Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Supervised Probabilistic Robust Embedding with Sparse Noise

Yu Zhang[†], Dit-Yan Yeung[†], Eric P. Xing[‡]
[†]Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong
[‡]Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
zhangyu@cse.ust.hk; dyyeung@cse.ust.hk; epxing@cs.cmu.edu

Abstract

Many noise models do not faithfully reflect the noise processes introduced during data collection in many real-world applications. In particular, we argue that a type of noise referred to as sparse noise is quite commonly found in many applications and many existing works have been proposed to model such sparse noise. However, all the existing works only focus on unsupervised learning without considering the supervised information, i.e., label information. In this paper, we consider how to model and handle sparse noise in the context of embedding high-dimensional data under a probabilistic formulation for supervised learning. We propose a supervised probabilistic robust embedding (SPRE) model in which data are corrupted either by sparse noise or by a combination of Gaussian and sparse noises. By using the Laplace distribution as a prior to model sparse noise, we devise a two-fold variational EM learning algorithm in which the update of model parameters has analytical solution. We report some classification experiments to compare SPRE with several related models.

Introduction

Consider the three images for the same person from the AR face database (Martínez and Benavente 1998) as shown in Figure 1. While most parts of the three images look similar, some parts are occluded (e.g., by sunglasses and scarf) in at least one of the images. If we regard occlusion as the result of contaminating the data with noise, it is clear that commonly used noise models such as the additive Gaussian noise model are not suitable for such applications. This is not only true for many computer vision applications such as the one shown in Figure 1, but is also quite commonly encountered in a large number of real-world applications in which data are collected in uncontrolled environments. One characteristic of this type of noise is that it is zero or close to zero in most parts of the data (e.g., image) but may be quite significant in some other parts. Due to its sparsity, this type of noise is referred to as sparse noise, as in (Wright et al. 2009).

There are several reasons and benefits for modeling sparse noise. If we can model the noise well, we may be able to apply a denoising process to recover the original data (image).

Figure 1: Three images of a person in the AR database.

Even if the goal is not to recover the original data, taking sparse noise into consideration may help to improve the performance of the subsequent learning task, such as classification, since the existence of sparse noise violates the assumption (such as Gaussian noise assumption) made by the underlying model about the noise in the data. Some methods (Gao 2008; Wright et al. 2009; Xu, Caramanis, and Sanghavi 2010; Candes et al. 2011; Agarwal, Negahban, and Wainwright 2011) have recently been proposed to address this problem. Moreover, we note that the data in many of these applications are high-dimensional and hence incur high storage and processing costs. Fortunately, many studies have shown that the intrinsic dimensionality of such high-dimensional data is often much lower. As a result, embedding techniques play an important role in these applications.

Even though there are some works that model sparse noise, they only handle it under unsupervised learning without considering supervised information such as label information. So in this paper, we consider how to model and handle sparse noise in probabilistic embedding under the supervised learning setting. Due to its sparsity, we use the Laplace distribution as a prior to model the noise. Note that the Laplace prior corresponds to an l_1 regularizer in the regularization framework. We propose a supervised probabilistic robust embedding (SPRE) model for the supervised learning setting. In our SPRE model, data are corrupted by a combination of Gaussian and sparse noises which are modeled by both Gaussian and Laplace priors. We also consider a variant of SPRE in which data is only corrupted by sparse noise. For inference, we devise a two-fold variational expectation-maximization (EM) (Dempster, Laird, and Rubin 1977) algorithm in which the update of model parameters has analyt-

ical solution instead of directly maximizing the variational lower bound of the log-likelihood of the whole data set via gradient techniques. In experiments, we first test SPRE on some image classification problems known to have sparse noise to demonstrate the effectiveness of its sparse noise modeling ability. Moreover, we also conduct experiments on some UCI benchmark datasets where the existence of sparse noise is not known. Experimental results show that the performance of SPRE is comparable to or even better than that of state-of-the-art probabilistic embedding techniques.

