Simple Exponential Family PCA

Jun Li and Dacheng Tao

***Abstract*—Principal component analysis (PCA) is a widely used model for dimensionality reduction. In this paper, we address the problem of determining the intrinsic dimensionality of a general type data population by selecting the number of principal components for a generalised PCA model. In particular, we propose a generalised Bayesian PCA model, which deals with general type data by employing exponential family distributions. Model selection is realised by empirical Bayesian inference of the model. We name the model as simple exponential family PCA (SePCA), since it embraces both the principal of using a simple model for data representation, and the practice of using a simplified computational procedure for the inference. Our analysis shows that the empirical Bayesian inference in SePCA formally realises an intuitive criterion for PCA model selection: a preserved principal component must sufficiently correlate to data variance that is uncorrelated to the other principal components. Experiments on synthetic and real data sets demonstrate effectiveness of SePCA and exemplify its characteristics for model selection.**

***Index Terms*—Exponential family PCA, Automatic Relevance Determination, Dimensionality Reduction**

I. Introduction

Principal component analysis (PCA) [1] represents a family of models that seeks reduced dimension representation of data and has arguably become the default pre-processing step for subsequent analysis tasks in broad application areas [2, 3, 4]. One important choice in applying of PCA is the number of factors to use, which determines the model complexity. A good choice should allow the model to fit the data well and in the meanwhile avoid over-fitting. The number can be set manually according to domain knowledge. One can also prescribe the portion of preserved variations to indirectly indicate the number. Another widely used trick is to employ cross-validation on held-out data. Despite the simplicity, these *ad hoc* strategies are task-specific, and can hardly be justified systematically.

Another practical concern on PCA is the assumption that all the observations are real-valued and constitute a Euclidean vector space. From this perspective, PCA reconstructs the observed data in a subspace. The model is learned by minimising the reconstruction errors, which are measured in Euclidean distance defined in the space of observations. This Euclidean assumption on the observations may be inappropriate, for instance given the contexts of some practical tasks, only binary, integer or non-negative values can be appropriate[5, 6, 7].

The two issues discussed above can both be addressed systematically by adopting the probabilistic interpretation of PCA. On one hand, probability provides a suitable language to unify the description of different models, as well as various principles for model selection. For example, when PCA is reformulated as the max likelihood estimation to a probabilistic model based on Gaussian distribution [8, 9], model selection can be assisted by Bayesian inference on the model [10]. On the other hand, probability provides a common notation regardless the specific form of the objects of interest. Particularly, if the quantities to be modelled are not real-valued, general exponential family distributions can be adopted [11]. In [5], probabilistic PCA is equipped with exponential family distributions for representing general type (non-real-valued) observations. However, less study has been devoted to the model selection problem for the model families handling non-real-valued observations. For example, the Bayesian computation in [10] depends on the Gaussian model and thus is not directly applicable to data of general types.

In this paper, we propose to employ ARD in a probabilistic formulation of exponential family PCA and show *maximum a posteriori* (MAP) for Bayesian learning. The Bayesian learning facilitates automatic decision of the minimum number of factors to represent data of general types. We name the model as simple exponential family PCA, or SePCA for short. Specifically, SePCA treats both the factors and the coefficients as random variables and employs exponential family distributions to define the likelihood function of the observations given the factors and the coefficients. The exponential family likelihood functions link real factors and coefficients to observations of general types. This link makes it possible to apply tools designed for real variables to control the complexity of a model representing general type data. We impose a Gaussian prior on the real coefficients. The variance of the Gaussian is controlled by automatic relevance determination (ARD) [12], which trims the model by pruning factors with low correlations to the observations.

ARD is an empirical Bayes method and can be implemented easily in conjunction with MAP inference of the factor/coefficient variables. Besides the mathematical convenience, the computation of MAP in an exponential family formulation helps understand how ARD controls the model complexity. We demonstrate that ARD formally requires each factor to contribute to explaining the observations to avoid being pruned.

II. Background

Probabilistic formulation advances PCA by improving efficiency, allowing mixture models, and addressing problems of small sample size and missing values [13, 9, 8, 14, 15]. From a stochastic perspective, the data are considered as draws from independent Gaussians centralised in a subspace [9]. PCA

This work was supported by the Australian Research Council Discovery Project DP-120103730. Jun Li and Dacheng Tao are with the Centre for Quantum Computation and Intelligent Systems, University of Technology, Sydney, NSW, Australia, 2007. Email:{jun.li, dacheng.tao}@uts.edu.au

finds the centres of the Gaussians by max likelihood estimation. The probabilistic treatment facilitates the application of statistical model selection and comparison. For example, by counting individual coefficients as unknown parameters, standard model selection criteria can be evaluated for PCA, including Akaike information criterion (AIC) [16], Bayesian information criterion (BIC) [17] and minimum description length (MDL) [18]. Although the standard criteria are theoretically justifiable, it is often preferable to develop methods that are specialised for controlling the complexity of linear factor models. For example, mixture models of coefficients encourage sharing of the coefficients and reduce the effective number of parameters [19]. In [10], Bayesian model selection has been developed for PCA by integrating over the coefficients on a Stiefel manifold. A dimension estimation algorithm simulating activation and inhibition mechanism of the biological neurons has been proposed for an online setting in [20]. In [21], the linear model selection problem has been addressed in a supervised setting. Recently, a dimension reduction method has been derived from both maximising classification margin and boosting data independency [22], and therefore can be applied for both the supervised and unsupervised problems. However, these specialised methods depend on the Gaussian likelihood of the data, i.e. the observations are real-valued.

Employing the *exponential family distributions* for the observation likelihood is a well-established technique, which allows the models to take into account for general types of data. For example, if the data are observed in binary scales, a Bernoulli model is more suited than a Gaussian one. In the case of linear regression models, exponential family distributions link the predicted variables and observations. The technique has been extensively discussed in [11]. For factor models, however, if non-Gaussian observation likelihood is used, learning the model will not be trivial because of the unknown coefficients (as opposed to the design matrix in a regression model, which is generally given). In [6], the coefficients are considered as model parameters and optimised by an EM algorithm. In [5], a connection to PCA has been established by viewing learning the generalised factor model as minimising the Bregman distance between the predictions and the observed data. The model is named exponential family PCA (EPCA) for this connection. Alternating optimisation is used to compute the factors and coefficients in EPCA.

The divided treatment to observations and parameters in EPCA provides the foundation of our technique of choosing the proper number of factors for a generalised latent factor model. The model uses the latent factors to represent a population of real-valued parameters; and the exponential family likelihood function connects parameters to observations. Therefore, although the model no longer takes the convenient form of a joint Gaussian, the problem leads itself to tools that deal with real-valued coefficient vectors of each factor. Particularly in this study, we employ automatic relevance determination (ARD). ARD is developed to control the complexity of neural networks [23, 12], where constrained Gaussian priors are imposed on the weights of the connections between the explanatory units and the hidden units. There is well-established connection between neural networks and feature extraction and dimension reduction models[24]. ARD has been employed to define the prior of the coefficients in a Bayesian model of PCA [25, 26]. A key motivation for the current study is the observation that in neural networks, the hidden layer separates the explanatory and the predicted variables. Therefore, the ARD prior is unaffected by how the predicted variables are modelled. Regarding to our problem, this provides the convenience that ARD readily works with general observation likelihood functions that extend beyond Gaussian to the exponential family.

In particular, ARD states that the coefficients associated with each factor follow a zero-mean isotropic Gaussian prior, where the variance of that Gaussian distribution is learned automatically. When the relevance of a factor to the observations is low, the corresponding variance of the corresponding coefficients decays to zero in the learning process, and the factor becomes ineffective [23, 12]. Since the prior variance of the coefficients is deduced from the data, the computation is referred to as *empirical Bayes* [27] (also as *evidence maximisation* or *type II maximum likelihood* in the literature). In this way, our utilisation of ARD is connected to the rich research on Bayesian methodology for limiting model complexity [28, 29, 30, 31, 32]. Given the design matrix, in [28], the scheme of ARD is adapted for the weights in a Bayesian formulation of kernel regression / classification models. The result is a set of most relevant training samples for the task. For the special case of ARD in [28], Wipf and Nagarajan [29] established the equivalence between learning the variance of the weights and performing MAP inference directly in the space of the weights using a sparsity-promoting prior. In [30], a detailed discussion is given, and it has been shown that the penalty on the weights can be extended to a certain class of non-factorial cost functions that are dependent on both the noise and the design matrix and can be generalised beyond the original Bayesian model.

For latent factor models, sparse learning has also been well studied. Seeger et al. [33] tackle the inference problem in a Bayesian sparse regression model and address the problem of learning a design matrix to extract most information from experimental observations. In [34], learning the coefficients for the latent factors is explicitly transformed to a regression problem; and the LARS method [35] is employed to produce a sparse solution. In [36], sparse PCA is directly relaxed as a convex optimisation problem and solved using semidefinite programming. Archambeau and Bach [37] develop ARD by proposing a prior of infinite mixture of scaled Gaussians. The prior is imposed independently (element-wisely) on the coefficients for spare PCA and sparse canonical correlation analysis (CCA). The computational aspects of the Bayesian sparse learning are treated extensively in [38].

Most sparse models we have discussed so far utilise Gaussian observation model to facilitate computation. More relevant to our interested problem, there is a series of works on regularised learning of models dealing with general types of data. Sparse penalty ℓ_1 is used for logistic regression in [39], and it is applied to obtain the sparse solution in learning EPCA [31,

Sec 3.2] (and [40] for a semi-supervised setting). In [41], a hierarchical Bayesian extension to EPCA has been proposed. The spike-and-slab prior is employed in a Bayesian model that learns sparse coefficients of EPCA in [31, Sec 3.3][32]. It is worth noting that the sparse prior on the coefficients is also used by [42] for a non-parametric Bayesian factor analysis model with Gaussian noise. In [42], the prior is defined on infinite latent features, where the number of factors is inferred from the learning of the model using a stochastic process [43]. In these works, over-fitting in the factor models is effectively avoided by penalising the non-zero coefficients or by not being over-confident about the point estimates. Because of the regulations, good prediction performance can be achieved by those models that are apparently over-complex for the data. The proposed ARD prior also regulates model complexity. Moreover, ARD provides an explicit number of latent factors, which can be useful for applications such as dimension reduction or data visualisation. The initial idea appears in our earlier paper [44]. The present paper further extends the basic scheme extensively.

