymbol{\theta})\right]^T. \tag{14}$$

$\tilde{\nabla}\ell(\boldsymbol{\theta})$ is called natural gradient of ℓ in the Riemannian manifold [Amari, 1998]. When the manifold is Euclidean space and the coordinate system is orthonormal, we have $\tilde{\nabla}\ell(\boldsymbol{\theta}) = \nabla\ell(\boldsymbol{\theta})$. This argument suggests the natural gradient ascent algorithm of the form

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \tau\tilde{\nabla}\ell(\boldsymbol{\theta}_t), \tag{15}$$

where τ is the learning rate that determines the step size. Since the Riemannian manifold is invariant under transformation of coordinate $\boldsymbol{\theta}$ and random variable $\mathbf{y}$, the natural gradient ascent algorithm is also invariant under the transformation. The natural gradient method has been successfully applied on various machine learning models, including neural network and ICA [Amari and Nagaoka, 2000].

3.2 Maximizing Likelihood of GLPCA

For GLPCA model, the log-likelihood of the observations is

$$\begin{aligned} L &= \sum_{i=1}^{n} \ell(\mathbf{y}_i;\boldsymbol{\theta}_i) \\ &= \sum_{i=1}^{n}\left(\frac{\mathbf{y}_i^T\boldsymbol{\theta}_i - A(\boldsymbol{\theta}_i)}{\psi} + C(\mathbf{y}_i,\psi)\right), \end{aligned} \tag{16}$$

where

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}(\boldsymbol{\eta}_i) = \boldsymbol{\theta}(\mathbf{W}^T\mathbf{x}_i + \mathbf{m}). \tag{17}$$

Our goal is to maximize L w.r.t. $\mathbf{X} = [\mathbf{x}_i]_{i=1}^n$,$\mathbf{W} = [\mathbf{w}_j]_{j=1}^d$, and $\mathbf{m}$. Note that ψ and $C(\mathbf{y}_i,\psi)$ have no influence on our objective.

To apply the natural gradient method to GLPCA model, we have to derive the gradients and the Fisher information matrices. However the parameter $\mathbf{W}$ and $\mathbf{X}$ are matrices. To proceed, in this paper we utilize the tensor notation [Kolda and Bader, 2009] for our derivation.

Taking the first derivative of (16) w.r.t. $\mathbf{X}$, $\mathbf{W}$ and $\mathbf{m}$ yields the gradients

$$\begin{aligned} \nabla_{\mathbf{X}}L =& \mathbf{W}\mathbf{Z}, \\ \nabla_{\mathbf{W}}L =& \mathbf{X}\mathbf{Z}^T, \\ \nabla_{\mathbf{m}}L =& \mathbf{Z}\mathbf{1}, \end{aligned} \tag{18}$$

where

$$\mathbf{Z} = [\nabla\boldsymbol{f}^T(\boldsymbol{\eta}_i)\boldsymbol{\Sigma}_i^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_i)]_{i=1}^n, \tag{19}$$

$\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}(\boldsymbol{\mu}_i)$ and $\mathbf{1} = [1,\ldots,1]^T$. By definition, one can easily show that the Fisher information matrix is

$$\mathbf{G}(\boldsymbol{\theta}) = -E_{\mathbf{y}}[\nabla^2_{\boldsymbol{\theta}}L]. \tag{20}$$

For our problem, since the parameters are of the matrix form, the Fisher information are tensors, which are given by[1]

$$\begin{aligned} \mathcal{G}(\mathbf{X}) &= -\mathcal{T}\times_1\mathbf{W}\times_3\mathbf{W}, \\ \mathcal{G}(\mathbf{W}) &= -\mathcal{T}\times_2\mathbf{X}\times_4\mathbf{X}, \\ \mathcal{G}(\mathbf{m}) &= -\mathcal{T}\times_2\mathbf{1}^T\times_4\mathbf{1}^T, \end{aligned} \tag{21}$$

[1] Due to the restriction of the paper length, we omit the proof.

Table 1: Characteristics of some exponential family distributions.

Distribution	$A(\theta)$	$g(\theta) = \nabla A(\theta)$	$\Sigma(\mu)$ or $\nu(\mu)$	ψ
Gaussian $N(\mu, \sigma^2)$	$\theta^2/2$	θ	1	σ^2
Bernoulli $B(\mu)$	$\ln(1 + e^\theta)$	$\ln\left(\frac{e^\theta}{1+e^\theta}\right)$	$\mu(1 - \mu)$	1
Poisson $P(\mu)$	e^θ	e^θ	μ	1
Gamma $A(\mu, \nu)$	$-\ln(-\theta)$	$-1/\theta$	μ^2	$1/\nu$

where $\mathcal{T}$ is a 4th-order tensor with entries

$$\mathcal{T}_{\cdot,i,\cdot,i} = \nabla f^T(\boldsymbol{\eta}_i)\boldsymbol{\Sigma}_i^{-1}\nabla^T f(\boldsymbol{\eta}_i) \tag{22}$$

and all the other entries being zeros. The notation $\times_t$ is the tensor t-mode product, i.e., multiplying a tensor by a matrix (or a vector) in mode t.