SPRE Model

Without loss of generality, we consider a C-class classification problem with a training set of n examples, $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \{1, \ldots, C\}$ for $i = 1, \ldots, n$. We define a class indicator vector $\mathbf{y}_i \in \mathbb{R}^C$ for $\mathbf{x}_i$ where the kth element of $\mathbf{y}_i$ is 1 if $y_i = k$ and 0 otherwise.

The Model

We propose here a latent variable model with sparse noise in the form of an additive noise term. To model the sparse noise, distributions such as the Laplace prior or Jeffreys prior may be used. We use the Laplace prior here for simplicity. The model formulation is detailed as follows:

$$\begin{aligned}
&\mathbf{x}_i = \boldsymbol{\mu}_1 + \mathbf{W}_1\mathbf{z}_i + \boldsymbol{\varepsilon}_i + \boldsymbol{\epsilon}_i \\
&\mathbf{y}_i = \boldsymbol{\mu}_2 + \mathbf{W}_2\mathbf{z}_i + \boldsymbol{\tau}_i \\
&\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&\varepsilon_{ij} \sim \mathcal{L}(0, b) \\
&\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_1) \\
&\boldsymbol{\tau}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_2),
\end{aligned} \tag{1}$$

where $\mathbf{0}$ denotes a zero vector (or matrix) of appropriate size, $\mathbf{I}$ denotes an identity matrix of appropriate size, $\mathcal{N}(\mathbf{m}, \boldsymbol{\Sigma})$ denotes a (multivariate) normal distribution with mean $\mathbf{m}$ and covariance matrix $\boldsymbol{\Sigma}$, and $\mathcal{L}(a, b)$ denotes the Laplace distribution with location parameter a and scale parameter $b > 0$ and its probability density function is

$$p(x) = \frac{1}{2b} \exp\Big\{ - \frac{|x - a|}{b} \Big\},$$

where $|\cdot|$ denotes the absolute value of a real scalar. In our SPRE model, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ capture the mass of the whole data set and labels respectively, $\mathbf{W}_1 \in \mathbb{R}^{D\times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{C\times d}$ denote the factor loading matrices for the data and also the label, $\mathbf{z}_i$ denotes a d-dimensional latent variable shared by both $\mathbf{x}_i$ and $\mathbf{y}_i$, $\boldsymbol{\varepsilon}_i$ denotes the sparse noise for $\mathbf{x}_i$ with ε_{ij} as its jth element, and $\boldsymbol{\epsilon}_i$ and $\boldsymbol{\tau}_i$ capture the Gaussian noise contained in $\mathbf{x}_i$ and $\mathbf{y}_i$. Here our model in Eq. (1) assumes that the data points possess sparse noise but their labels do not. Moreover, as $b \to 0$ which means the effect of sparse noise becomes very small, model (1) will become the probabilistic linear discriminant analysis (PLDA) model (Bach and Jordan 2005) which is a well-known probabilistic model capable of recovering the solution of linear discriminant analysis (LDA).

In the SPRE model, for simplicity, we assume that the Laplace prior for each element of $\boldsymbol{\varepsilon}_i$ has the same scale parameter b. In general, we may assign different scale parameters for different elements of $\boldsymbol{\varepsilon}_i$. There also exist other possibilities between these two extremes. For example, if some elements of $\boldsymbol{\varepsilon}_i$ tend to be sparse together, they can be assigned the same scale parameter. This may be regarded as applying group LASSO (Yuan and Lin 2006) to the noise term. When this group version of the model is used, a benefit of the probabilistic formulation is that the group information, if not available, can be learned from data via a Bayesian approach based on some prior, e.g., Dirichlet process mixture (Antoniak 1974). However, such extension is beyond the scope of this paper and we will investigate it in our future research.

