Simple Exponential Family PCA

Jun Li
School of Computer Engineering
Nanyang Technological University

Dacheng Tao
School of Computer Engineering
Nanyang Technological University

Abstract

Bayesian principal component analysis (BPCA), a probabilistic reformulation of PCA with Bayesian model selection, is a systematic approach to determining the number of essential principal components (PCs) for data representation. However, it assumes that data are Gaussian distributed and thus it cannot handle all types of practical observations, *e.g.* integers and binary values.

In this paper, we propose simple exponential family PCA (SePCA), a generalised family of probabilistic principal component analysers. SePCA employs exponential family distributions to handle general types of observations. By using Bayesian inference, SePCA also automatically discovers the number of essential PCs. We discuss techniques for fitting the model, develop the corresponding mixture model, and show the effectiveness of the model based on experiments.

1 Introduction

Principal component analysis (PCA) is a popular dimension reduction tool for statistical data analysis. In applications of PCA, it is necessary to determine the number of essential principal components (vectors of PC loadings, "PCs" for short) for data representation. Bayesian PCA (BPCA) (Bishop, 1999a), an important extension to probabilistic PCA (PPCA) (Tipping and Bishop, 1999), has been proposed to determine the number of essential PCs. However, PPCA and BPCA assume that observations are sampled from Gaussian distributions in the *sample space*. In practice, however, this assumption is not usually true. Particularly, it certainly fails when the observations are not in the form of real-valued vectors, *e.g.* the sample space can be of integers or binary values.

Exponential family PCA (EPCA) (Collins et al., 2002), on the other hand, models general types of observed data by treating PCA in a maximum likelihood framework and defining the likelihood with exponential family distributions. However, EPCA is a deterministic model in terms of the *latent variables*, where latent variables refer to PCs and low dimensional representations (Welling et al., 2008). Thus EPCA has difficulty using Bayesian inference to deal with over-fitting problems. Recently, a probabilistic treatment to EPCA has been proposed (Mohamed et al., 2009). However, it lacks an explicit scheme for model selection, which is important in practice.

In this paper, we develop a new family of generative models of PCA, simple exponential family PCA (SePCA), in which parameters of a set of exponential family distributions are represented by combining PCs. In other words, given observations, these exponential family distributions define the likelihood functions of the latent variables, *i.e.* the PCs and the low-dimensional representations. Such likelihood functions link real-valued latent variables and observations of general types, *e.g.* integers and binary values. The link is promising because we can impose *automatic relevance determination* (ARD) (MacKay, 1995) on the real-valued latent variables and thus determine the effective number of necessary PCs. Roughly speaking, SePCA prefers simple models, and the chosen number of PCs reflects the complexity of the hidden structures in the observed data.

ARD prior makes fitting SePCA easy. In SePCA, *maximum a posteriori* (MAP) estimation can be employed for the inference on the latent variables. This is useful in the context of the exponential family likelihood functions, where exact marginalisation is commonly difficult. The simple inference also forgoes the need for configuring the conjugate prior for each specific distribution of the exponential family. We also construct mixtures of the model, where both the partition of the data and the local models are automatically determined as a part of the fitting.

In the remainder of this paper, we review relevant literatures and introduce the necessary background of PCA in

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9.

Section 2. The proposed SePCA model and the corresponding mixture extension are developed in Sections 3 and 4, respectively. In Section 5, we test the proposed model with experiments and compare the results against those from related techniques. Section 6 concludes the paper.

2 Related Work

PCA is a popular data analysis tool and has been applied to a wide variety of applications (Jolliffe, 1986; Turk and Pentland, 1991). It searches for PCs, onto which the projections of data have the maximum variance (Hotelling, 1933). Alternatively, it minimises the squared error for data reconstruction by using PCs. One concern regarding PCA and most of its variants is that they are deterministic models based on task-specific heuristic premises. It is difficult to justify in principle those heuristics, and they lack semantics for systematically treating the problem of model selection.

Probabilistic models, on the other side, provide a systematic approach to the tradeoff between over-fitting and model complexity. Tipping and Bishop (1999) proposed PPCA, where the low dimensional representation is taken as underlying latent random variables and PCs are the maximum likelihood estimation of parameters. A further step in probabilistic treatment of PCA, Bayesian PCA (BPCA), is then introduced by Bishop Bishop (1999a,b). BPCA treats PCs as random variables rather than parameters. This treatment permits prior belief on the PCs and leads to automatic model selection. The prior used by BPCA is motivated by ARD regularisation of neural networks (MacKay, 1995). ARD treats each PC as a class of weights and computes its “decay rate” (MacKay, 1995, Sec. 3 and 7) to determine whether it is relevant. It works only for real-valued PCs. Therefore, given discrete or binary observations, ARD cannot work together with the models that search in the *sample space* to find PCs for reconstructing the observed data.

On the other hand, the exponential family distributions have been used to model discrete or binary data (McCullagh and Nelder, 1989; Collins et al., 2002; Sajama and Orlitsky, 2005; Mohamed et al., 2009; Guo, 2009). By incorporating the exponential family distributions into generative models, we reach a new family of probabilistic PCA models, which can handle general types of observations, can select models automatically, and can be extended to mixture models.

Before introducing the proposed model, it is helpful to understand the maximum likelihood interpretation of PCA (in contrast to the viewpoint of maximum variance) and the integration of exponential family distributions under the PCA framework.