III. Simple Exponential Family PCA

We consider N observed samples of d variables, which is represented by a $[d \times N]$ matrix $\mathbf{X}$. For analysis, it is convenient to treat the data $\mathbf{X}$ as $\mathbf{X} = \mu(\mathbf{\Theta}) + \mathbf{E}$, where $\mu(\mathbf{\Theta})$ and $\mathbf{E}$ represents the underlying patterns and the noises, respectively. In particular, $\mathbf{\Theta}$ is the *canonical paramters* of the model and $\mu(\cdot)$ is a link function transforming $\mathbf{\Theta}$ to the *mean parameters* in the space of $\mathbf{X}$. In standard PCA, $\mu(\cdot)$ is the identical function and $\mathbf{\Theta}$ is a low rank matrix represented by $q < d$ factors $\mathbf{\Theta} = \mathbf{WY}$, where $\mathbf{W} \in \mathbb{R}^{d\times q}$ and $\mathbf{Y} \in \mathbb{R}^{q\times N}$. PCA finds $\mathbf{W}$ and $\mathbf{Y}$ by minimising the squared error between $\mu(\mathbf{\Theta}) = \mathbf{\Theta}$ and $\mathbf{X}$. This equals to max likelihood estimation in a model of $\mathbf{X}$, where the mean (and canonical) parameters are $\mathbf{\Theta}$.

The Gaussian distributions in probabilistic PCA can be replaced with general exponential family distributions [5]. This generalisation allows the model to deal with non-real-valued data, since general $\mu(\cdot)$ allows $\mathbf{\Theta}$ and $\mathbf{X}$ to be in different spaces. Without loss of generality, we consider an element of $\mathbf{X}$. If an exponential family likelihood function is used, the distribution takes the form of

$$p(x|\theta) = \exp\{x\theta + g(\theta) + h(x)\}, \tag{1}$$

where $g(\cdot)$ and $h(\cdot)$ characterise the distribution [11]. For example, if Bernoulli likelihood is used, (1) is realised as

$$p(x|\mu(\theta)) = \mu^x \cdot (1-\mu)^{1-x} = \exp\{x \cdot \theta + g(\theta) + h(x)\},$$

where we have $\mu(\theta) = \frac{e^\theta}{1+e^\theta}$, $g(\theta) = \log \frac{1}{1+e^\theta}$ and $h(x) = 1$.

A. Model definition

In SePCA, the prior of a latent factor $\boldsymbol{y}_n$ is a Gaussian

$$p(\boldsymbol{y}_n) = \mathcal{N}(\boldsymbol{y}_n|\mathbf{0}, \mathbf{I}), \tag{2}$$

where $\mathbf{0}$ and $\mathbf{I}$ are zero vector and identity matrix, respectively. According to ARD, the coefficient vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_q$ follow

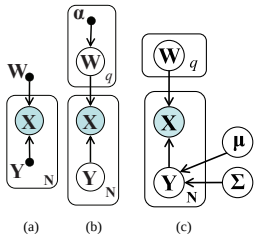


Figure 1. Structures of three classes of factor models. **(a)** models treating $\mathbf{W}$ and $\mathbf{Y}$ as fixed parameters; **(b)** empirical Bayes models; **(c)** complete Bayesian models. See text for details.

a zero-mean isotropic Gaussian prior controlled by a precision (inverse variance) hyper-parameter,

$$p(\mathbf{W}|\boldsymbol{\alpha}) = \textstyle\prod_{j=1}^{q} \mathcal{N}(\boldsymbol{w}_j|\mathbf{0}, \alpha_j^{-1}\mathbf{I}), \tag{3}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_q\}$ represents the set of precision hyper-parameters. The number of factors q can be tentatively set sufficiently large, *e.g.* $q = d - 1$. When learning the model, $\boldsymbol{\alpha}$ will be automatically tuned according to the relevance of each factor to the data. The coefficients of irrelevant factors will diminish to remove unnecessary complexity from the resultant model [12].

Given $\mathbf{W}$ and $\boldsymbol{y}_n$, an exponential family distribution can be specified by the canonical parameter vector $\boldsymbol{\theta}_n = \mathbf{W}\boldsymbol{y}_n$. Then $\boldsymbol{x}_n$ is drawn from

$$p(\boldsymbol{x}_n|\mathbf{W}, \boldsymbol{y}_n) = Expon(\boldsymbol{x}_n|\boldsymbol{\theta}_n) \tag{4}$$

where $Expon(\boldsymbol{x}_n|\boldsymbol{\theta}_n)$ represents a vector form of the exponential family distribution, which is the product of individual scalar densities defined in (1). Putting (2), (3) and (4) together, we arrive at the joint distribution

$$\begin{aligned} &\log p(\mathbf{X}, \mathbf{Y}, \mathbf{W}|\boldsymbol{\alpha}) \\ =&\textstyle\sum_n [\boldsymbol{x}_n^T \mathbf{W}\boldsymbol{y}_n + g(\mathbf{W}\boldsymbol{y}_n) + h(\boldsymbol{x}_n) - \frac{1}{2}\|\boldsymbol{y}_n\|_2^2] \\ &-\tfrac{1}{2}\textstyle\sum_{j=1}^{q} \left(\alpha_j \|\boldsymbol{w}_j\|_2^2 - d \log \alpha_j\right) + \text{Const.} \end{aligned} \tag{5}$$

Note that for a vector input, $g(\cdot)$ and $h(\cdot)$ in (5) represent the element-wise sum of functions $g(\cdot)$ and $h(\cdot)$.

For a comprehensive perspective SePCA, we graphically compare the structures of three classes of latent factor models in Fig. 1. The models are mainly categorised based on their treatment of the coefficients $\mathbf{W}$ and the factors $\mathbf{Y}$[1]. The first class treats the unknowns as fixed parameters to be determined in the training process. This umbrella model family has members of standard PCA [45, 8], EPCA [5, 46] and some sparse variants [31, Sec 3.2][40, 34, 36]. Regarding to the treatment of the coefficients, we also put the probabilistic models in [6] and [8] in this class. In [6, 8], fitting the model is to optimise the coefficients as deterministic parameters with the factors being marginalised out. Fig. 1(a) illustrates the structure of this model class, where both $\mathbf{W}$ and $\mathbf{Y}$ are shown as dots indicating fixed parameters.

[1]Due to the extensive diversity of the models, the classification is inevitably subjective and Procrustean to some extent. Each classe represents a rich family, which is impossible to exhaust here. As examples, we mainly recap works that are mentioned in Sec. II. For a detailed account of the literature with discussion in depth, [31] can be referred.

For lucidity, we consider the third model class before the second one. The factor models [41, 32, 42] of the third class employ the complete Bayesian formalism, where all interested quantities are taken as random variables. This probabilistic treatment includes not only $\mathbf{W}$ and $\mathbf{Y}$, but also any relevant parameter that is required to define the priors of $\mathbf{W}$ and $\mathbf{Y}$. This line of construction can be carried along for several steps, and generally results in a hierarchical model. Sampling based techniques are often employed for inference and prediction. Although sharing the Bayesian principle, in this class, the specific structure can vary fairly from one model to another. We take Bayesian exponential family PCA (BEPCA) [41] as an example for illustration in Fig 1(c). In BEPCA, the coefficients follow a conjugate prior w.r.t. the exponential family likelihood of the observations. The latent factors follow a diagonal Gaussian prior, whose mean and variance are also random variables.

The second model class refers to the empirical Bayes methods. Besides the proposed SePCA using ARD [23, 12], we attribute [25, 29, 30, 37] to this methodology. Like complete Bayes, empirical Bayes uses random variables to represent the coefficients and factors and is concerned about their posterior distributions. However, in empirical Bayes, prior parameters carrying interested information of $\mathbf{W}$ and/or $\mathbf{Y}$ are learned from data in a deterministic and usually iterative procedure. For example, ARD implements sparse Bayesian learning by optimising the prior variance of the coefficients or factors: if the prior variance converges to zero, the corresponding coefficients or factors are effectively eliminated from the model. Fig 1(b) displays the structure of SePCA as an example, where ARD is applied to the coefficients $\mathbf{W}$.

Both empirical and complete Bayes are probabilistic methodologies and are utilised to meet overlapped demands, such as dealing with missing values or providing declarative representation of the data population. Some empirical Bayes methods have equivalent complete Bayes formulation. For example, applying ARD on a variable corresponds to a hierarchical construction of a Student's-t prior on the variable. More generally, the Student's-t prior is a special case of a family of Gaussian scale-mixture distributions [47]. On the other hand, empirical Bayes procedures can often be cast as an optimisation problem [29] to which many fast and well-established computational routines are available, and from the alternative formulation, the problem can be generalised beyond the original Bayesian model [30]. For factor models, the empirical Bayes employed by SePCA provides us an explicit answer to the model selection problem. Besides overcoming the over-fitting problem, this is particularly helpful for applications such as dimension reduction and data visualisation. Moreover, we will show that the simple computation procedure of SePCA enables us to explain the rationale behind ARD as the intuitive principle of explaining data by using simple models.

B. Estimate of $\boldsymbol{\alpha}$

Given the data $\mathbf{X}$, $\boldsymbol{\alpha}$ is determined by maximising the marginal conditional likelihood $p(\mathbf{X}|\boldsymbol{\alpha})$, which is intractable to compute. We therefore employ a generalised EM algorithm [48] to iteratively estimate $\boldsymbol{\alpha}$ and the posterior distribution of the latent random variables $\mathbf{W}$ and $\mathbf{Y}$.