As a variant of the above model, we only have sparse noise in the data. Its formulation is defined as follows:

$$\begin{aligned}
&\mathbf{x}_i = \boldsymbol{\mu}_1 + \mathbf{W}_1\mathbf{z}_i + \boldsymbol{\varepsilon}_i \\
&\mathbf{y}_i = \boldsymbol{\mu}_2 + \mathbf{W}_2\mathbf{z}_i + \boldsymbol{\tau}_i \\
&\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&\varepsilon_{ij} \sim \mathcal{L}(0, b) \\
&\boldsymbol{\tau}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_2).
\end{aligned} \tag{2}$$

It is easy to see that as the covariance matrix $\boldsymbol{\Sigma}_1$ approaches the zero matrix, model (1) will degenerate to model (2) and hence model (2) can naturally be viewed as a special case of model (1). Moreover, as discussed above, model (1) will become the PLDA model when $b \to 0$. Thus, both model (2) and PLDA can be regarded as special cases of model (1).

Parameter Learning and Inference

Since model (1) is more general than model (2), we first discuss parameter learning and inference for model (1). Afterwards, we will briefly discuss the modification needed to make it work for model (2).

We note that the log-likelihood of model (1) cannot be calculated directly due to the Laplace prior on the noise $\{\boldsymbol{\varepsilon}_i\}$. So we resort to a variational method (Jordan et al. 1999). It is easy to show that the Laplace prior on each ε_{ij} can be rewritten as

$$\begin{aligned}
p(\varepsilon_{ij}) &= \frac{1}{2b} \exp\Big\{ - \frac{|\varepsilon_{ij}|}{b} \Big\} \\
&= \max_{\eta_{ij}>0} \big\{ \phi(\eta_{ij}) \mathcal{N}(\varepsilon_{ij} | 0, b^2\eta_{ij}) \big\},
\end{aligned} \tag{3}$$

where η_{ij} is a variational parameter and $\phi(\eta_{ij}) = \frac{1}{2}\exp\{-\frac{1}{2}\eta_{ij}\}\sqrt{2\pi\eta_{ij}}$. Eq. (3) holds since

$$\begin{aligned}
\phi(\eta_{ij})\mathcal{N}(\varepsilon_{ij}|0, b^2\eta_{ij}) &= \frac{1}{2b} \exp\Big\{ - \frac{1}{2}\Big(\eta_{ij} + \frac{\varepsilon_{ij}^2}{b^2\eta_{ij}}\Big)\Big\} \\
&\le \frac{1}{2b} \exp\Big\{ - \frac{|\varepsilon_{ij}|}{b} \Big\} = p(\varepsilon_{ij}),
\end{aligned}$$

where the equality holds when $\eta_{ij} = \frac{|\varepsilon_{ij}|}{b}$. Then, based on the independence property of $\{\varepsilon_{ij}\}_{j=1}^D$, we can obtain an alternative formulation for $p(\boldsymbol{\varepsilon}_i)$ as

$$p(\boldsymbol{\varepsilon}_i) = \max_{\boldsymbol{\eta}_i} \Big\{ \mathcal{N}(\boldsymbol{\varepsilon}_i | \mathbf{0}, b^2\boldsymbol{\Lambda}_i) \prod_{j=1}^D \phi(\eta_{ij}) \Big\},$$

where $\boldsymbol{\eta}_i = (\eta_{i1}, \ldots, \eta_{iD})^T$, $\boldsymbol{\Lambda}_i = \mathrm{diag}(\boldsymbol{\eta}_i)$ denotes a diagonal matrix with each diagonal element being the corresponding element in $\boldsymbol{\eta}_i$. Here for notational simplicity, we define new variables as

$$\tilde{\mathbf{x}}_i = \begin{pmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{pmatrix} \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}$$
$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_2 \end{pmatrix}.$$