Maximum Likelihood Interpretation In PCA, a sample $\mathbf{x}$ is represented with a set of scores, which is used to combine the PCs. Then the corresponding combination $\boldsymbol{\theta}$ is the reconstruction of $\mathbf{x}$ by the PCA model (See Sec. 3 for more details about the denotations). From a probabilistic viewpoint, the discrepancies between $\mathbf{x}$ and $\boldsymbol{\theta}$ can be taken as noises with a particular distribution. Then $(\mathbf{x} - \boldsymbol{\theta})$ is a sample of a random noise variable, *i.e.* samples are drawn from a distribution with mean $\boldsymbol{\theta}$

Table 1: Frequently used notations

N	The number of samples in the observed data set
D	Dimension of the sample space, *i.e.* the number of features in the input space
d	Dimension of the latent space (latent dimension), *i.e.* the number of PCs
$\mathbf{x}_n$	The n-th sample in the observed data set as a D-dimensional tuple
$\mathbf{X}$	The observed data set as a matrix of size $D \times N$
$\mathbf{y}_n \in \mathbb{R}^d$	The vector of the n-th PC scores
$\mathbf{Y} \in \mathbb{R}^{d \times N}$	The matrix of PC scores
$\mathbf{w}_j \in \mathbb{R}^D$	The vector of the j-th PC loadings
$\mathbf{W} \in \mathbb{R}^{D \times d}$	The matrix of PC loadings

$$p(\mathbf{x}|\boldsymbol{\theta}) = p_{Noise}(\mathbf{x} - \boldsymbol{\theta}). \tag{1}$$

This is the likelihood function of $\boldsymbol{\theta}$ for the observed sample $\mathbf{x}$. PCA can then be interpreted as maximising the likelihood of a set of $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$. This maximum likelihood interpretation relates to the traditional view of PCA as follows: if p_{Noise} is chosen to be a Gaussian, the negative log-likelihood reduces to the squared reconstruction error.

Exponential Family PCA Based on the maximum likelihood interpretation, we can replace Gaussian distribution with general distributions, *e.g.* the exponential family (Collins et al., 2002), to naturally extend PCA to other noise models.

The conditional probability of $\mathbf{x}$ on $\boldsymbol{\theta}$ takes the *canonical form* of the exponential family of distributions

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp\{\mathbf{x}^T\boldsymbol{\theta} + g(\boldsymbol{\theta}) + h(\mathbf{x})\}, \tag{2}$$

where $\boldsymbol{\theta}$ encodes the *natural parameters*. In general, $\boldsymbol{\theta}$ does not lie in the same sample space as $\mathbf{x}$, as it does in the special case of the Gaussian distribution. The *parameter space* of $\boldsymbol{\theta}$ is Euclidean, *i.e.* the outputs in (2) are not themselves noise-free “virtual data points” as the output $\boldsymbol{\theta}$ in (1), when the data is discrete or of binary values. Instead, they are of the form of a standard real-valued vector regardless of the type of the observed data.

3 Simple Exponential Family PCA

This section details the proposed simple exponential family PCA, or SePCA for short, that can be applied to modelling general types of data. This paper uses the following notation conventions. Boldface Latin letters represent matrices and vectors, where uppercase and lowercase are for matrices and vectors, respectively. We use column vectors by default, and a vector also refers to a column in the corresponding matrix, *e.g.* $\mathbf{x}_n$ is the n-th column of $\mathbf{X}$. Frequently used symbols are listed in Tab. 1.

SePCA is defined as follows. Given observations $\mathbf{X}$, SePCA automatically determines d by introducing ARD (MacKay, 1995), and finds d vectors of PC loadings, $\mathbf{W}$, and d PC scores for each observation, collected in $\mathbf{Y}$, so each column of $\boldsymbol{\Theta} = \mathbf{W}\mathbf{Y}$ specifies the natural parameters of an exponential family distribution to generate the corresponding column of $\mathbf{X}$.

3.1 Model specification

Figure 1(a) shows the generative process of SePCA. To generate $\mathbf{x}_n$, we begin by drawing d scores as $\mathbf{y}_n$, which is the low dimensional representation of $\mathbf{x}_n$, from a Gaussian prior,

$$p(\mathbf{y}_n) = \mathcal{N}(\mathbf{y}_n|\mathbf{0}, \mathbf{I}), \tag{3}$$

where $\mathbf{0}$ and $\mathbf{I}$ are all-zero vectors and identity matrix with appropriate dimensions, respectively. The following step is to generate the PCs $\mathbf{w}_1, \ldots, \mathbf{w}_d$. Each $\mathbf{w}_j$ has an isotropic Gaussian prior controlled by a precision (inverse variance) hyper-parameter, and thus

$$p(\mathbf{W}|\boldsymbol{\alpha}) = \prod_{j=1}^{d} \mathcal{N}(\mathbf{w}_j|\mathbf{0}, \alpha_j^{-1}\mathbf{I}), \tag{4}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_d\}$ represents the set of precision hyper-parameters. The latent dimension d can be tentatively set sufficiently large, *e.g.* $d = D - 1$. Then fitted $\boldsymbol{\alpha}$ will control whether each $\mathbf{w}_j$ is valid or switched off, which is motivated by the "weight decay parameters" in ARD (MacKay, 1995).

With $\mathbf{W}$ and $\mathbf{y}_n$, the natural parameter vector $\boldsymbol{\theta}_n = \mathbf{W}\mathbf{y}_n$ is ready for the exponential family distribution that generates $\mathbf{x}_n$. Thus $\mathbf{x}_n$ is drawn from

$$p(\mathbf{x}_n|\mathbf{W}, \mathbf{y}_n) = Exp(\mathbf{x}_n|\boldsymbol{\theta}_n), \quad \boldsymbol{\theta} = \mathbf{W}\mathbf{y}_n, \tag{5}$$

where $Exp(\mathbf{x}_n|\boldsymbol{\theta}_n)$ is the exponential family distribution defined in (2). Then $p(\mathbf{x}_n|\mathbf{W}, \mathbf{y}_n)$ is the likelihood function of $\mathbf{W}$ and $\mathbf{y}_n$. Note that we may also write it as $p(\mathbf{x}|\boldsymbol{\theta}_n)$ for convenience.