To compute the natural gradient, we have to unfold the tensors $\mathcal{G}(\cdot)$ into the matrices $\mathbf{G}(\mathrm{vec}(\cdot))$. The unfolded Fisher information matrix $\mathbf{G}(\mathrm{vec}(\mathbf{X}))$ is a block diagonal matrix with the diagonal blocks of size $d \times d$ given by

$$\mathbf{G}_{ii}(\mathrm{vec}(\mathbf{X})) = -\mathbf{W}\mathcal{T}_{\cdot,i,\cdot,i}\mathbf{W}^T. \tag{23}$$

The Fisher information matrix $\mathbf{G}(\mathrm{vec}(\mathbf{W}))$ comprises blocks of size $d \times d$ in which block jk is given by

$$\mathbf{G}_{jk}(\mathrm{vec}(\mathbf{W})) = -\mathbf{X}\mathcal{T}_{j,\cdot,k,\cdot}\mathbf{X}^T. \tag{24}$$

The Fisher information matrix $\mathbf{G}(\mathbf{m})$ is simply

$$\mathbf{G}(\mathbf{m}) = -\sum_{i=1}^{n} \mathcal{T}_{\cdot,i,\cdot,i}. \tag{25}$$

The gradients are easily unfolded as

$$\begin{aligned} \nabla_{\mathrm{vec}(\mathbf{X})}L &= \mathrm{vec}(\nabla_{\mathbf{X}}L), \\ \nabla_{\mathrm{vec}(\mathbf{W})}L &= \mathrm{vec}(\nabla_{\mathbf{W}}L). \end{aligned} \tag{26}$$

Then the natural gradients w.r.t. to $\mathbf{X}$, $\mathbf{W}$ and $\mathbf{m}$ are

$$\begin{aligned} \tilde{\nabla}_{\mathrm{vec}(\mathbf{X})}L &= \mathbf{G}(\mathrm{vec}(\mathbf{X}))^{-1}\nabla_{\mathrm{vec}(\mathbf{X})}L, \\ \tilde{\nabla}_{\mathrm{vec}(\mathbf{W})}L &= \mathbf{G}(\mathrm{vec}(\mathbf{W}))^{-1}\nabla_{\mathrm{vec}(\mathbf{W})}L, \\ \tilde{\nabla}_{\mathbf{m}}L &= \mathbf{G}(\mathbf{m})^{-1}\nabla_{\mathbf{m}}L. \end{aligned} \tag{27}$$

Our algorithm for solving the ML solution of (16) is to iterate through the following three steps

$$\begin{aligned} \mathrm{vec}(\mathbf{X}^*) =& \mathrm{vec}(\mathbf{X}) + \tau\tilde{\nabla}_{\mathrm{vec}(\mathbf{X})}L, \\ \mathrm{vec}(\mathbf{W}^*) =& \mathrm{vec}(\mathbf{W}) + \tau\tilde{\nabla}_{\mathrm{vec}(\mathbf{W})}L, \\ \mathbf{m}^* =& \mathbf{m} + \tau\tilde{\nabla}_{\mathbf{m}}L, \end{aligned} \tag{28}$$

until converge. Note that in practice, the natural gradients in (27) need not be exactly computed. One can apply the preconditioned conjugate gradient descent method to solve the linear system

$$\mathbf{G}\tilde{\nabla}L = \nabla L \tag{29}$$

for $\tilde{\nabla}L$ in a few steps to produce approximated natural gradients with an acceptable precision which is sufficient for updating our parameters. This method is very efficient.

4 Exponential Family PCA

If a canonical link function is used, we recover the exponential family PCA. Following our natural gradient method for ML estimation, we obtain an efficient algorithm for exponential family PCA.