Then, given the variational parameters $\{\eta_{ij}\}$, we can lower-bound the log-likelihood of the whole data set as

$$\begin{aligned} &\ln p(\mathbf{X}, \mathbf{Y}) \\ &= \sum_{i=1}^{n} \ln p(\tilde{\mathbf{x}}_i) \\ &= \sum_{i=1}^{n} \ln \int p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) p(\boldsymbol{\varepsilon}_i) d\boldsymbol{\varepsilon}_i \\ &\geq \sum_{i,j} \ln \phi(\eta_{ij}) + \sum_{i=1}^{n} \ln \int p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) p(\boldsymbol{\varepsilon}_i|\boldsymbol{\eta}_i) d\boldsymbol{\varepsilon}_i, \end{aligned} \quad (4)$$

where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ denotes the data matrix and $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ the label matrix. Even though the lower bound in Eq. (4) can be calculated analytically because

$$\begin{aligned} p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) &= \int p(\mathbf{z}_i) p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i, \mathbf{z}_i) d\mathbf{z}_i \\ &= \mathcal{N}(\boldsymbol{\mu} + \tilde{\boldsymbol{\varepsilon}}_i, \boldsymbol{\Sigma} + \mathbf{W}\mathbf{W}^T) \end{aligned}$$

and

$$\int p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) p(\boldsymbol{\varepsilon}_i|\boldsymbol{\eta}_i) d\boldsymbol{\varepsilon}_i = \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + b^2 \tilde{\boldsymbol{\Lambda}}_i + \boldsymbol{\Sigma}),$$

where $\tilde{\boldsymbol{\varepsilon}}_i = (\boldsymbol{\varepsilon}_i^T, \mathbf{0})^T$ and $\tilde{\boldsymbol{\Lambda}}_i = \mathrm{diag}((\boldsymbol{\eta}_i^T, \mathbf{0}))$ according to the linear-Gaussian model (Bishop 2006), the optimization problem is complicated and it is not easy for the optimization procedure to find a good local optimum since the number of variables to optimize is very large especially for high-dimensional data. Instead, we adopt an expectation-maximization (EM) algorithm to optimize the lower bound in Eq. (4). Here $\{\boldsymbol{\varepsilon}_i\}$ and $\{\mathbf{z}_i\}$ are hidden variables and the model parameters to be learned are denoted by $\boldsymbol{\Theta} = (\boldsymbol{\mu}, \mathbf{W}, b, \boldsymbol{\Sigma}, \{\eta_{ij}\})$. Suppose we are in the $(k+1)$st iteration of the EM algorithm and the current estimates of the model parameters are denoted by $\boldsymbol{\Theta}^{(k)} = (\boldsymbol{\mu}^{(k)}, \mathbf{W}^{(k)}, b^{(k)}, \boldsymbol{\Sigma}^{(k)}, \{\eta_{ij}^{(k)}\})$. We need to find $\boldsymbol{\Theta}^{(k+1)}$ to update the model parameters.

By introducing a variational distribution $q_1(\boldsymbol{\varepsilon}_i)$ on $\boldsymbol{\varepsilon}_i$, we can lower-bound $\ln \int p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i, \boldsymbol{\eta}_i) p(\boldsymbol{\varepsilon}_i|\boldsymbol{\eta}_i) d\boldsymbol{\varepsilon}_i$ via Jensen's inequality due to the concavity of the logarithm function

$$\begin{aligned} &\ln \int p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) p(\boldsymbol{\varepsilon}_i|\boldsymbol{\eta}_i) d\boldsymbol{\varepsilon}_i \\ &\geq \int q_1(\boldsymbol{\varepsilon}_i) \ln \frac{p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) p(\boldsymbol{\varepsilon}_i|\boldsymbol{\eta}_i)}{q_1(\boldsymbol{\varepsilon}_i)} d\boldsymbol{\varepsilon}_i \end{aligned}$$

subject to the constraint on $q_1(\boldsymbol{\varepsilon}_i)$ that

$$\int q_1(\boldsymbol{\varepsilon}_i) d\boldsymbol{\varepsilon}_i = 1.$$

By introducing a Lagrange multiplier λ, the Lagrangian is formulated as

$$\begin{aligned} F = &\int q_1(\boldsymbol{\varepsilon}_i) \ln \frac{p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) p(\boldsymbol{\varepsilon}_i|\boldsymbol{\eta}_i)}{q_1(\boldsymbol{\varepsilon}_i)} d\boldsymbol{\varepsilon}_i \\ &-\lambda \Big(\int q_1(\boldsymbol{\varepsilon}_i) d\boldsymbol{\varepsilon}_i - 1 \Big). \end{aligned}$$