Algorithm 1: MAP Estimation of $\mathbf{W}$ 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**
  $\mathbf{Y}^{\text{MP}} \leftarrow \arg\max_{\mathbf{Y}} \big(\log p(\mathbf{X}|\mathbf{W}^{\text{MP}}, \mathbf{Y}) + \log p(\mathbf{Y})\big)$
  $\mathbf{W}^{\text{MP}} \leftarrow \arg\max_{\mathbf{W}} \big(\log p(\mathbf{W}, \mathbf{Y}^{\text{MP}}) + \log p(\mathbf{W}|\boldsymbol{\alpha})\big)$
end

In the M-step, we estimate $\boldsymbol{\alpha}$ by maximising a lower bound of the log-evidence. Given the estimation of the posterior of $\mathbf{W}$ and $\mathbf{Y}$ as $q(\cdot)$, the lower bound can be obtained by reformulating the log-evidence

$$\begin{aligned}
&\log p(\mathbf{X}|\boldsymbol{\alpha}) = \textstyle\int_{\mathbf{W},\mathbf{Y}} q(\mathbf{W},\mathbf{Y}) \log p(\mathbf{X}|\boldsymbol{\alpha})\, \mathrm{d}\mathbf{W}\, \mathrm{d}\mathbf{Y} \\
=&\textstyle\int_{\mathbf{W},\mathbf{Y}} q(\mathbf{W},\mathbf{Y}) \log \Big[\frac{q(\mathbf{W},\mathbf{Y})}{\frac{p(\mathbf{X},\mathbf{W},\mathbf{Y}|\boldsymbol{\alpha})}{p(\mathbf{X}|\boldsymbol{\alpha})}} \frac{p(\mathbf{X},\mathbf{W},\mathbf{Y}|\boldsymbol{\alpha})}{q(\mathbf{W},\mathbf{Y})} \Big]\, \mathrm{d}\mathbf{W}\, \mathrm{d}\mathbf{Y} \\
=&\mathcal{L}(\boldsymbol{\alpha}) + \mathbf{KL}(q\|p), \qquad (6)
\end{aligned}$$

where $\mathbf{KL}(q\|p) \geq 0$ is the Kullback-Leibler divergence between $q(\cdot)$ and the true posterior of $(\mathbf{W},\mathbf{Y})$ given $\mathbf{X}$ and $\boldsymbol{\alpha}$. Then the lower bound is

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\alpha}) &= \textstyle\int_{\mathbf{W},\mathbf{Y}} q(\mathbf{W},\mathbf{Y}) \log \Big[\frac{p(\mathbf{X},\mathbf{W},\mathbf{Y}|\boldsymbol{\alpha})}{q(\mathbf{W},\mathbf{Y})} \Big]\, \mathrm{d}\mathbf{W}\, \mathrm{d}\mathbf{Y} \\
&= \mathbb{E}\big[\log p(\mathbf{X},\mathbf{W},\mathbf{Y}|\boldsymbol{\alpha})\big] + \mathbf{H}[q], \qquad (7)
\end{aligned}$$

where $\mathbf{H}$ represents entropy, and the expectation $\mathbb{E}[\cdot]$ is taken over $q(\mathbf{W},\mathbf{Y})$.

At the optimum, the derivative of the lower bound $\mathcal{L}(\boldsymbol{\alpha})$ w.r.t. each α_j is zero, therefore we have

$$\begin{aligned}
&\tfrac{\partial \mathcal{L}}{\partial \alpha_j} = \mathbb{E}\Big[\tfrac{\partial \log p(\mathbf{X},\mathbf{W},\mathbf{Y}|\boldsymbol{\alpha})}{\partial \alpha_j}\Big] = -\tfrac{1}{2}\mathbb{E}[\|\boldsymbol{w}_j\|_2^2] - \tfrac{d}{\alpha_j} = 0 \\
&\text{then} \quad \alpha_j = \tfrac{d}{\mathbb{E}[\|\boldsymbol{w}_j\|_2^2]} \quad \text{for } j = 1, \ldots, q. \qquad (8)
\end{aligned}$$

In the training process, when a factor-j has coefficients of small magnitude $\|\boldsymbol{w}_j\|_2$, the variance of the coefficients $\boldsymbol{w}_j$ is estimated to be small. If this iterates progressively, the variance of $\boldsymbol{w}_j$ converges to zero. In terms of ARD, the precision of $\boldsymbol{w}_j$, α_j, goes to infinity, and the corresponding factor is effectively cut off from the resultant data representation.

C. Inference of latent variables

Updating $\boldsymbol{\alpha}$ according to (8) involves computing expectation of $\|\boldsymbol{w}_j\|_2^2$ over the posterior distribution of $\mathbf{W}$ and $\mathbf{Y}$. In theory, any estimated posterior can be employed for learning SePCA. We will develop algorithms based on MAP and the Monte Carlo method.

a) MAP: Given $\boldsymbol{\alpha}$, the MAP of $\mathbf{W}$ and $\mathbf{Y}$ can be obtained by maximising the log-joint probability (5). In particular, we consider the posterior as $L_{WY} = \log p(\mathbf{Y},\mathbf{W}|\mathbf{X},\boldsymbol{\alpha}) = \log p(\mathbf{X},\mathbf{Y},\mathbf{W}|\boldsymbol{\alpha}) - \log p(\mathbf{X}|\boldsymbol{\alpha})$, where $\log p(\mathbf{X}|\boldsymbol{\alpha})$ is constant w.r.t. $\mathbf{W}$ or $\mathbf{Y}$. In general, the global optimum is not tractable. However, fixing $\mathbf{W}$ or $\mathbf{Y}$ and optimising L_{WY} w.r.t. the other is a concave problem. Taking $\mathbf{Y}$ for example, the problem is concave because L_{WY} consists of a negative

quadratic term of $\mathbf{Y}$ and a linear term transformed by a concave function g. Note that for an exponential family distribution in the form of (1), the second derivative $g''(\theta) < 0$ for all θ [11]. Therefore, L_{WY} is concave w.r.t. $\mathbf{Y}$ given $\mathbf{W}$. The same holds for $\mathbf{W}$ given $\mathbf{Y}$ as well. The detailed derivation is given in (12) and (10) below. We adopt a similar alternating optimisation scheme as suggested in [5] [46], which is summarised in Algorithm 1. The algorithm can be implemented by optimising variables associated with individual factors, i.e. we compute the MAP estimation one column of $\mathbf{W}$ and the corresponding row of $\mathbf{Y}$, consecutively. The derivatives of L_{WY} w.r.t. $\mathbf{W}$ and $\mathbf{Y}$ required for the optimisation are as follows

$$\frac{\partial L_{WY}}{\partial w_{i,j}} = \boldsymbol{x}_{i,\cdot}\boldsymbol{y}_{j,\cdot}^T + \textstyle\sum_s g'(\theta_{i,s})Y_{j,s} - \alpha_j w_{i,j}, \quad (9)$$

$$\frac{\partial^2 L_{WY}}{(\partial w_{i,j})^2} = -\alpha_j + \textstyle\sum_s g''(\theta_{i,s})y_{j,s}^2, \quad (10)$$

$$\frac{\partial L_{WY}}{\partial y_{j,n}} = \boldsymbol{x}_n^T\boldsymbol{w}_j + \textstyle\sum_r g'(\theta_{r,n})W_{r,j} - y_{j,n}, \quad (11)$$

$$\frac{\partial^2 L_{WY}}{(\partial y_{j,n})^2} = -1 + \textstyle\sum_r g''(\theta_{r,n})w_{r,j}^2, \quad (12)$$

where $\theta_{\cdot,\cdot}$ refers to an element of the matrix $\boldsymbol{\Theta} = \mathbf{WY}$.

b) Hybrid Monte Carlo: For performing ARD, an alternative to MAP is to draw samples of $\mathbf{W}$ and $\mathbf{Y}$ from the posterior distribution and then use the samples to compute the expectation in (8)[2]. Following [41], we adopt *Hybrid Monte Carlo* (HMC) to draw samples of $\mathbf{W}$ and $\mathbf{Y}$, because both variables are continuous, and the derivatives to L_{WY} are given by (9) and (11). In contrast to general Gibbs sampling, HMC uses the gradient of the probability density. This alleviates random walking, and the Markov chain converges to the posterior faster than it does in Gibbs sampling [49, 50, 51]. The procedure of HMC is shown in Algorithm 2. We use $\mathbf{V}$ to collectively represent the latent variables $\mathbf{W}$ and $\mathbf{Y}$. The gradient in Ln. 3 is computed as that in 9 and 11. After the Hamilton simulation, a new sample is accepted in Ln. 6 at the rate $\min(1, exp(H_{\text{Old}} - H_{\text{New}}))$, where $H = \|\boldsymbol{p}\|_2^2/2 + E$ and E is the energy computed in Ln. 4 and Ln. 6, respectively. Note that the efficiency of the algorithm can be affected by the implementation of the Hamilton simulation loop in Line 5: a finer step size ϵ provides higher numerical accuracy and may lead to higher acceptance, on the other hand, it increases the number of steps of the loop. In practice, an adaptive scheme may be employed to improve efficiency, where ϵ is adjusted so that the acceptance rate is within a range, e.g. $[0.5, 0.8]$.

Generally, SePCA can be implemented by using any suitable inference method. The two approaches developed above represent the point estimate (MAP) and random simulation (HMC) of the posterior of $(\mathbf{W}, \mathbf{Y})$. Besides these two methods, for example, it is also possible to partially marginalise the posterior (EM) [6]. We choose to use MAP, which is generally faster and can be readily implemented. More importantly, MAP also helps us understand the ARD procedure, which will be discussed in the next section. However, we should be aware of the disadvantages of using MAP. MAP can be wasteful in the first few steps, because the coordinate descent in the joint space of $\mathbf{W}$ and $\mathbf{Y}$ may spend much time optimising many correlated factors only to be discarded later. MAP also lacks the theoretical guarantee that sampling methods provide and can be less accurate. We will investigate these issues by using an example later in Sec. (V-A).

[2]We should not directly estimate (8) using S posterior samples $\{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(S)}\}$, because the samples are not separately identifiable: e.g. joint permutation of the factors will not change L_{WY}. We can use $\hat{\boldsymbol{\Theta}} = \frac{1}{S}\sum_s \mathbf{W}^{(s)}\mathbf{Y}^{(s)}$ as a point estimate of $\boldsymbol{\Theta}$ and obtain $\mathbf{W}$ from $\hat{\boldsymbol{\Theta}}$, e.g. by standard PCA.