If the canonical link function $\boldsymbol{f}(\cdot) = \boldsymbol{g}(\cdot) = \nabla A(\cdot)$ is used in GLPCA, the gradients in (18) become

$$\begin{aligned} \nabla_{\mathbf{X}}L =& \mathbf{W}[\mathbf{Y} - \boldsymbol{f}(\mathbf{W}^T\mathbf{X} + \mathbf{m}\mathbf{1}^T)], \\ \nabla_{\mathbf{W}}L =& \mathbf{X}[\mathbf{Y} - \boldsymbol{f}(\mathbf{W}^T\mathbf{X} + \mathbf{m}\mathbf{1}^T)]^T, \\ \nabla_{\mathbf{m}}L =& [\mathbf{Y} - \boldsymbol{f}(\mathbf{W}^T\mathbf{X} + \mathbf{m}\mathbf{1}^T)]\mathbf{1}. \end{aligned} \tag{30}$$

The tensor $\mathcal{T}$ can be simplified with entries $\mathcal{T}_{\cdot,i,\cdot,i} = \boldsymbol{\Sigma}_i$ and all other entries being zeros. If the distribution is minimally represented in the form of natural exponential family (3), $\boldsymbol{\Sigma}_i$ is a diagonal matrix. Denote $\mathbf{V}_i = \mathcal{T}_{\cdot,i,\cdot,i}$ and $\mathbf{U}_j = \mathcal{T}_{j,\cdot,j,\cdot}$, we have

$$\begin{aligned} \mathbf{V}_i &= \mathrm{diag}[\nu(g(\mathbf{w}_j^T\mathbf{x}_i + \mathbf{m}))]_{j=1}^{d} \\ \mathbf{U}_j &= \mathrm{diag}[\nu(g(\mathbf{w}_j^T\mathbf{x}_i + \mathbf{m}))]_{i=1}^{n} \end{aligned} \tag{31}$$

The algorithm (28) now becomes

$$\begin{aligned} \mathbf{x}_i^* &= \mathbf{x}_i + \tau(\mathbf{W}\mathbf{V}_i\mathbf{W}^T)^{-1}\mathbf{W}[\mathbf{y}_i - \boldsymbol{f}(\mathbf{W}^T\mathbf{x}_i + \mathbf{m})], \\ \mathbf{w}_j^* &= \mathbf{w}_j + \tau(\mathbf{X}\mathbf{U}_j\mathbf{X}^T)^{-1}\mathbf{X}[\mathbf{Y}_{j\cdot}^T - \boldsymbol{f}(\mathbf{X}^T\mathbf{w}_j + m_j\mathbf{1}^T)], \\ \mathbf{m}^* &= \mathbf{m} + \tau\mathbf{V}^{-1}[\mathbf{Y} - \boldsymbol{f}(\mathbf{W}^T\mathbf{X} + \mathbf{m}\mathbf{1}^T)]\mathbf{1}, \end{aligned} \tag{32}$$

where $\mathbf{V} = \sum_{i=1}^{n} \mathbf{V}_i$.

In [Collins *et al.*, 2002], the authors proposed the exponential family PCA (EPCA) model which minimizes the Bregman divergence

$$\sum_{i=1}^{n} B_F(\mathbf{y}_i \| \boldsymbol{g}(\boldsymbol{\theta}_i)) = \sum_{i=1}^{n}\sum_{j=1}^{d} B_F(y_{ij} \| g(\theta_{ij})), \tag{33}$$

where B_F is the Bregman divergence associated with function F. There is a duality relationship between the Bregman divergence $B_F(y\|g(\theta))$ and the natural exponential family $p(y) = \exp(y\theta - A(\theta) + C(y))$. We can express the negative log-likelihood through Bregman divergence as [Azoury and Warmuth, 2001; Forster and Warmuth, 2002]

$$-\ln p(y|\theta) = -\ln C(y) - F(y) + B_F(y\|g(\theta)). \tag{34}$$

The model implicitly assumes that the distribution of the sample vector $\mathbf{y}$ satisfies the factorization condition (7). In other words, the distribution of the samples must be minimally represented by (3) and the degree of freedom of any sample $\mathbf{y}_i$ must be d. This assumption excludes some commonly used distributions, such as von Mises-Fisher. The Bregman divergence based EPCA model is equivalent to a special case of our GLPCA model, where the distribution used is assumed satisfying the factorization condition (7) and the canonical link function corresponding to the distribution is used.

Table 2: Link functions.