We set the derivative of F with respect to $q(\boldsymbol{\varepsilon}_i)$ to zero and get

$$q_1(\boldsymbol{\varepsilon}_i) = \frac{p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) p(\boldsymbol{\varepsilon}_i|\boldsymbol{\eta}_i)}{\exp\{1+\lambda\}}.$$

Due to the constraint $\int q_1(\boldsymbol{\varepsilon}_i) d\boldsymbol{\varepsilon}_i = 1$, we can get the optimal $q_1^\star(\boldsymbol{\varepsilon}_i)$ as

$$\begin{aligned} &q_1^\star(\boldsymbol{\varepsilon}_i) \\ &= \frac{p(\mathbf{x}_i, \mathbf{y}_i|\boldsymbol{\varepsilon}_i) \mathcal{N}(\boldsymbol{\varepsilon}_i|\mathbf{0}, b^2 \boldsymbol{\Lambda}_i^{(k)})}{\int p(\mathbf{x}_i, \mathbf{y}_i|\boldsymbol{\varepsilon}_i) \mathcal{N}(\boldsymbol{\varepsilon}_i|\mathbf{0}, b^2 \boldsymbol{\Lambda}_i^{(k)}) d\boldsymbol{\varepsilon}_i} \\ &= \mathcal{N}\left(\mathbf{M}_i^{(k)} \left(\boldsymbol{\Omega}_{11}^{(k)} (\mathbf{x}_i - \boldsymbol{\mu}_1) + \boldsymbol{\Omega}_{12}^{(k)} (\mathbf{y}_i - \boldsymbol{\mu}_2) \right), \mathbf{M}_i^{(k)} \right), \end{aligned}$$

where $\begin{pmatrix} \boldsymbol{\Omega}_{11}^{(k)} & \boldsymbol{\Omega}_{12}^{(k)} \\ \boldsymbol{\Omega}_{21}^{(k)} & \boldsymbol{\Omega}_{22}^{(k)} \end{pmatrix} = (\boldsymbol{\Sigma}^{(k)} + \mathbf{W}^{(k)} (\mathbf{W}^{(k)})^T)^{-1}$ with $\boldsymbol{\Omega}_{11}^{(k)} \in \mathbb{R}^{D \times D}$ and $\boldsymbol{\Omega}_{12}^{(k)} \in \mathbb{R}^{D \times C}$, $\boldsymbol{\Lambda}_i^{(k)} = \mathrm{diag}(\eta_{i1}^{(k)}, \ldots, \eta_{iD}^{(k)})$, and $\mathbf{M}_i^{(k)} = \left(\boldsymbol{\Omega}_{11}^{(k)} + (b^{(k)})^{-2} (\boldsymbol{\Lambda}_i^{(k)})^{-1} \right)^{-1}$. Then we can get

$$\begin{aligned} &\ln \int p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i, \boldsymbol{\eta}_i) p(\boldsymbol{\varepsilon}_i|\boldsymbol{\eta}_i) d\boldsymbol{\varepsilon}_i \geq \int q_1^\star(\boldsymbol{\varepsilon}_i) \ln p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) d\boldsymbol{\varepsilon}_i \\ &+ \int q_1^\star(\boldsymbol{\varepsilon}_i) \ln \mathcal{N}(\boldsymbol{\varepsilon}_i|\mathbf{0}, b^2 \boldsymbol{\Lambda}_i) d\boldsymbol{\varepsilon}_i + \mathrm{Const}. \end{aligned} \quad (5)$$

The constant term in Eq. (5) has no effect on the learning of model parameters and hence can be omitted.