Putting (3), (4) and (5) together, we arrive at the joint distribution. Its logarithmic form is given by

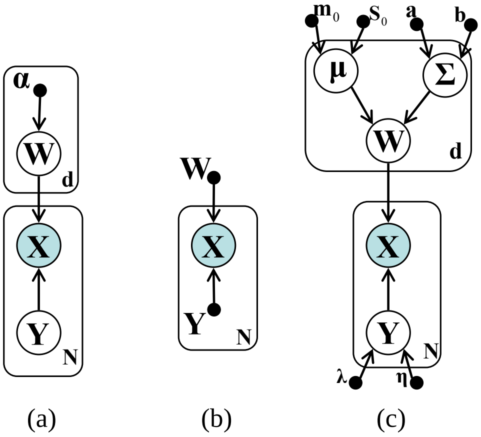


Figure 1: Diagram showing the generative process of the models: **(a)** SePCA and Bayesian PCA (Bishop, 1999a), **(b)** Exponential family PCA (Collins et al., 2002), and **(c)** Bayesian EPCA (Mohamed et al., 2009).

$$\begin{aligned}
&\log p(\mathbf{X}, \mathbf{Y}, \mathbf{W}|\boldsymbol{\alpha}) \\
=&\log p(\mathbf{X}|\mathbf{Y}, \mathbf{W}) + \log p(\mathbf{Y}) + \log p(\mathbf{W}|\boldsymbol{\alpha}) \\
=&\sum_n \Big(\mathbf{x}_n^T \mathbf{W}\mathbf{y}_n + g(\mathbf{W}\mathbf{y}_n) + h(\mathbf{x}_n)\Big) \\
&- \frac{1}{2}\Big(d(N+D)\log(2\pi) + \sum_n \|\mathbf{y}_n\|_2^2 \\
&+ \sum_{j=1}^{d} \big(\alpha_j \|\mathbf{w}_j\|_2^2 - D\log\alpha_j\big)\Big).
\end{aligned} \tag{6}$$

We graphically compare the proposed SePCA against Bayesian PCA (BPCA Bishop, 1999a), exponential family PCA (EPCA Collins et al., 2002) and Bayesian exponential family PCA (BEPCA) (Mohamed et al., 2009) in Fig. 1. The probabilistic structure of the proposed model (Fig. 1(a)) appears to resemble that of BPCA. The main difference between them is the conditional distribution of $\mathbf{x}$ given $\mathbf{W}$ and $\mathbf{y}$. In BPCA, $p(\mathbf{x}|\mathbf{W}, \mathbf{y})$ is a Gaussian, while we adopt the exponential family for $p(\mathbf{x}|\mathbf{W}, \mathbf{y})$ in order to generalise the scope of $\mathbf{x}$ from real-valued vectors to general types. EPCA is essentially a deterministic model (Fig. 1(b)). We turn its parameters $\mathbf{W}$ and $\mathbf{Y}$ to latent variables, where the probabilistic treatment helps handle issues such as over-fitting. BEPCA is a fully probabilistic treatment of EPCA (Fig. 1(c)). It introduces an additional layer of probabilistic structure as the prior of $\mathbf{Y}$. BEPCA differs from the proposed model in that the prior of each PC $\mathbf{w}$ is assumed to be the conjugate prior of the chosen likelihood $p(\mathbf{x}|\boldsymbol{\theta})$. Conjugate prior provides mathematical convenience; at the cost of this convenience, SePCA achieves (i) forgoing hyper-parameter tuning, (ii) explicitly yielding the effective number of necessary PCs and (iii) simplifying implementation by using the prior of Gaussian distribution for different likelihood functions.

In the rest of this section, we provide details on fitting the SePCA model in an EM framework Hunter and Lange (2004); Gormley and Murphy (2008).

3.2 Parameter estimation

The number of effective PCs is determined by estimating the continuous parameter $\boldsymbol{\alpha}$. The marginal likelihood function of $\boldsymbol{\alpha}$ for observations $\mathbf{X}$ is

$$\begin{aligned} p(\mathbf{X}|\boldsymbol{\alpha}) &= \int_{\mathbf{W},\mathbf{Y}} p(\mathbf{X},\mathbf{Y},\mathbf{W}|\boldsymbol{\alpha})\mathrm{d}\mathbf{Y}\mathrm{d}\mathbf{W} \qquad (7)\\ &= \int_{\mathbf{W},\mathbf{Y}} p(\mathbf{X}|\mathbf{Y},\mathbf{W})p(\mathbf{Y})p(\mathbf{W}|\boldsymbol{\alpha})\mathrm{d}\mathbf{Y}\mathrm{d}\mathbf{W}. \end{aligned}$$

In order to maximise $p(\mathbf{X}|\boldsymbol{\alpha})$, we consider it as marginalisation over $\mathbf{W}$

$$p(\mathbf{X}|\boldsymbol{\alpha}) = \int_{\mathbf{W}} p(\mathbf{X}|\mathbf{W})p(\mathbf{W}|\boldsymbol{\alpha})\mathrm{d}\mathbf{W}, \qquad (8)$$

where

$$p(\mathbf{X}|\mathbf{W}) = \int_{\mathbf{Y}} p(\mathbf{X}|\mathbf{Y},\mathbf{W})p(\mathbf{Y})\mathrm{d}\mathbf{Y}. \qquad (9)$$

Then the derivative of $p(\mathbf{X}|\boldsymbol{\alpha})$ w.r.t. each α_j is

$$\begin{aligned} \frac{\partial p(\mathbf{X}|\boldsymbol{\alpha})}{\partial \alpha_j} &= \int_{\mathbf{W}} p(\mathbf{X}|\mathbf{W})p(\mathbf{W}|\boldsymbol{\alpha})\left[\frac{D}{2\alpha_j} - \frac{\|\mathbf{w}_j\|_2^2}{2}\right]\mathrm{d}\mathbf{W}\\ = \frac{p(\mathbf{X}|\boldsymbol{\alpha})}{2}&\left[\frac{D}{\alpha_j} - \int_{\mathbf{W}} p(\mathbf{W}|\mathbf{X},\boldsymbol{\alpha})\|\mathbf{w}_j\|_2^2\mathrm{d}\mathbf{W}\right], \qquad (10) \end{aligned}$$

where $p(\mathbf{W}|\mathbf{X},\boldsymbol{\alpha}) = p(\mathbf{X}|\mathbf{W})p(\mathbf{W}|\boldsymbol{\alpha})/p(\mathbf{X}|\boldsymbol{\alpha})$. Setting this derivative to zero, the estimation of $\alpha_j \in \boldsymbol{\alpha}$ is as follows,

$$\alpha_j = \frac{D}{\mathbb{E}_{\mathbf{W}|\mathbf{X}}\left[\|\mathbf{w}_j\|_2^2\right]}, \qquad (11)$$

where the expectation is taken over the posterior distribution $p(\mathbf{W}|\mathbf{X},\boldsymbol{\alpha})$. Note that the evaluation of (11) is iterative, because the posterior of $\mathbf{W}$ is itself affected by $\boldsymbol{\alpha}$ (*cf.* the integrand in (8)).