Link	$\eta = g(\mu)$	$\mu = g^{-1}(\eta)$	Range
Log	$\ln \mu$	e^{η}	$\mu \geq 0$
Negative log-log	$-\ln(-\ln(\mu))$	$\exp(-\exp(-\eta))$	$\mu \in [0, 1]$
Complementary log-log	$\ln(-\ln(1-\mu))$	$1 - \exp(-\exp(\eta))$	$\mu \in [0, 1]$
Negative Binomial	$\ln\left(\frac{\mu}{\mu+\frac{1}{k}}\right)$	$\frac{\exp(\eta)}{k(1-\exp(\eta))}$	$\mu \geq 0$
Probit	$\Phi^{-1}(\mu)$	$\Phi(\eta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\eta} e^{-t^2/2} dt$	$\mu \in [0, 1]$

5 Example Models

5.1 Linear PCA

When the Gaussian noise model is assumed and the canonical link (identity function) is used, the GLPCA algorithm simply iterates through following steps

$$\begin{aligned} \mathbf{X} &= (\mathbf{W}\mathbf{W}^T)^{-1}\mathbf{W}(\mathbf{Y} - \mathbf{m}\mathbf{1}^T), \\ \mathbf{W} &= (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}(\mathbf{Y} - \mathbf{m}\mathbf{1}^T)^T, \\ \mathbf{m} &= (\mathbf{Y} - \mathbf{W}^T\mathbf{X})\mathbf{1}/n. \end{aligned} \tag{35}$$

This algorithm is similar to the EM algorithm for PCA (EMPCA) which was introduced in [Roweis, 1998]. The difference is that in EMPCA the bias term $\mathbf{m}$ is set to be the mean of the data a priori and subtracted from the data before applying the EM algorithm. In our algorithm, the bias term is optimized together with other parameters. When the algorithm converges, the bias term obtained is not necessarily the mean of the data, since the solution is not unique. However the subspace spanned by $\mathbf{W}$ is identical to the one obtained by standard PCA.

5.2 Logistic PCA

Suppose we are given a data set of which each sample $\mathbf{y}_i$ is a binary valued vector. It is convenient to model the data with the multivariate Bernoulli distribution, i.e.

$$\begin{aligned} p(y_j|\theta_j) &= \theta_j^{y_j}(1-\theta_j)^{1-y_j}, \\ p(\mathbf{y}|\boldsymbol{\theta}) &= \prod_{j=1}^{d} p(y_j|\theta_j). \end{aligned} \tag{36}$$

The canonical link function of Bernoulli distribution is the logistic function. In GLPCA, if Bernoulli and logistic link function are used, we call this model logistic PCA. For logistic PCA, Following (31), we have

$$\begin{aligned} \mathbf{V}_i &= \mathrm{diag}[\nu(\mu_{ij})]_{j=1}^{d}, \\ \mathbf{U}_j &= \mathrm{diag}[\nu(\mu_{ij})]_{i=1}^{n}. \end{aligned} \tag{37}$$

where $\nu(x) = x(1-x)$ and μ_{ij} is the j-th element of $\boldsymbol{\mu}_i$. Then the ML solution of logistic PCA can be obtained by iterating through (32) with the definition (37).

5.3 Probit PCA

Besides the logistic model, we can also use the probit model for binary data. Probit PCA has the same assumption of multivariate Bernoulli distribution as logistic PCA. The difference is that we use the cumulative Gaussian link $\mu_{ij} = \Phi(\eta_{ij})$, where

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt \tag{38}$$

is the CDF of standard normal distribution. Following (22), we have

$$\begin{aligned} \mathbf{V}_i &= \mathrm{diag}[\phi(\eta_{ij})^2/\nu(\mu_{i1})]_{j=1}^{d}, \\ \mathbf{U}_i &= \mathrm{diag}[\phi(\eta_{ij})^2/\nu(\mu_{i1})]_{i=1}^{n}, \end{aligned} \tag{39}$$

where $\phi(\cdot)$ is the PDF of standard normal distribution and η_{ij} is the j-th element of $\boldsymbol{\eta}_i$. The ML solution of probit PCA can also be obtained with algorithm (32). Probit PCA provides a nice alternative to logistic PCA.

6 Experiments

In this section, we demonstrate our GLPCA method on some benchmark data sets. We compare the logistic and probit GLPCA with EPCA on binary data of varying size, dimensionality, and sparsity. Results of standard PCA based on singular value decomposition of mean-centered data is also presented. We use two datasets from the UCI machine learning repository. One is the anonymous Microsoft Web Database and we select the first 5000 instances with 285 binary variables in each instance for experiments. The proportion of nonzero variables in the data is $\rho = 0.011$. The other dataset is the Advertisement Data consisting of both continuous and binary variables. We only use the 1555 binary features for each instance and remove the corrupted ones. Finally we have 3264 instances with $\rho = 0.0082$.

We use the reconstruction accuracy as the evaluation criterion. The binary reconstructions are computed by thresholding the continuous values estimated from the low-dimensional representation. Two kinds of measures, a minimum error rate and a balanced error rate, are employed. The minimum error rate is obtained by choosing the best threshold to minimize the overall error rate, and the balanced error rate is obtained by choosing the threshold to equalize the false positive and false negative error rates. As the experimental data are very sparse, both error rates capture different notions of the performance.