Algorithm 2: Hybrid Monte Carlo on $\mathbf{W}$ and $\mathbf{Y}$ [51]

Input: $\mathbf{X}$, $\boldsymbol{\alpha}$, $\mathbf{V}_{\text{Init}} := (\mathbf{W}_{\text{Init}}, \mathbf{Y}_{\text{Init}})$
Output: A sample $\mathbf{V} := (\mathbf{W}, \mathbf{Y})$
1 **repeat**
2 Draw sample $\boldsymbol{p}$ of size($\mathbf{V}$) from independent $\mathcal{N}(0, 1)$
3 $\boldsymbol{g} \leftarrow -\partial/\partial\mathbf{V}\big(\log p(\mathbf{X}, \mathbf{W}, \mathbf{Y}|\boldsymbol{\alpha})\big)$
4 $E \leftarrow -\log p(\mathbf{X}, \mathbf{W}, \mathbf{Y}|\boldsymbol{\alpha})$
5 **for** $\tau = 1 \ldots T$ **do** $\boldsymbol{p} \leftarrow \boldsymbol{p} - \epsilon\boldsymbol{g}$, $\mathbf{V} \leftarrow \mathbf{V} + \epsilon\boldsymbol{p}$; `/* Hamilton */`
6 Update E and determine acceptance of the sample
7 **until** *Burning-in steps finish*;

IV. ARD PRIOR IN SEPCA MODEL LEARNING

The learning of $\boldsymbol{\alpha}$ in the ARD scheme can be seen as an empirical Bayes treatment of the model, where we assume a diffuse prior for the elements in $\boldsymbol{\alpha}$. The empirical Bayes estimation of the α-parameters and the inference of the factor variables jointly define the ARD scheme, which we will discuss in this section. Based on the understanding of ARD, we will also discuss how to weigh the evidence of the samples in practical implementation.

A. Understanding ARD

For one factor, we examine the computation of the corresponding coefficients in $\mathbf{W}$ and that of the ARD parameter α. For each iteration in Algorithm 1, we obtain the local optimum of $\mathbf{W}^{\text{MP},*}$ and $\mathbf{Y}^{\text{MP},*}$, where the * stands for intermediate results. Let the j-th latent dimension be of interest. In this section, we use simplified symbols in our discussion to be less cluttered. We denote the coefficients of the j-th factor, i.e. the j-th column in $\mathbf{W}$, as $\boldsymbol{w}$. The values of the corresponding factor among the data, i.e. the j-th row of $\mathbf{Y}$, as a N-dimensional *row* vector $\boldsymbol{y} = [y_1, \ldots, y_N]$. The j-th ARD parameter is α. Formally, $\boldsymbol{w}$, $\boldsymbol{y}$ and α stand for $\boldsymbol{w}_j^{\text{MP},*}$, $\boldsymbol{y}^{j,\text{MP},*}$, and α_j^* respectively in this section.

Since it is the magnitude of $\boldsymbol{w}$ that will be used to update α as in (8), we write $\boldsymbol{w} = t\boldsymbol{w}_0$, where $\boldsymbol{w}_0$ is the optimal direction, $\|\boldsymbol{w}_0\|_2 = 1$, and t is the magnitude of $\boldsymbol{w}$. Using this representation of $\boldsymbol{w}$ and comparing Ln. 4 in Algorithm 1, we can see that at the local optimum, t satisfies

$$t = \arg\max_t \left\{ [\mathbf{X} \odot \boldsymbol{\Theta} + g(\boldsymbol{\Theta}) + h(\mathbf{X})]_{\text{es}} - \frac{\alpha}{2}t^2 \right\}, \quad (13)$$

$$\text{where } \boldsymbol{\Theta} = \mathbf{WY} = \boldsymbol{\Theta}_{\setminus j} + t\boldsymbol{w}_0\boldsymbol{y} \text{ and } \boldsymbol{\Theta}_{\setminus j} = \mathbf{W}_{\setminus j}\mathbf{Y}_{\setminus j},$$

the subscript $_{\setminus j}$ represents "all but the j-th", $[\cdot]_{\text{es}}$ is the element-wise sum and $\odot$ is the element-wise product. Dissecting the relevant terms and presenting the dependence on t explicitly, we have

$$t = \arg\max_t \Big\{ t\left[\mathbf{X} \odot (\boldsymbol{w}_0 \boldsymbol{y})\right]_{\text{es}} + \left[g(\boldsymbol{\Theta}_{\setminus j} + t\boldsymbol{w}_0\boldsymbol{y})\right]_{\text{es}} - \tfrac{\alpha}{2}t^2 \Big\}. \quad (14)$$

At the optimum, the derivative of (14) w.r.t. t is zero. With some algebra, we have

$$\langle \mathbf{X} + g'(\boldsymbol{\Theta}), \boldsymbol{w}_0\boldsymbol{y}\rangle = \alpha t, \quad (15)$$

where $\langle \boldsymbol{A}, \boldsymbol{B}\rangle$ represents the sum of element-wise product of $\boldsymbol{A}$ and $\boldsymbol{B}$. We can consider it as the inner product of two matrices.

c) Implications of eliminating a factor: Equation (15) shows that when ARD determines to discard the j-th factor, both sides of (15) are equal to 0, and $\boldsymbol{\Theta} = \boldsymbol{\Theta}_{\setminus j}$. Considering the term $g'(\boldsymbol{\Theta}_{\setminus j})$, as a property of the exponential family distributions, we have

$$g'(\boldsymbol{\Theta}_{\setminus j}) = -\mathbb{E}_{\boldsymbol{\Theta}_{\setminus j}}[\mathbf{X}], \quad (16)$$

where $\mathbb{E}_{\boldsymbol{\Theta}_{\setminus j}}[\mathbf{X}]$ is the expectation of $\mathbf{X}$ over the distribution specified by the canonical parameter $\boldsymbol{\Theta}_{\setminus j}$, and $\boldsymbol{\Theta}_{\setminus j}$ has interpretation given by (13) as the canonical parameter constructed by using all the factors except the j-th one of our interest. In the following, we denote $\mathbb{E}_{\boldsymbol{\Theta}_{\setminus j}}[\mathbf{X}]$ as $\bar{\mathbf{X}}_{\setminus j}$, and (15) becomes

$$\langle \mathbf{X}, \boldsymbol{w}_0\boldsymbol{y}\rangle = \langle \bar{\mathbf{X}}_{\setminus j}, \boldsymbol{w}_0\boldsymbol{y}\rangle. \quad (17)$$

From (1), the fitness of a set of parameters to the data are positively related to their inner product. We can roughly consider the inner product to be how much the parameters *explain* the data. Therefore the inner product equation of (17) has the following implications. We remove the j-th factor from the model and obtain an exponential family distribution that has the mean of $\bar{\mathbf{X}}_{\setminus j}$. This model prediction differs from the observation by $\mathbf{X} - \bar{\mathbf{X}}_{\setminus j}$. Then (17) states that adding the j-th factor to the model does not help explain the discrepancy between the prediction and the observation.

d) Condition of eliminating a factor: We have discussed the indication of eliminating the j-th factor by ARD. We can now consider the condition that makes the elimination take place, and analyse the corresponding indications.

Taking $\boldsymbol{y}$ into consideration, the relative terms in (14) are

$$t\left[\mathbf{X} \odot (\boldsymbol{w}_0\boldsymbol{y})\right]_{\text{es}} + \left[g(\boldsymbol{\Theta})\right]_{\text{es}} - \frac{1}{2}\|\boldsymbol{y}\|_2^2. \quad (18)$$

Setting the derivative of (18) w.r.t. $\{y_n\}_{n=1}^N$ to zero, we obtain an equation array

$$\langle \mathbf{X}_n, t\boldsymbol{w}_0\rangle + \langle g'(\boldsymbol{\Theta}(\cdot, n)), t\boldsymbol{w}_0\rangle = y_n, \quad n = 1 \ldots N. \quad (19)$$

Multiplying both sides of each equation in (19) with y_n respectively and summing up, we arrive at

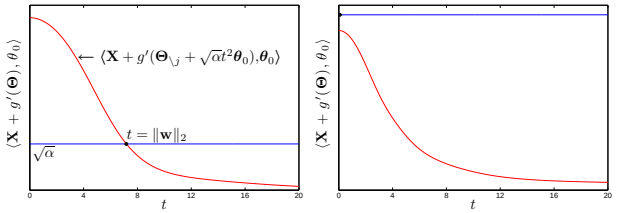
$$\langle \mathbf{X} + g'(\boldsymbol{\Theta}), t\boldsymbol{w}_0\boldsymbol{y}\rangle = \|\boldsymbol{y}\|_2^2. \quad (20)$$

Multiplying both sides of (15) with t, we have

$$\langle \mathbf{X} + g'(\boldsymbol{\Theta}), t\boldsymbol{w}_0\boldsymbol{y}\rangle = \alpha t^2. \quad (21)$$

Comparing (20) and (21) yields

$$\sqrt{\alpha}t = \|\boldsymbol{y}\|_2. \quad (22)$$



(a) $q = 2$ (b) $q = 4$

Figure 2. ARD for two candidate latent factors. X-axis represents t, the magnitude of the latent factor. The curve (red) and the horizontal lines (blue) show the l.h.s. and r.h.s. of (24), respectively. The intersection (or lack of it) indicates a positive (or zero) solution for t. **(a)**: a positive solution of t exists, thus the 2nd latent factor is preserved. **(b)**: no intersection exists, and (23) holds only for $t = 0$. The 4th factor is discarded.