If the number of observations is sufficiently large and the posterior of $\mathbf{W}$ is peaked, for evaluating (11), it is sufficient to make a further simplification by replacing the expectation with the point estimate. We then approximate the update of α_j with

$$\alpha_j = \frac{D}{\|\mathbf{w}_j^{\mathrm{MP}}\|_2^2}, \qquad (12)$$

where $\mathbf{w}_j^{\mathrm{MP}}$ is the MAP estimation of $\mathbf{w}_j$. Therefore the problem of estimating the model parameter is reformulated as an optimisation w.r.t. $\mathbf{W}$.

3.3 Inference

Inference on W Consider the posterior distribution of $\mathbf{W}$ given $\mathbf{X}$ and $\boldsymbol{\alpha}$

$$p(\mathbf{W}|\mathbf{X},\boldsymbol{\alpha}) \propto p(\mathbf{X}|\mathbf{W})p(\mathbf{W}|\boldsymbol{\alpha}). \qquad (13)$$

To maximise $p(\mathbf{W}|\mathbf{X},\boldsymbol{\alpha})$, we can instead maximise $\log p(\mathbf{X}|\mathbf{W}) + \log p(\mathbf{W}|\boldsymbol{\alpha})$. However, it is difficult to carry out this optimisation directly. The term $\log p(\mathbf{X}|\mathbf{W})$ in the objective function cannot be represented explicitly in terms of $\mathbf{W}$, because $p(\mathbf{X}|\mathbf{Y},\mathbf{W})$ contains a term of some general exponential family density (*cf.* (9)). In this paper, we approximate the log-marginal likelihood with a lower bound and adopt the expectation-maximisation (EM) scheme for optimisation.

The lower bound is derived as follows. For an arbitrary distribution $q(\mathbf{Y})$ on $\mathbf{Y}$, according to Jensen's inequality, we have

$$\begin{aligned} \log p(\mathbf{X}|\mathbf{W}) &= \int_{\mathbf{Y}} q(\mathbf{Y})\log p(\mathbf{X}|\mathbf{W})\mathrm{d}\mathbf{Y}\\ &= \mathcal{L}_{q_{\mathbf{Y}}}(\mathbf{W}) + \mathrm{KL}(q\|p), \qquad (14) \end{aligned}$$

where $\mathrm{KL}(q\|p)$ is the Kullback-Leibler divergence between $q(\mathbf{Y})$ and the posterior distribution $p(\mathbf{Y}|\mathbf{W},\mathbf{X})$, and $\mathcal{L}_{q_{\mathbf{Y}}}(\mathbf{W})$ is the lower bound. It is defined by

$$\mathcal{L}_{q_{\mathbf{Y}}}(\mathbf{W}) = \int_{\mathbf{Y}} q(\mathbf{Y})\log p(\mathbf{X},\mathbf{Y}|\mathbf{W})\mathrm{d}\mathbf{Y} + \mathbf{H}_{q_{\mathbf{Y}}}, \qquad (15)$$

where $\mathbf{H}_{q_{\mathbf{Y}}}$ is the entropy of q and independent of $\mathbf{W}$.

The mode of the posterior distribution of $\mathbf{W}$ can be approximated by the $\mathbf{W}$ that maximises $\mathcal{L}_{q_{\mathbf{Y}}}(\mathbf{W}) + \log p(\mathbf{W}|\boldsymbol{\alpha})$

$$\mathbf{W}^{\mathrm{MP}} \approx \underset{\mathbf{W}}{\mathrm{argmax}}\left\{\mathcal{L}_{q_{\mathbf{Y}}}(\mathbf{W}) + \log p(\mathbf{W}|\boldsymbol{\alpha})\right\}. \qquad (16)$$

Since $\mathcal{L}_{q_{\mathbf{Y}}}(\mathbf{W})$ contains $\log p(\mathbf{X},\mathbf{Y}|\mathbf{W})$, which generally involves non-trivial transformations of $\mathbf{W}$, we then adopt gradient-based maximisation to update $\mathbf{W}$. The gradient of the maximisation objective in (16) w.r.t. $\mathbf{W}$ is

$$\frac{\partial(\mathcal{L}_{q_{\mathbf{Y}}}(\mathbf{W}) + \log p(\mathbf{W}|\boldsymbol{\alpha}))}{\partial \mathbf{W}} = \frac{\partial \mathcal{L}_{q_{\mathbf{Y}}}(\mathbf{W})}{\partial \mathbf{W}} - \mathbf{W}\mathrm{diag}[\boldsymbol{\alpha}], \qquad (17)$$

where $\mathrm{diag}[\boldsymbol{\alpha}]$ represents a $d \times d$ diagonal matrix with diagonal elements $\alpha_1, \ldots, \alpha_d$. For $\partial \mathcal{L}_{q_{\mathbf{Y}}}(\mathbf{W})/\partial \mathbf{W}$, substitute $p(\mathbf{X},\mathbf{Y}|\mathbf{W}) = p(\mathbf{X}|\mathbf{Y},\mathbf{W})p(\mathbf{Y})$ into the integrand in (15) and refer to (6). We have (omitting the constant term $h(\mathbf{x})$)