The experimental results are shown in Table 3. We can see that both logistic and probit GLPCA outperform the standard PCA on all tasks. Our probit GLPCA outperforms the logistic EPCA algorithm [Schein *et al.*, 2003] in some experiments. It shows different link functions may have strength for modeling different data sets. For binary data, the probit GLPCA is a good alternative of logistic model.

Table 3: Minimum and balanced error rates on two datasets for the task of binary data reconstruction. Each dataset is a $d \times n$ binary matrices with ρdn nonzero elements. The best results for each setting are shown in bold.

Anonymous Microsoft Web Database ($n = 5000$, $d = 285$, $\rho = 0.011$)

q	Minimum Error Rates (%)				Balanced Error Rates (%)			
	Linear PCA	Logistic EPCA	Logistic GLPCA	Probit GLPCA	Linear PCA	Logistic EPCA	Logistic GLPCA	Probit GLPCA
1	0.905	0.926	**0.907**	0.969	14.9	12.2	**11.6**	11.7
2	0.830	0.718	**0.713**	0.757	13.9	10.6	9.75	**8.05**
4	0.725	0.499	**0.472**	0.625	13.4	6.01	5.57	**4.31**
8	0.654	0.148	**0.137**	0.148	13.1	1.97	1.74	**1.47**

Advertisement Data ($n = 3264$, $d = 1555$, $\rho = 0.0082$)

q	Minimum Error Rates (%)				Balanced Error Rates (%)			
	Linear PCA	Logistic EPCA	Logistic GLPCA	Probit GLPCA	Linear PCA	Logistic EPCA	Logistic GLPCA	Probit GLPCA
1	0.724	**0.697**	0.699	**0.697**	25.8	18.6	14.8	**13.22**
2	0.698	0.500	**0.444**	0.451	22.8	7.34	6.80	**6.52**
4	0.652	0.223	0.224	**0.222**	20.8	3.58	3.37	**3.29**
8	0.629	0.0427	**0.0415**	0.0428	20.4	0.852	0.789	**0.563**

7 Conclusion

In this paper we proposed a general dimensionality reduction method based on a generalized linear model, called GLPCA. This general method admits a larger range of distributions and link functions. It further extends the previous exponential family PCA model to use non-canonical link functions. An unified meta algorithm based on natural gradient ascent in Riemannian manifold of distributions was proposed for maximizing the likelihood of the model. Deriving the algorithm for specific distribution and link function can be easily done by plugging in corresponding quantities. We also derived the logistic and probit GLPCA from our framework for binary data. Experimental results were shown to validate the effectiveness of the proposed method.

References

[Amari and Nagaoka, 2000] S. Amari and H. Nagaoka. *Methods of information geometry*. Amer Mathematical Society, 2000.

[Amari, 1998] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[Azoury and Warmuth, 2001] K.S. Azoury and MK Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

[Box and Cox, 1964] GEP Box and DR Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211–252, 1964.

[Collins *et al.*, 2002] M. Collins, S. Dasgupta, and R.E. Schapire. A generalization of principal component analysis to the exponential family. *Advances in Neural Information Processing Systems*, 1:617–624, 2002.

[Forster and Warmuth, 2002] J. Forster and M.K. Warmuth. Relative expected instantaneous loss bounds. *Journal of Computer and System Sciences*, 64(1):76–102, 2002.

[Gordon, 2003] Geoffrey J Gordon. Generalized$\hat{2}$ linear$\hat{2}$ models. *Advances in Neural Information Processing Systems*, pages 593–600, 2003.

[Kivinen and Warmuth, 2001] Jyrki Kivinen and Manfred K. Warmuth. Relative loss bounds for multidimensional regression problems. *Machine Learning*, 45(3):301–329, 2001.

[Kolda and Bader, 2009] T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[Nelder and Wedderburn, 1972] JA Nelder and RWM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972.

[Rish *et al.*, 2008] Irina Rish, Genady Grabarnik, Guillermo Cecchi, Francisco Pereira, and Geoffrey J Gordon. Closed-form supervised dimensionality reduction with generalized linear models. pages 832–839, 2008.

[Roweis, 1998] S. Roweis. EM algorithms for PCA and SPCA. *Advances in Neural Information Processing Systems*, pages 626–632, 1998.

[Schein *et al.*, 2003] A.I. Schein, L.K. Saul, and L.H. Ungar. A generalized linear model for principal component analysis of binary data. page 546431, 2003.

[Wainwright and Jordan, 2008] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.