Substituting (22) into (20), and letting $\boldsymbol{y} = \|\boldsymbol{y}\|_2\boldsymbol{y}_0$, $\boldsymbol{\theta}_0 = \boldsymbol{w}_0\boldsymbol{y}_0$, we have a condition of the optimal solution

$$\langle \mathbf{X} + g'(\boldsymbol{\Theta}_{\setminus j} + \sqrt{\alpha}t^2\boldsymbol{\theta}_0), t^2\boldsymbol{\theta}_0\rangle = \sqrt{\alpha}t^2. \quad (23)$$

The optimal condition (23) holds in two situations: (i) when $t = 0$, i.e. the j-th factor is discarded as we have discussed above; or (ii) for some $t > 0$

$$\langle \mathbf{X} + g'(\boldsymbol{\Theta}_{\setminus j} + \sqrt{\alpha}t^2\boldsymbol{\theta}_0), \boldsymbol{\theta}_0\rangle = \sqrt{\alpha} \quad (24)$$

The derivative to t of the l.h.s. of (24) is $2t\sqrt{\alpha}\langle g''(\boldsymbol{\Theta}), \boldsymbol{\theta}_0 \odot \boldsymbol{\theta}_0\rangle$, which is always less than zero, as $g''(\boldsymbol{\Theta}) < 0$ according to the property of exponential family distributions. Therefore (23) and (24) hold for some $t > 0$ if and only if

$$\langle \mathbf{X} + g'(\boldsymbol{\Theta}_{\setminus j} + \sqrt{\alpha}t^2\boldsymbol{\theta}_0), \boldsymbol{\theta}_0\rangle|_{t=0} = \langle \mathbf{X} - \bar{\mathbf{X}}_{\setminus j}, \boldsymbol{\theta}_0\rangle > \sqrt{\alpha}. \quad (25)$$

The ARD parameter α therefore makes a threshold for a factor to survive: the factor needs to contribute to explain the discrepancy between the model prediction without it and the observations (cf. interpretation of (17)).

Figure 2 illustrates the optimal conditions for t discussed above. The figure shows the computation of the l.h.s. of (24) and the threshold for two factors, when SePCA is fitted to a set of synthetic data (cf. Sec. V-A). Ideally, the data can be explained by using 3 factors. The figures demonstrate the computation for the 2nd and the 4th factors, where the former has a positive solution and the latter is discarded.

The discussion shows that the ARD embodies the prime principle of dimension reduction: using more factors to represent the data increases the model complexity. The complexity involved in one factor can only be justified if that factor explains sufficient information in the data.

B. Sample size and ARD

In practice, we observe that the solution of t to the optimal condition (24) is affected by the sample size, given that all the other aspects are the same. Specifically, let us assume that $t^{(1)}$ solves (24) for some $\Phi^{(1)} = \{\mathbf{X}, \boldsymbol{\Theta}_{\setminus j}, \boldsymbol{\theta}_0, \alpha\}$. We then consider duplicating the samples such that $\mathbf{X}' = [\mathbf{X}\,\mathbf{X}], \boldsymbol{\Theta}'_{\setminus j} = [\boldsymbol{\Theta}_{\setminus j}\,\boldsymbol{\Theta}_{\setminus j}], \boldsymbol{\theta}'_0 = [\boldsymbol{\theta}_0\,\boldsymbol{\theta}_0]$ and constructing a new equation by using $\Phi^{(2)} = \{\mathbf{X}', \boldsymbol{\Theta}'_{\setminus j}, \boldsymbol{\theta}'_0, \alpha\}$ in (24). In

general, the solution $t^{(2)}$ to the new equation is not the same as $t^{(1)}$. When the difference is significant, the corresponding factor may be preserved in one case and discarded in the other; therefore, whether to preserve a factor can be to some extent an arbitrary and case-by-case decision. However, in data analysis, we want the model to behave according to the underlying population, regardless of irrelevant aspects of the observation.

To deal with the sample size problem, we alter the SePCA model by using a factor ν to scale $\log p(\mathbf{X}|\mathbf{Y},\mathbf{W})+\log p(\mathbf{Y})$ (and maintaining $\log p(\mathbf{W}|\boldsymbol{\alpha})$ unchanged) in (5). This corresponds to weighting the data by a factor ν. In particular, if we let ν inversely proportional to the sample size, we will maintain the ratio between the two parts of the joint likelihood of (5): the evidence from the data and the penalty by the ARD prior. By maintaining this ratio, we expect ARD yield models of similar complexity for datasets underlain by the same population but sampled with different conditions.

It should be stressed that having ARD give consistent results does not mean that the result is optimal. Instead, if we have obtained some weight ν that makes SePCA yield a desired model on one dataset, then the consistency allows us to reuse the (adjusted) weight on other datasets of the same kind of underlying patterns but being observed differently. Another confusion to be avoided is to mistake determining ν for directly specifying the number of factors. The weight ν works at the level of model selection. Given the value of ν, one still needs to fit SePCA and let ARD determine the number of factors for a dataset. Roughly, ν describes our prior belief on the property of the data. The appropriateness of a particular ν can be assessed with the *training data per se*, while a predetermined number of factors can only be verified by using held-out data. We will illustrate the choice of ν with comprehensive examples in Sec. V-B.

V. Experiments

A. SePCA as a generalised latent factor model

We first apply SePCA a synthetic dataset of 16 binary features. The data are generated following [41] and [52]. In particular, we randomly draw 3 *prototype* binary vectors of 16 bits. Each sample is based on one prototype, where the bits are flipped with a small probability γ. The up-left panel in Fig. 3 displays an example of the prototypes and noisy bits, where the flipping rate $\gamma = 0.1$ and there are 40 samples for each prototype making a data set of 16×120 binary bits.

A series of relevant latent factor models have been applied to this synthetic dataset, including standard PCA [45], SPCA [34], EPCA [5], sparse EPCA (SpEPCA), infinite sparse factor analysis (ISFA) [42], Bayesian exponential PCA (BEPCA) [41] and SePCA. SpEPCA is a variant of EPCA, where we regularise the latent factors by using ℓ_1 penalty following [39, 40, 31]. In particular, when solving for $\mathbf{Y}$ in the alternating optimisation of EPCA, we let

$$\mathbf{Y} = \arg\min_{\mathbf{Y}} \big\{ -\log p(\mathbf{X}|\mathbf{W},\mathbf{Y}) - \log p(\mathbf{Y}) + \lambda \|\mathbf{Y}\|_1 \big\}.$$

We also adapt the model of ISFA by using exponential family likelihood, therefore the model is referred to as eISFA. According to the binary nature of the data, the Bernoulli likelihood is used for the exponential family models. It is also worth noting that compared to the Bayesian models, SpEPCA and SPCA are given an extra advantage in the test: their sparsity is adjusted according to the ideal number of factors, which is known to be three in this example. For SPCA, the penalty is chosen so that the number of non-zero coefficients is close to $16 \times 3 = 48$. For SpEPCA, the sparsity is adjusted so that a latent factor vector of a training sample contains about three non-zero elements in average. This bias does not harm our goal of demonstrating the usefulness of the Bayesian methods.

In Fig. 3, the left part visualises the results of training the models by displaying images of the learned coefficients (in hinton graphs) and those of the reconstructed parameters. In the experiment, q is set to 15 for all models except eISFA, which infers on infinite latent factors. The visualised results of eISFA and BEPCA show one sample of each model; and for eISFA, we choose a sample of three latent factors for demonstration.

Since $q = 15$ represents an over complex model for the data generated using three prototypes, standard PCA and EPCA produce obvious over-fitting. The reconstructions closely match the noisy training data and with high confidence. The sparse coefficients provide some regularisation for SPCA. However, since many factors can be used to represent one sample, the model also causes over-fitting. On the other hand, SpEPCA penalises the latent factors of each sample and alleviates over-fitting. In fact, several latent factors are not used by any sample and become invalid. All three Bayesian models (sISFA, BEPCA and SePCA) effectively handle over-fitting. The reconstructed parameters reflect the underlying prototypes rather than the noisy training data; and the models are more prudent about the confidence of the reconstruction. Furthermore, eISFA and SePCA give an explicit number of factors. Note that the 3-factor model is one of many possible outcomes of eISFA, which is chosen here for demonstration. On the other hand, SePCA consistently converges to three factors on this data.

Fig. 3(a) and (b) show the log-likelihood of the training and test data given by the models using different number of factors. The test data are generated using the same prototypes and flipping rate as the training data. The log-likelihood refers to $p(\mathbf{X}|\mathbf{W}) = \int_{\mathbf{Y}} p(\mathbf{X}|\mathbf{W},\mathbf{Y})p(\mathbf{Y})\,\mathrm{d}\mathbf{Y}$, where $\mathbf{W}$ represents the learned coefficients and $p(\mathbf{Y})$ is unit Gaussian[3]. The log-likelihood is normalised to $[0,1]$, because our main concern is how consistent a model's behaviour is on the training and test data for various q, and because Bernoulli and Gaussian noise give probability mass and density, respectively, which are not directly comparable. Note that scores are obtained from 10 tests on random datasets, and the average scores are shown in the figure. From the scores, we can make the following observations: (i) Bernoulli observation model is more suitable than a Gaussian one, (ii) over-fitting is most obvious in PCA, SPCA and EPCA; it is alleviated in SpEPCA

[3]BEPCA infers $p(\mathbf{Y})$ from data. This difference becomes irrelevant because we normalise the scores and compare only the trend of change w.r.t. q.