$$\begin{aligned} \frac{\partial \mathcal{L}_{q_{\mathbf{Y}}}(\mathbf{W})}{\partial \mathbf{W}} &= \int_{\mathbf{Y}} q(\mathbf{Y})\frac{\partial \log p(\mathbf{X},\mathbf{Y}|\mathbf{W})}{\partial \mathbf{W}}\mathrm{d}\mathbf{Y}\\ &= \int_{\mathbf{Y}} q(\mathbf{Y})\left[g'(\boldsymbol{\Theta})\mathbf{Y}^T + \mathbf{X}\mathbf{Y}^T\right]\mathrm{d}\mathbf{Y}\\ &= \mathbb{E}_{\mathbf{Y}}\left[g'(\boldsymbol{\Theta})\mathbf{Y}^T\right] + \mathbf{X}\mathbb{E}_{\mathbf{Y}}\left[\mathbf{Y}\right]^T, \qquad (18) \end{aligned}$$

where $\mathbf{\Theta} = \mathbf{WY}$. Note that for an array (vector or matrix) of inputs, $g(\cdot)$ in (6) represents the sum of element-wise evaluation, $g(\mathbf{\Theta}) = \sum_{i,j} g(\theta_{i,j})$. Then $g'(\mathbf{\Theta})$ in (18) is a matrix, where the (i,j)-th entry is $g'(\theta_{i,j})$. $\mathbb{E}_{\mathbf{Y}}$ is the expectation over the distribution $q(\mathbf{Y})$. In the expectation step, we match $q(\mathbf{Y})$ to the posterior distribution $p(\mathbf{Y}|\mathbf{X},\mathbf{W})$. Details are given below.

Inference on $\mathbf{Y}$ For $q(\mathbf{Y})$, it is necessary to have (i) its representation and (ii) a method of computing the expectations over it. The simplest method is to find the MAP estimation of $\mathbf{Y}$ by maximising $\log p(\mathbf{X},\mathbf{Y}|\mathbf{W})$. The expectation in (18) can then be approximated in the way similar to the simplification of the inference for $\mathbf{W}$. Thus in the simple scheme, the inference on $\mathbf{W}$ and $\mathbf{Y}$ is an alternating optimisation algorithm. Algorithm 1 summarises the key steps to which we add the following remarks. First, the two steps of the alternating optimisation are both convex. However, the joint optimisation is not convex in general. Second, if we carry out the optimisation sequentially[1], the algorithm can be deemed as a regularised version of that of EPCA, which is itself reminiscent of a classical method (Csiszár and Tusnády, 1984) of solving conventional PCA (Collins et al., 2002). Thus, the algorithms of (Csiszár and Tusnády, 1984) and EPCA, which are both deterministic, have a new interpretation as solving for the MAP estimation of a probabilistic model.

Algorithm 1: Inference on the latent variables $\mathbf{W}$ and $\mathbf{Y}$

```
Input: $\mathbf{X}$, $\boldsymbol{\alpha}$, $\mathbf{W}_{\text{Init}}$
Output: $\mathbf{W}^{\text{MP}}$, $\mathbf{Y}^{\text{MP}}$
$\mathbf{W}^{\text{MP}} \leftarrow \mathbf{W}_{\text{Init}}$
while not converge do
    $\mathbf{Y}^{\text{MP}} \leftarrow \text{argmax}_{\mathbf{Y}} \left(\log p(\mathbf{X}|\mathbf{W}^{\text{MP}},\mathbf{Y}) + \log p(\mathbf{Y})\right)$
    $\mathbf{W}^{\text{MP}} \leftarrow \text{argmax}_{\mathbf{W}} \left(\mathcal{L}_{q_{\mathbf{Y}}}(\mathbf{W}) + \log p(\mathbf{W}|\boldsymbol{\alpha})\right)$
end
```

4 Mixtures of SePCA

The probabilistic characteristic of SePCA grants us a natural mechanism to build a mixture of local models to deal with complex data structures.

Consider M local SePCA models with the mixing proportion parameters $\mathbf{\Pi} = \{\pi_1,\ldots,\pi_M\}$, $\sum_m \pi_m = 1$ and $\pi_m > 0$, we have

$$p(\mathbf{x}) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}|m), \tag{19}$$

where $p(\mathbf{x}|m)$ is the evidence of a local model m for observing $\mathbf{x}$. In our model, we have $p(\mathbf{x}|m) = p(\mathbf{x}|\boldsymbol{\alpha}^{(m)})$ (*cf.* (7)), where the superscript $^{(m)}$ indicates the local SePCA model. Then the marginal log-likelihood function of a mixture model is

$$\begin{aligned}\mathbf{L}(\mathbf{A},\mathbf{\Pi}) &= \sum_{n=1}^{N} \log p(\mathbf{x}_n|\mathbf{A},\mathbf{\Pi}) \\ &= \sum_{n=1}^{N} \log \left\{ \sum_{\mathbf{z}_n} \prod_{m=1}^{M} \left[\pi_m p(\mathbf{x}_n|\boldsymbol{\alpha}^{(m)})\right]^{\mathbf{z}_{n,m}} \right\},\end{aligned} \tag{20}$$

where $\mathbf{A} = \{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(M)}\}$ and $\mathbf{z}_n$ is an M-dimensional binary latent variable for each sample, $\mathbf{z}_{n,m} = 1$ when the n-th sample is associated with the m-th local SePCA model and $\mathbf{z}_{n,m} = 0$, otherwise. Therefore for $\mathbf{z}_{n,m} = 1$, $p(\mathbf{z}_n|\mathbf{\Pi}) = \pi_m$ and $p(\mathbf{x}_n|\mathbf{z}_n,\mathbf{A}) = p(\mathbf{x}_n|\boldsymbol{\alpha}^{(m)})$.

The mixture model is fitted by using an EM algorithm. According to Jensen's inequality, a lower bound of (20) is

$$\begin{aligned}\mathbf{L}(\mathbf{A},\mathbf{\Pi}) &\geq \mathcal{L}_{q_{\mathbf{Z}}}(\mathbf{A},\mathbf{\Pi}) + \mathbf{H}_{q_{\mathbf{z}}} \\ &= \sum_{n=1}^{N} \sum_{\mathbf{z}_n} q(\mathbf{z}_n) \log\{p(\mathbf{x}_n,\mathbf{z}_n|\mathbf{A},\mathbf{\Pi})\} + \mathbf{H}_{q_{\mathbf{z}}}\end{aligned} \tag{21}$$

for an arbitrary distribution $q(\mathbf{z}_n)$. The $\mathbf{H}_{q_{\mathbf{z}}}$ is the entropy of $q(\mathbf{z}_n)$. We can estimate $\mathbf{A}$ and $\mathbf{\Pi}$ by maximising $\mathcal{L}_{q_{\mathbf{Z}}}(\mathbf{A},\mathbf{\Pi})$.

1. Estimation of $\mathbf{\Pi}$ is similar to the corresponding procedure used in fitting Gaussian mixtures (*e.g.* Bishop, 2007, Ch. 9 and 10). We find $\mathbf{\Pi} = \text{argmax}_{\mathbf{\Pi}} \mathcal{L}_{q_{\mathbf{Z}}}(\mathbf{A},\mathbf{\Pi})$, subject to $\sum_m \pi_m = 1$ and $\pi_m > 0$. Based on the Lagrangian method, the estimate for $\pi_k \in \mathbf{\Pi}$ is

$$\pi_k = \frac{\sum_n R_{n,k}}{\sum_{m,n} R_{n,m}} = \frac{R_k}{\sum_m R_m}, \tag{22}$$

where the matrix $\mathbf{R}$ represents $q(\mathbf{z})$, $R_{n,m}$ is the probability $Prob\{\mathbf{z}_{n,m} = 1\}$, according to $q(\mathbf{z})$, and $R_m = \sum_n R_{n,m}$.

2. Estimation of $\mathbf{A}$ consists of updating $\boldsymbol{\alpha}^{(k)}$ for each local SePCA model. For the k-th local model, updating $\alpha_j^{(k)}$ is analogous to the re-estimation of $\boldsymbol{\alpha}$ in (11) for a single model

$$\alpha_j^{(k)} = \frac{D}{\|\mathbf{w}_j^{(k),\text{MP}}\|_2^2}, \tag{23}$$

where $\mathbf{w}_j^{(k),\text{MP}}$ is the j-th column of $\mathbf{W}^{(k),\text{MP}}$, which is the MAP estimation of the PCs in the k-th local model. As in (11), equation (23) is an approximated expectation over the posterior of $\mathbf{W}^{(k)}$. The posterior mode is estimated by maximising the weighted log-likelihood

$$\begin{aligned}\mathbf{W}^{\text{(k),MP}} = \underset{\mathbf{W}^{(k)}}{\text{argmax}} \Big\{ &\log p(\mathbf{W}^{(k)}|\boldsymbol{\alpha}^{(k)}) \\ &+ \sum_n \frac{R_{n,k}}{R_k} \log p(\mathbf{x}_n|\mathbf{W}^{(k)}) \Big\}.\end{aligned} \tag{24}$$

[1] Introduce an inner loop, in each step, the latent variables responsible for one latent dimension are optimised, *i.e.* Loop over $j = 1,\ldots,d$, for each j, alternately optimise $\mathbf{w}_j$ and $\mathbf{y}_1(j),\cdots,\mathbf{y}_n(j)$.

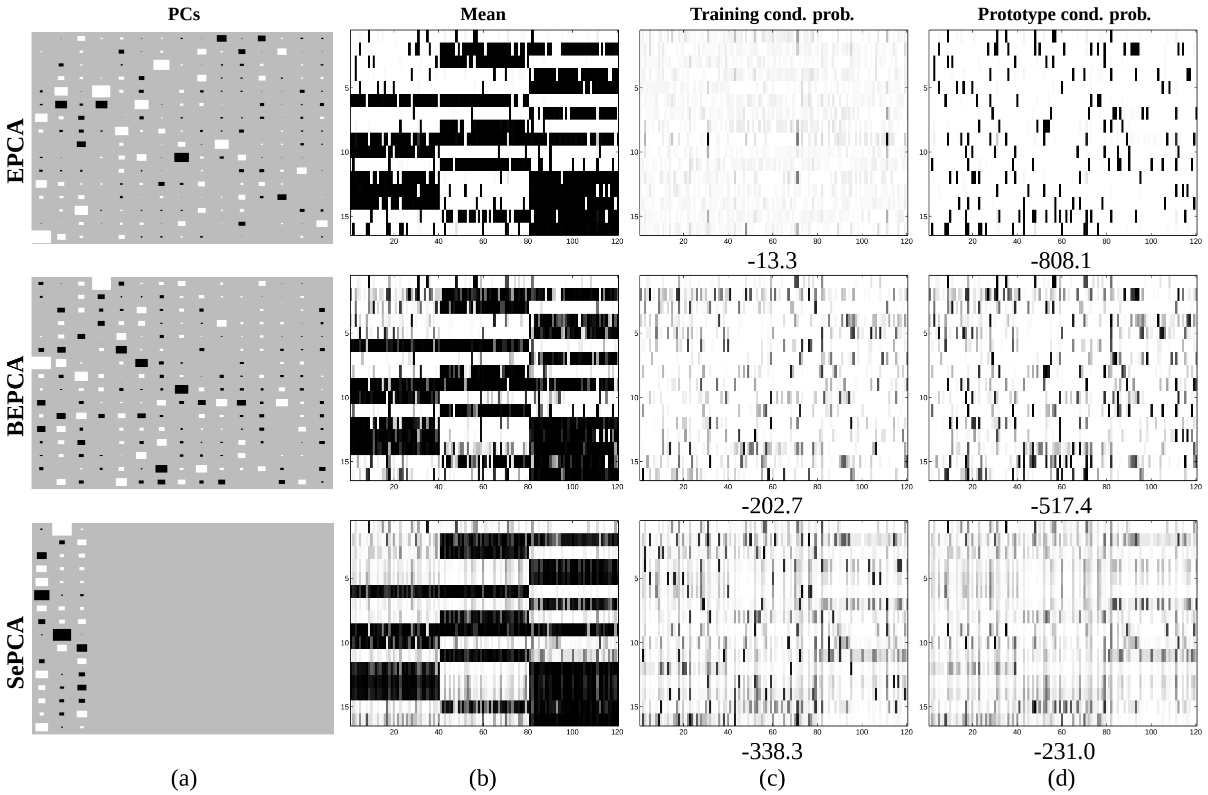


Figure 3: Models learnt by EPCA, BEPCA and SePCA
(a) Hinton diagrams of PCs; **(b)** Mean parameters; **(c)** Conditional probability of the training data given the learnt mean; **(d)** Conditional probability of the prototypes (*cf.* Fig. 2(a)) given the learnt mean. In the Hinton diagrams in (a), black and white indicates negative and positive elements, respectively; and the box size indicates element magnitude. In (c-d), the gray level indicates numerical values in $[0, 1]$, where black indicates 0 and white indicates 1.