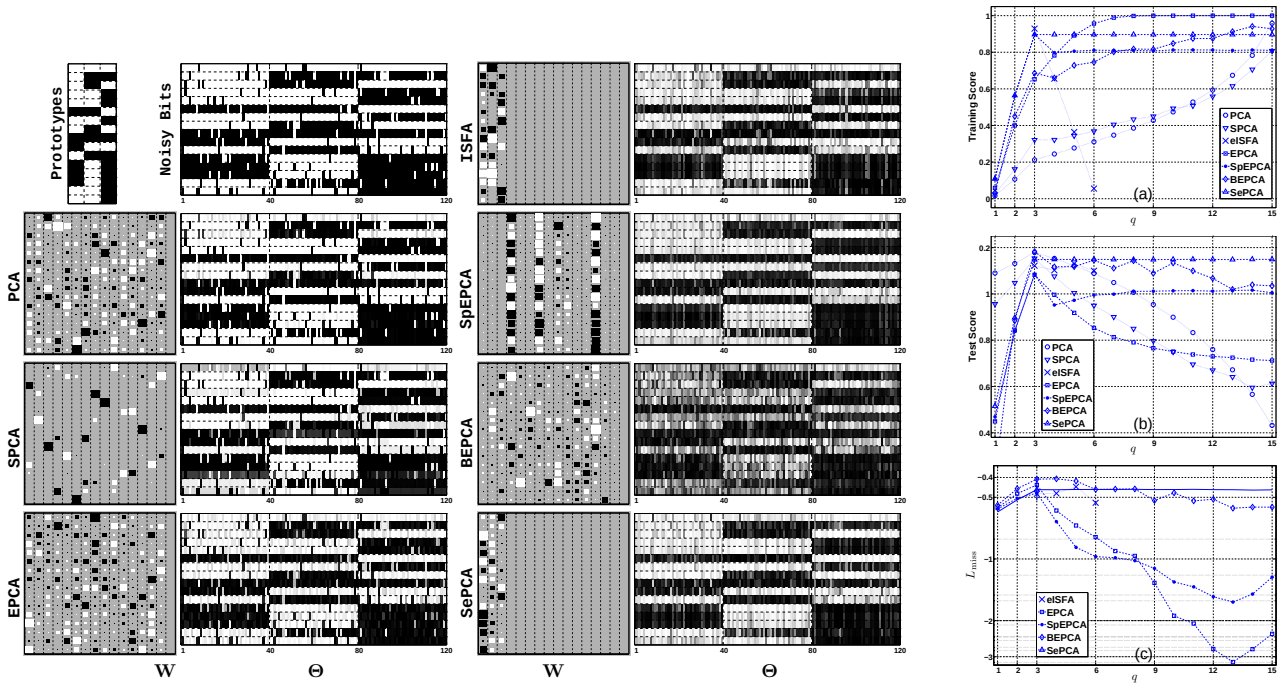


Figure 3. Experiment on synthetic noisy binary data. **(Upper-left)**: 3 prototypes and the training data. **(Hinton graphs)**: learned coefficients. **(Gray-scale images):** reconstructed parameters, 16×120. **(a–c)**: quantitative evaluations on training, test and missing data, respectively.

and Bayesian models and (iii) SePCA produces the most consistent results. When the number of factors is initially set to $q > 3$, ARD eliminates the surplus latent factors and the model automatically persists to using at most three latent factors.

To directly assess how the learned $\mathbf{W}$ and $\mathbf{Y}$ reconstruct the data, we mark 10% of the data as missing in training and consider the likelihood of these missing data

$$L_{\text{miss}} = \log p(\mathbf{X}_{\text{miss}}|\mathbf{W}, \mathbf{Y}). \tag{26}$$

In this test, over-fitting to the observations is manifested by poor prediction about the missing values. Fig. 3(c) shows that the average likelihood of a missing bit given by the exponential family models. The results are consistent with those discussed above, where regularised and Bayesian models achieve superior performance. Note that the unnormalised likelihood is shown in logarithm scale for clarity.

We study how the inference method for $\mathbf{W}$ and $\mathbf{Y}$ affects the performance of ARD. In this test, we follow similar steps as described above. We set the number of prototypes, i.e. the ideal number of factors, to 3, 4 and 5. For each setting, 10 datasets are generate with 10% missing values. SePCA models are learned by MAP and hybrid Monte Carlo (HMC), respectively. Fig. 4 compares the likelihood of the missing values and the estimated number of factors given by the models. The models computed by HMC have slightly better predicative performance than the models by MAP have, but they are comparable (Fig. 4(a)). Both methods allow SePCA to reveal the optimal number of factors in most tests (Fig. 4(b)). However, SePCA arrives the optimal number of factors in fewer iterations when using MAP as shown in Fig. 4(c). For efficiency, we employ MAP for the rest of the experiments.

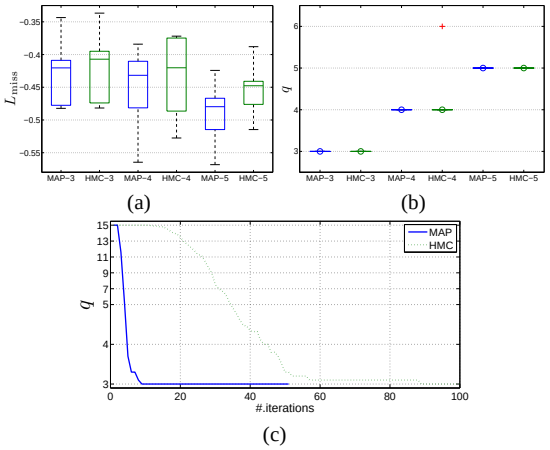


Figure 4. Inference of SePCA by MAP and HMC. X-axes of (a) and (b) correspond to the inference methods and the number of prototypes. **(a)**: missing data log-likelihood; **(b)** estimated q; **(c)** the number of iterations needed by HMC and MAP in a test where $q = 3$.

B. Effect of ν on learning

As we have shown in Sec. IV, in practice, we can affect ARD in SePCA by weighting the evidence of the observations. The weight is realised as a parameter ν and reflects our prior knowledge about the attributes of the data. In this experiment, we study how ν affects the learning of SePCA and the analysis of data.

We follow the similar procedures of the preceding experiment, but vary specific configurations to generate multiple binary datasets of different attributes. In particular, we generate the binary datasets of different sample sizes and by using different noise levels, i.e. flipping the bits at different rates. On each

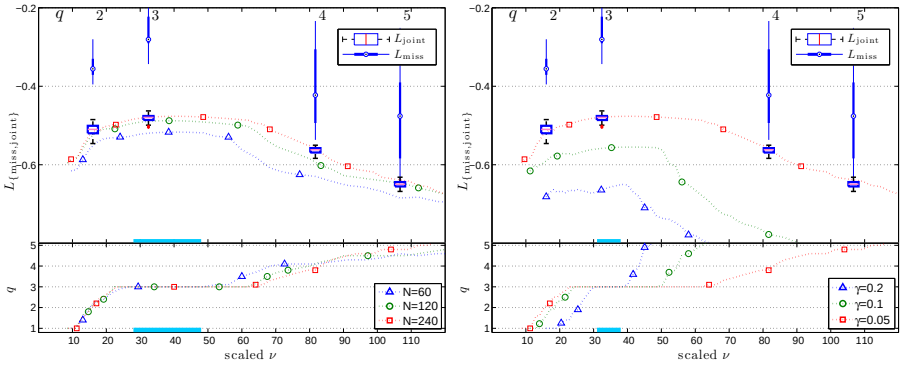


Figure 5. Choice of ν for different datasets. **(a):** testing SePCA for sample sizes $\{60, 120, 240\}$; **(b):** testing SePCA for flipping rates $\gamma \in \{0.05, 0.1, 0.20\}$. The top part shows log-likelihoods. The curves stand for L_{joint} of different ν. Three curves correspond to experiments on three datasets. The box-plots shows the statistics of L_{joint} and L_{miss} w.r.t. $q \in \{2, 3, 4, 5\}$. The bottom part shows the resultant q. We show ν-values scaled by the sample size. The bold lines on the ν-axis indicate that the ideal $q = 3$ is resulted.

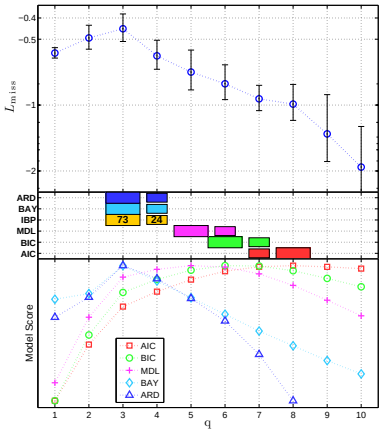


Figure 6. Model selection on synthetic data. (**top**) L_{miss} given by EPCA models with different q; (**middle**) two highest scored models by different criteria and (**bottom**) normalised model scores. ARD, BAY (Bayesian) and IBP (Indian buffet process) correspond to priors of SePCA, BEPCA and eISFA respectively.

of the dataset, we fit SePCA using a variety of ν-values. In all tests, we mark 10% of the data as missing when fitting the model, and compute the likelihood of the missing data given by the resultant models. The results of experimenting with varying sample sizes and noise levels are displayed in Fig. 5 (a) and (b), respectively. Each reported numerical result is averaged over those obtained by running 10 experiments of identical configurations on randomly generated data sets.

The top part of the graphs shows two types of log-likelihoods. First, the curve shows the joint likelihood of the fitted SePCA models, $L_{\text{joint}} = \log P(\mathbf{X}_{\text{observe}}, \mathbf{W}, \mathbf{Y}|\boldsymbol{\alpha})$ (cf. eq. 5). Second, the box-plots shows the statistics of L_{joint} and L_{miss}, where L_{miss} is defined in (26). The statistics are organised w.r.t. the resultant factor number for $q \in \{2, 3, 4, 5\}$. The bottom part of the graphs shows the resultant factor numbers, q, for different values of ν. The fractional numbers are results of averaging 10 tests. We indicate the tests where the ideal $q = 3$ is discovered by bold lines on the ν-axis in the graphs. Subfigure (a) corresponds to tests using sample sizes of 60, 120 and 240, with flipping rate $\gamma = 0.05$. Subfigure (b) corresponds to tests using flipping rates $\gamma \in \{0.05, 0.10, 0.20\}$, with 240 samples.

It should be noted that L_{joint} is evaluated using only observed data. If we consider L_{joint} as the MAP approximation of $P(\mathbf{X}_{\text{observe}}|\boldsymbol{\alpha})$, the likelihood reflects the evidence of a model obtained by ARD *without* using held-out data. The evidence then serves as a criterion for selecting a proper number of factors. The experiment outcome shows that L_{joint} is an effective criterion as follows. As expected, the highest L_{miss}'s are from the tests where $q = 3$ result. On the other hand, the tests where L_{joint} achieves its high values overlap with those yielding $q = 3$ and high L_{miss}. This agreement indicates that we can determine an appropriate ν for SePCA by using observed data *per se*, and the resultant model can be expected to have an appropriate number of factors and generalise well to unseen data.

The discussion in Sec. (IV-B) suggests that if we keep ν inversely proportional to the sample size, the model selection of ARD should behave consistently, in despite of the changes of how the data are sampled. This speculation is supported by the results of the current experiment. The results in Fig. 5 are plotted against ν scaled by the respective sizes of the training samples. The graphs clearly show that using the same scaled ν, SePCA selects similar models for all tested datasets. In particular, when the scaled ν is between 30 and 40, SePCA yields the desired $q = 3$ and high L_{joint} in all tests.