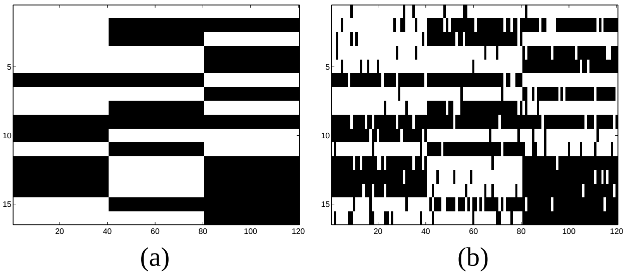


Figure 2: Synthetic binary vector data set
(a) 3 prototype binary vectors, 40 copies of each prototype; **(b)** training data generated by randomly flipping bits in (a) with probability 0.1. Each image contains 16 rows and 120 columns of binary bits, where each column represents one data vector (clearer in (b)).

3. In the E-step, we update $q(\mathbf{z})$ and let it match the true posterior of the "membership" of each sample

$$R_{n,k} = \frac{\pi_k p(\mathbf{x}_n|k)}{p(\mathbf{x}_n)}, \tag{25}$$

where $p(\mathbf{x}_n)$ is given in (19) and $p(\mathbf{x}_n|k)$ is computed using the updated mixture parameters in the M-step.

5 Experiments

We first test the SePCA on the synthetic data used in Mohamed et al. (2009) and Tipping and Bishop (1999). Three 16-D binary prototype vectors are generated with each bit drawn from $\{0, 1\}$ with equal probability. The data consists of 120 samples, 40 from each prototype. Then each bit is flipped with probability 0.1. Figure 2 shows one data set from the randomly generated prototypes in (a), and (b) shows the data set after the noises have been added. The samples in (b) will be used to train the model, whose those in (a) will used to assess the fitted models.

We apply EPCA (Collins et al., 2002), BEPCA (Mohamed et al., 2009), and SePCA on this binary data set. The specific exponential family distribution is chosen to be Bernoulli, and the latent dimension is set to $d = 15$.

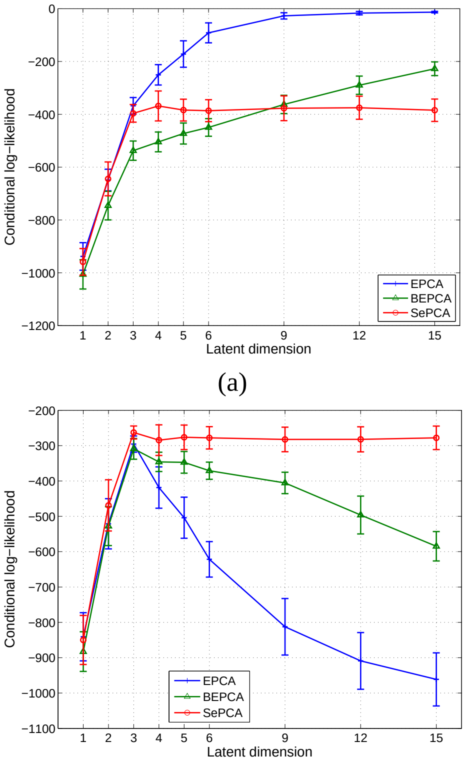
Figure 3 shows the results of fitting models. In the figure, from top to bottom, rows represent the results of EPCA, BEPCA and SePCA respectively. Column (a) shows the fitted PCs in Hinton diagrams. Column (b) shows the fitted mean parameters of the Bernoulli distribution. In the images, each pixel corresponds to a bit in the data, and its brightness represents how likely that the bit is 1. Column (c) represents matching the mean parameters in (b) against the noisy training data, *i.e.* the likelihood at each bit of the fitted parameter for the training data. The number below each sub-figure is the sum of the log-likelihood at each pixel, $\sum_{p,q} \log p(\mathbf{X}_{p,q}|\mu_{p,q})$. This sum is always negative, and the closer to zero it is, the better the match. Column (d), in contrast to column (c), represents matching mean parameters against the prototypes.

The results in column (c) indicate that EPCA fits the training data bit by bit precisely and achieved high likelihood. However, column (d) shows that EPCA has been adapted to the noise seriously and resulted in considerable over-fitting. On the contrary, SePCA effectively discovered the latent dimensionality (*cf.* column (a)). Roughly speaking, by using limited PCs, SePCA fitted to the prototypes that mainly underlie observations, and avoided fitting to the noise. The performance of BEPCA is in between. It is less prone to the over-fitting problem because of the prior. However, it can not explicitly reveal the latent dimensionality; and cannot automatically control the model complexity.

We trained models by using different estimated latent dimensions. For each of the fitted models, the log-likelihood of both the noisy training data and the prototypes are calculated. Figure 4 compares results. The error bars are obtained by running the experiment 10 times on randomly generated data sets. EPCA tends to over-fit when more than three PCs (latent dimension) are provided. The model complexity and over-fitting mildly increased in BEPCA with the estimated latent dimension. On the contrary, SePCA discovered the true latent dimension and rejected the surplus dimensions, and thus it is immune to over-fitting caused by the tentative choice of the latent dimension.