The results also quantitatively explain the intuition that the quality of the data affects the difficulty of choosing a proper model. When the samples are plenty and the noise is low, SePCA can discover the proper number of factors over a wide range of ν. On the other hand, when the sample size becomes smaller and flipping rate increases, the tolerated range of ν shrinks accordingly. This characteristic is consistent with natural expectation, and can also be predicted from (25): if the noise is significant, then a factor can be preserved to represent the noise instead of the main pattern, and a more complex model can result.

C. Model selection

In the following experiments, we use SePCA to analyse data of various types, with special attention paid to how ARD helps select appropriate models. The first experiment is on the synthetic data as introduced in V-A, for which we have the knowledge of ideal number of factors being 3. Besides using ARD of SePCA, we also test three standard model selection criteria: AIC [16], BIC [17] and MDL [18], as well as two Bayesian methods eISFA and BEPCA. For AIC, BIC and MDL, we evaluate the criteria for a range of candidate factor numbers. The evaluation is based on EPCA models with particular numbers of factors. Because SePCA does not accept

a preset number of factors, we vary ν to produce models of different numbers of factors. The score given by SePCA is the log-joint probability w.r.t. the *training* data, L_{joint} (see above in Sec. V-B). BEPCA employs Bayesian method to avoid over-fit, thus we fit one BEPCA model using the greatest number of factors in the tested range and take the parameter space produced by BEPCA, which is a set of real vectors. Then BIC scores are computed for standard PCA models with different number of factors fitted to the parameter vectors. For eISFA, we directly record the numbers of factors of the last 100 MCMC samples (10 from each running of the experiment).

In the tests, we assess $q = 1 \dots 10$. Figure 6 demonstrates the results of model selection by different methods. As an independent benchmark, we train an EPCA model using each $q = 1 \dots 10$. The top part of the graph shows log-likelihood assigned to the missing values by those EPCA models. As expected, the most suitable q is three for this dataset. We show the factor numbers selected by different criteria by drawing boxes in the graph. Each row corresponds to a model selection method, where the bigger box represents the q receiving the highest score from the method, and the smaller box indicates the second most preferred q. Note that in this figure, we name the Bayesian models by how they control complexity, ARD for SePCA, BAY (Bayesian) for BEPCA and IBP (Indian buffet process) for eISFA, which also improves the lucidity of the illustration. The occurrences of the two most frequently sampled factor numbers are displayed for eISFA. The bottom of the graph shows the normalised scores of different q given by the criteria. For SePCA, this is L_{joint}.

In this experiment, AIC, BIC and MDL choose suboptimal models, possibly because their assumptions of the marginal likelihood match poorly for exponential family latent factor models. On the other hand, three probabilistic methods agree consistently on the desired number of factors. It worth noting that BIC fails when being directly applied to the data, but succeeds when being used on the parameter space recovered by BEPCA, which suggests that new model selection criteria for general exponential family models could be further developed, e.g. based on lower bounds on the marginal likelihood.

The second model selection experiment is on the 20 newsgroup dataset[4]. We compile three corpora containing documents from 2, 3 and 4 newsgroups (cf. Fig. 7), respectively, taking 50 documents from each newsgroup and 200 words from each document. Each document is represented by a 200-D binary vector according to the presence and absence of the words. As above, we let 10% of the data be missing values for evaluating models, and repeat the experiment 10 times using randomly taken documents.

Fig. 7 shows the results of applying different model selection methods on the three corpora. The figure is organised similarly as Fig. 6, where we omit the specific scores for clarity. From the evaluation by missing value likelihood, it is clear that the appropriate factor number increases with the heterogeneity of the data, which is expected. Of the tested methods, BIC and MDL fail to reflect the requirement of using more factors for more complex data. One possible reason is that the noise in real data is higher than that in the synthetic data. Therefore when increasing number of factors, the gain of likelihood cannot compensate the penalty inflicted by BIC or MDL. AIC and the probabilistic methods produce assessments that are more consistent with the result obtained from testing on missing values. When the data contain 4 newsgroups, eISFA yields about 20 samples of 4 factors (*not* displayed in Fig. 7(c)). Thus for this setting, the estimate given by SePCA and eISFA is more accurate. In addition, according to our discussion in Sec. V-B, since we can assume that the datasets of three corpora have similar attributes, the parameter ν can be determined on one corpora and used for the other two. We record the optimal $\hat{\nu}$ on corpus-1. For corpus-2 and 3, we use $\hat{\nu}$ and report the two factor numbers given by SePCA in most of the 10 repeated tests (Fig. 7(b) and (c)).

		M_0		M_1		M_A		M_B	
		q	L_{miss}	q	L_{miss}	q	L_{miss}	q	L_{miss}
wine	e	12	-1.931	3	-0.950	2	-0.955		
iris	e	3	-2.664	2	-0.854	2	-0.854		
b.can	b	7	-46.40	1	-9.302	2	-9.577	1	-9.302
USPS-1	b	50	-8.007	10	-6.124	8	-6.127	2	-6.209
USPS-2	b	50	-17.737	20	-8.488	15	-8.585	1	-9.628
USPS-012	b	1	-9.442	25	-7.865	19	-7.909	2	-8.978

Table I
MODEL SELECTION ON UCI DATASETS

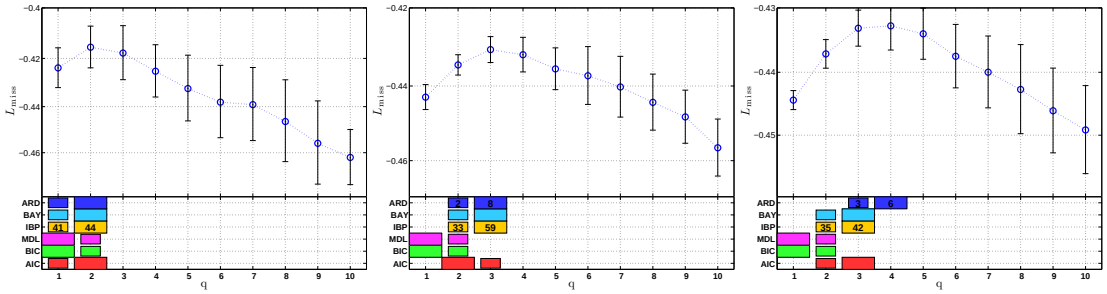
Col-1: datasets, *b.can* for "breast cancer (original)"; **Col-2**: likelihood functions, "e" for exponential and "b" for binomial. **Col**-M_0 and M_1: the model of poorest and best prediction of test data; **Col**-M_A: model selected by ARD in SePCA; **Col**-M_B: model selected by BEPCA followed by BIC.

The binary document data can be used as an example to verify the usefulness of SePCA for our task of dimension reduction. Fig. 8 compares the latent factors resulted by applying PCA and SePCA on corpus-1 for one training and one test dataset, respectively. The colours in the graphs correspond to the two newsgroups in the corpus. A classification boundary given by Fisher discriminant analysis is also delineated in the graphs. This visualised result shows that the Bernoulli likelihood is more suitable for the data, and the choice of two latent factors is appropriate.

We have also applied SePCA for model selection on several datasets from the UCI repository [5]. In particular, we assessed models on the "wine" and "iris" datasets, where the exponential distribution is employed as the likelihood function for a non-negative analysis. We also assess models on the "original breast cancer" (*b.can*) dataset and three subsets of the USPS hand-written digits, using binomial likelihood. In *b.can*, the observed values vary in $1 \dots 10$, thus we model them using $\text{binom}(11, \cdot)$. For USPS images, the images are resized to 10×10 and the grayscale is quantised to $1 \dots 16$. The likelihood function is $\text{binom}(17, \cdot)$. We take three subsets of USPS, each consisting of totally 100 images from digit "1", "2", and "012", respectively. By constructing subset "1", "2" and "012", we intend to introduce progressive levels of heterogeneity as we have done in the last experiment on

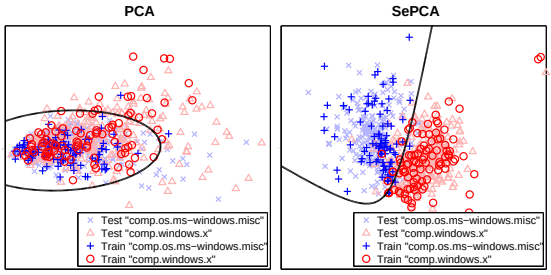
[4]http://www.ai.mit.edu/people/jrennie/20Newsgroups/

[5]http://archive.ics.uci.edu/ml/



(a) (b) (c)

Figure 7. Model selection on newsgroup data. The result is visualised in the similar way as in Fig. 6. **(a)**: corpus-1: documents from "windows.misc" and "windows.x"; **(b)** corpus-2: corpus-1 ∪ documents from "alt.atheism"; and **(c)**: corpus-3: corpus-2 ∪ documents from "sci.med". In (b) and (c), the numbers in the ARD boxes indicate how many times the corresponding q has been resulted in 10 repeated tests.



(a) (b)

Figure 8. 2D Representation of newsgroups data. **(a)** by PCA; **(b)** by SePCA. Markers of solid and faded colours illustrate the training and test data, respectively. Bold curves represent Fisher's classification boundaries.

newsgroup data (Digit "2" is written with more varieties than digit "1" is). As above, a series of EPCA models are fitted to each dataset using q in a reasonable range[6], and benchmark evaluations are obtained by using the fitted models to compute the log-likelihood of 10% missing data.

Table I lists the results of the experiment. Since there can be more than one suitable factor numbers for a practical dataset, we should not expect SePCA to select exactly the same q with which EPCA model performs best. Instead, we give a reference range of the missing data log-likelihood, against which we can compare how the chosen factor number is suited for the data. In the table, M_0 stands for the model gives the lowest missing data likelihood. We list the corresponding factor number and the log-likelihood. On the other side, M_1 stands for the highest missing data likelihood. Thus M_0 means the most *unsuitable* model and M_1 means the most suitable. M_A represents the model with q selected by SePCA. To reduce irrelevant influences on the comparison, the reported log-likelihood of M_A is computed by an EPCA model with the chosen q. SePCA selects appropriated factor numbers in all the tests, which can be verified by comparing the log-likelihood of M_A with those of M_0 and M_1. In addition, for the three USPS subsets, SePCA suggests three numbers that vary consistently with the complexity of the data. For the binomial likelihood models, we also test the BEPCA-followed-by-BIC scheme as we have described above. The results are listed in the table as M_B. In this experiment, the BIC scores do not agree with the proper q for the parameter space recovered by BEPCA. Note that the under-fitting of M_B is *not* caused by BEPCA. In fact, in all tests, BEPCA gives a log-likelihood for the missing data that is comparable to the optimal model of M_1. The under-fitting is the result of an EPCA model using a mis-chosen number of factors by BIC.

[6]For wine, iris, and *b.can*, we use $q \in \{1, \ldots, d-1\}$, where d is the number of observed features of the data; and for USPS sets, we test $q \in \{1, \ldots, 50\}$.

VI. Conclusion and Discussions

In this paper, we propose SePCA, a family of generative latent factor models. The proposed model handles data of general types using exponential family distributions, which are parameterised by the latent factors and coefficients. By applying automatic relevance determination (ARD) to the coefficients, SePCA automatically determines the appropriate number of latent factors for representing the data. Exponential family distributions play the essential role of linking real-valued factors and coefficients to data of general types. This enables ARD to operate on general type data population.

We provide a discussion on how the computation of ARD quantitatively fulfils the intuition that a factor should be useful for representing the data. The discussion leads to a sample weight parameter ν. SePCA determines a ν in the *training* stage. It would be enlightening if we can formalise the procedure into the Bayesian framework in future study. In particular, the model is affected by the sample size and noise, which is indicated by the discussion in Sec. V on results shown in Fig. 5. A systematic exploration in this aspect will be a suitable future topic.

Another related problem for future study is the choice of the observation model. To clarify, e.g. when we represent the documents from the newsgroups as integer word counts and adopt Poisson distributions as the observation model, SePCA returns more complex model than it does when the observation model is Bernoulli. A possible explanation is as follows. Given a set of predicted parameters, if we measure their fitness to the observations by using a Poisson likelihood function, the fitness will weights larger than one that is measured by a Bernoulli likelihood function. If the penalty from the prior distributions remains the same, then a Poisson likelihood function leads to a more complex model. More importantly, the current work assumes the likelihood manifests itself given the type of the

observations. However, it is often that the observations are real-valued and apparently suggest a standard Gaussian model, but the Gaussian likelihood actually fails accounting for some important prior knowledge about the physical process yielding the observations and thus is suboptimal. To this end, a model that integrates general infinitely divisible distributions would be a suitable choice. Further study will focus on the design and computation in such general models.

Another possible extension is to study model selection in supervised and semi-supervised settings. The supervised model selection problem for real-valued observation has been discussed in [21], to which dealing with general observations can be a helpful complementation. Semi-supervised learning problems are often discussed within the framework of locally linear subspace models [53], where extension of SePCA may be studied to determine the model complexity.

REFERENCES

[1] I. T. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[2] M. Bartlett, J. Movellan, and T. Sejnowski, "Face recognition by independent component analysis," *IEEE Trans. Neural Networks*, 2002.

[3] W. Bian and D. Tao, "Max-min distance analysis by using sequential SDP relaxation for dimension reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 1037–1050, 2011.

[4] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 260–274, 2009.

[5] M. Collins, S. Dasgupta, and R. E. Schapire, "A generalization of PCA to the exponential family," in *NIPS*, 2002.

[6] I. Moustaki and M. Knott, "Generalized latent trait models," *Psychometrika*, 2000.

[7] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for nonnegative matrix factorization," *IEEE Trans. Signal Proc.*, vol. 60, no. 6, pp. 2882–2898, 2012.

[8] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic PCA," *Neural Comput.*, 1999.

[9] S. Roweis, "EM algorithms for PCA and SPCA," in *NIPS*, 1997.

[10] T. Minka, "Expectation propagation for approximate Bayesian inference," in *Uncertain. Artif. Intell.*, 2001.

[11] P. McCullagh and J. A. Nelder, *Generalized Linear Models*. CRC Press, 1989, vol. 37.

[12] D. MacKay, "Probable networks and plausible predictions," *Network: Comput. in Neural Syst.*, 1995.

[13] Z. Ghahramani and G. E. Hinton, "The EM algorithm for mixtures of factor analysis," Univ. of Toronto, Tech. Rep., 1996.

[14] J. Zhao, P. Yu, and J. Kwok, "Bilinear probabilistic principal component analysis," *IEEE Trans. Neural Networks & Learn. Syst.*, vol. 23, no. 3, pp. 492–503, 2012.

[15] J. Li and D. Tao, "On preserving original variables in bayesian PCA with application to image analysis," *IEEE Trans. Image Proc.*, vol. 21, no. 12, pp. 4830–4843, 2012.

[16] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, 1974.

[17] G. E. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.

[18] P. D. Grunwald, *The Minimum Description Length Principle*. MIT Press, 2007.

[19] S. Nowlan and G. Hinton, "Simplifying neural networks by soft weight-sharing," *Neural Comput.*, 1992.

[20] J. Lv, Z. Yi, and K. Tan, "Determination of the number of principal directions in a biologically plausible PCA model," *IEEE Trans. Neural Networks*, 2007.

[21] S. Ji and J. Ye, "Generalized linear discriminant analysis: A unified framework and efficient model selection," *IEEE Trans. Neural Networks*, 2008.

[22] S. Moon and H. Qi, "Hybrid dimensionality reduction method based on support vector machine and independent component analysis," *IEEE Trans. Neural Networks & Learn. Syst.*, vol. 23, no. 5, pp. 749–761, 2012.

[23] R. Neal, *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996.

[24] A. Stuhlsatz, J. Lippel, and T. Zielke, "Feature extraction with deep neural networks by a generalized discriminant analysis," *IEEE Trans. Neural Networks & Learn. Syst.*, vol. 23, no. 4, pp. 596–608, 2012.

[25] C. M. Bishop, "Bayesian PCA," in *NIPS*, 1999.

[26] ——, "Variational principal components," in *Proc. Int. Conf. Artif. Neural Networks*, 1999.

[27] J. Berger, *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 1985.

[28] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, 2001.

[29] D. Wipf and S. Nagarajan, "A new view of automatic relevance determination," in *NIPS*, 2007.

[30] D. Wipf, B. Rao, and S. Nagarajan, "Latent variable Bayesian models for promoting sparsity," *IEEE Trans. Inform. Theory*, 2010.

[31] S. Mohamed, "Generalised Bayesian matrix factorisation models," Ph.D. dissertation, Univ. of Cambridge, 2011.

[32] S. Mohamed, K. Heller, and Z. Ghahramani, "Bayesian and L1 approaches to sparse unsupervised learning," Tech. Rep., 2011.

[33] M. Seeger, F. Steinke, and K. Tsuda, "Bayesian inference and optimal design in the sparse linear model," in *AISTATS*, 2007.

[34] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, 2006.

[35] B. Efron, T. Hastie, and I. Johnstone, "Least angle regression," *Ann. Statist.*, 2004.

[36] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," in *NIPS*, 2004.

[37] C. Archambeau and F. R. Bach, "Sparse probabilistic projections," in *NIPS*, 2009.

[38] M. Rattray, O. Stegle, K. Sharp, and J. Winn, "Inference algorithms and learning theory for Bayesian sparse factor analysis," in *Statistical-Mechanical Informatics*, 2009.

[39] S. Lee, H. Lee, P. Abbeel, and A. Y. Ng, "Efficient L1 regularized logistic regression," in *AAAI*, 2006.

[40] H. Lee, R. Raina, A. Teichman, and A. Y. Ng, "Exponential family sparse coding with applications to self-taught learning," in *IJCAI*, 2009, pp. 1113–1119.

[41] S. Mohamed, K. Heller, and Z. Ghahramani, "Bayesian exponential family PCA," in *NIPS*, 2009.

[42] D. A. Knowles and Z. Ghahramani, "Nonparametric Bayesian sparse factor models with application to gene expression modelling," *Ann. Appl. Statist.*, 2010.

[43] T. L. Griffiths and Z. Ghahramani, "Infinite latent feature models and the Indian buffet process," in *NIPS*, 2006.

[44] J. Li and D. Tao, "Simple exponential family PCA," in *AISTATS*, 2010.

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

[46] A. I. Schein, L. K. Saul, and L. H. Ungar, "A generalized linear model for principal component analysis of binary data," in *AISTATS*, 2003.

[47] N. G. Polson and J. G. Scott, "Shrink globally, act locally: Sparse Bayesian regularization and prediction," *Bayesian Analysis*, 2010.

[48] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variant," in *Learning in Graphical Models*, 1999.

[49] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid

Monte Carlo," *Phys. Lett. B*, 1987.
[50] R. M. Neal, "Probabilistic inference using Markov chain Monte Carlo methods," Tech. Rep., 1993.
[51] D. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.
[52] M. E. Tipping, "Probabilistic visualisation of high-dimensional binary data," in *NIPS*, 1999.
[53] F. Wang and C. Zhang, "Semisupervised learning based on generalized point charge model," *IEEE Trans. Neural Networks*, vol. 19, no. 7, pp. 1307–1311, 2008.

JunLi Jun Li received his BS degree in computer science and technology from Shandong University, Jinan, China, in 2003, the MSc degree in information and signal processing from Peking University, Beijing, China, in 2006, and the PhD degree in computer science from Queen Mary, University of London, London, UK in 2009. He is currently a research fellow with with the Centre for Quantum Computation and Information Systems and the Faculty of Engineering and Information Technology in the University of Technology, Sydney.

Dacheng Tao Dacheng Tao (M'07-SM'12) is Professor of Computer Science with the Centre for Quantum Computation and Intelligent Systems and the Faculty of Engineering and Information Technology in the University of Technology, Sydney. He mainly applies statistics and mathematics for data analysis problems in computer vision, data mining, machine learning, multimedia, and video surveillance. He has authored more than 100 scientific articles at top venues including IEEE T-PAMI, T-IP, ICDM, and CVPR. He received the best theory/algorithm paper runner up award in IEEE ICDM'07, K. C. WONG Education Foundation Award and honorable mention for the "Outstanding Young Researcher in Image and Vision Computing".