In the second experiment, the USPS hand-written digits are used to test the mixture model. We use 600 binary images of the digits 2, 3 and 4 to train 3 mixtures. The size of the images is 16×16. Since it is inconvenient to specify mixtures in the parameter space, we initialise $q(\mathbf{z}_n)$ with $R_{n,\cdot} = \{0.34, 0.33, 0.33\}$ (*cf.* (22)) and start from the M-step.

Figure 5 shows the results of the fitted PCs. The first and second rows are the first and second PCs of each mixture respectively. The PCs in (a) are those from the first M-step. They appear identical and indicate considerable confusion and overlapping between the local models. Thus there is no evidence of a clear head start (0.34 over 0.33). The fitted PCs of the mixtures are shown in (b). Figure 6 shows several images generated by each local SePCA model. In this case, each fitted local model has captured the underlying factors that are responsible for one digit.



(a)

(b)

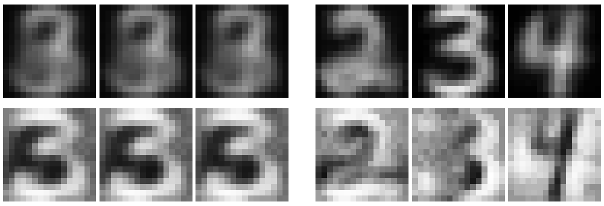
Figure 4: Conditional log-likelihood versus latent dimensions
(a): conditional probability of the training data given reconstructed parameters; **(b)**: conditional probability of the prototypes, which is the information to be learned from the training data (*cf.* Fig. 2(a)). The conditional probability reflects the match between the data and the model. It indicates the model has been fitted to the noise, if the training data have a high probability and the prototype data have a low one.

6 Conclusion and Discussions

In this paper, we have proposed SePCA, a new family of generative models. The proposed model handles general type observations with exponential family distributions that are parameterised by the latent variable of PCs and low dimensional representations. By introducing automatic relevance determination (ARD) to SePCA, the model automatically determines the appropriate number of latent variables.

It is enlightening to consider our method from the viewpoint of intrinsic dimension estimation. PCA-based projection methods often treat the problem as preserving or discarding PCs (Camastra, 2004), and ARD is a systematic approach. ARD employs a Gaussian prior for PCs.

This approach contradicts traditional PCA on some practical data types, *e.g.* integers or binary values. This is



(a) (b)

Figure 5: PCs of the local SePCA models in a trained mixture model
(a): First two (in rows) local PCs resulted in the first iteration; **(b)**: PCs in the trained mixture model. The images of the PCs show that three local SePCA models in the mixture are fit to digits 2, 3 and 4, respectively.

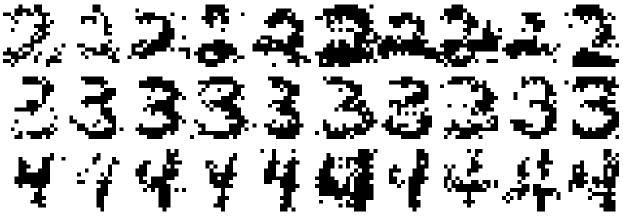


Figure 6: Generated samples for hand-written digits
Each row of the images are randomly drawn following the generative process of one local SePCA in the trained mixture model. The generated images visually verify that local models are duly fitted to three digits.

because PCA finds PCs in the sample space, and Gaussian is *not* a distribution in such sample spaces. However, the exponential family distributions provide a link between general type observations and continuous latent variables, on which an ARD prior can be imposed. Therefore, based on this link, we reach a principled method of determining the intrinsic dimension of general distributions regardless of the type of observations.

Acknowledgements

The project was fully supported by the Nanyang Technological University SUG Grant (under project number M58020010).

References

Bishop, C. M., 1999a. Bayesian PCA. In: Proceedings of the 1998 conference on Advances in neural information processing systems.

Bishop, C. M., 1999b. Variational principal components. In: Proceedings of the Ninth International Conference on Artificial Neural Networks.

Bishop, C. M., 2007. Pattern Recognition and Machine Learning. Springer.

Camastra, F., 2004. Data dimensionality estimation methods: A survey. Pattern Recognition 36, 2945–2954.

Collins, M., Dasgupta, S., Schapire, R. E., 2002. A generalization of principal component analysis to the exponential family. In: Proceedings of the 2001 conference on Advances in neural information processing systems.

Csiszár, I., Tusnády, G., 1984. Information geometry and alternating minimization procedures. Statistics and Decisions, Supplement Issue 1, 205–237.

Gormley, I. C., Murphy, T. B., 2008. A mixture of experts model for rank data with applications in election studies. Annals of Applied Statistics 2 (4), 1452–1477.

Guo, Y., 2009. Supervised exponential family principal component analysis via convex optimization. In: Proceedings of the 2008 conference on Advances in neural information processing systems.

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

Hunter, D., Lange, K., 2004. A tutorial on mm algorithms. The American Statistician 58, 30–37.

Jolliffe, I. T., 1986. Principal Component Analysis. Springer-Verlag.

MacKay, D. J. C., 1995. Probable networks and plausible predictions – a reivew of practical bayesian methods for supervised neural networks. Network: Computation in Neural Systems 6 (3), 469–505.

McCullagh, P., Nelder, J. A., 1989. Generalized Linear Models. Vol. 37 of Monographs on statistics and applied probability. CRC Press.

Mohamed, S., Heller, K., Ghahramani, Z., 2009. Bayesian exponential family PCA. In: Proceedings of the 2008 conference on Advances in neural information processing systems.

Sajama, Orlitsky, A., 2005. Semi-parametric exponential family PCA. In: Proceedings of the 2004 conference on Advances in neural information processing systems.

Tipping, M. E., Bishop, C. M., 1999. Mixtures of probabilistic principal component analyzers. Neural Computation 11 (2), 443–482.

Turk, M., Pentland, A., 1991. Eigenfaces for recognition. Journal of Cognitive Neuroscience 3 (1), 71–86.

Welling, M., Chemudugunta, C., Sutter, N., 2008. Deterministic latent variable models and their pitfalls. In: Proceedings of SIAM Conference on Data Mining.