Combining Eq. (4) and (5), we obtain a lower bound of the log-likelihood of the whole data set as

$$\begin{aligned} &\ln p(\mathbf{X}, \mathbf{Y}) \\ &\geq \sum_{i,j} \ln \phi(\eta_{ij}) + \sum_{i=1}^{n} \int q_1^\star(\boldsymbol{\varepsilon}_i) \ln \mathcal{N}(\boldsymbol{\varepsilon}_i|\mathbf{0}, b^2 \boldsymbol{\Lambda}_i) d\boldsymbol{\varepsilon}_i \\ &\quad + \sum_{i=1}^{n} \int q_1^\star(\boldsymbol{\varepsilon}_i) \ln p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) d\boldsymbol{\varepsilon}_i + \mathrm{Const} \\ &= \sum_{i,j} -\frac{\eta_{ij}}{2} - \sum_{i=1}^{n} \frac{b^{-2} \mathrm{tr}(\boldsymbol{\Lambda}_i^{-1} \mathbb{E}[\boldsymbol{\varepsilon}_i \boldsymbol{\varepsilon}_i^T])}{2} - nD \ln b \\ &\quad + \sum_{i=1}^{n} \int q_1^\star(\boldsymbol{\varepsilon}_i) \ln p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) d\boldsymbol{\varepsilon}_i + \mathrm{Const}, \end{aligned} \quad (6)$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix and $\mathbb{E}[\cdot]$ denotes the expectation operator. Note that the fourth and fifth terms of the lower bound in Eq. (6) are independent of b and $\{\eta_{ij}\}$. By setting the derivative of the lower bound in Eq. (6) with respect to b and $\{\eta_{ij}\}$ to zero, we can obtain the

estimates of b and $\{\eta_{ij}\}$ as

$$\boldsymbol{\eta}_i^{(k+1)} = \frac{\sqrt{\mathrm{diag_m}(\mathbb{E}[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T])}}{b^{(k)}} \quad (7)$$

$$b^{(k+1)} = \sqrt{\frac{1}{nD}\sum_{i=1}^{n}\mathrm{tr}\Big(\big(\boldsymbol{\Lambda}_i^{(k+1)}\big)^{-1}\mathbb{E}[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T]\Big)}, \quad (8)$$

where $\boldsymbol{\Lambda}_i^{(k+1)} = \mathrm{diag}(\boldsymbol{\eta}_i^{(k+1)})$, $\mathrm{diag_m}(\cdot)$ extracts the diagonal elements of a square matrix as a column vector, and $\sqrt{\cdot}$ denotes the square root of a scalar, and also the elementwise square root of a vector.

We next discuss how to update $\boldsymbol{\mu}$, $\mathbf{W}$ and $\boldsymbol{\Sigma}$. Since only the fourth term of the lower bound in Eq. (6) depends on these parameters, it suffices to consider how to maximize the term $\sum_{i=1}^{n}\int q_1^\star(\boldsymbol{\varepsilon}_i)\ln p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i)d\boldsymbol{\varepsilon}_i$ only with respect to them. Since $p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) = \mathcal{N}(\boldsymbol{\mu}+\tilde{\boldsymbol{\varepsilon}}_i, \boldsymbol{\Sigma}+\mathbf{W}\mathbf{W}^T)$, in principle we can calculate the integral analytically by integrating out $\boldsymbol{\varepsilon}_i$ and use a gradient method to solve the optimization problem. However, the optimization problem is non-convex and the number of optimization variables is not small, making it hard to find a good local optimum. We again resort to the EM algorithm. By introducing a variational distribution $q_2(\mathbf{z}_i)$ on $\mathbf{z}_i$, we get

$$\begin{aligned}\ln p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) &= \ln\int p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i,\mathbf{z}_i)p(\mathbf{z}_i)d\mathbf{z}_i\\ &\geq \int q_2(\mathbf{z}_i)\ln\frac{p(\tilde{\mathbf{x}}_i|\mathbf{z}_i,\boldsymbol{\varepsilon}_i)p(\mathbf{z}_i)}{q_2(\mathbf{z}_i)}d\mathbf{z}_i.\end{aligned}$$

Similar to above, the optimal $q_2^\star(\mathbf{z}_i)$ is computed as

$$\begin{aligned}&q_2^\star(\mathbf{z}_i)\\ &= \frac{p(\mathbf{x}_i,\mathbf{y}_i|\boldsymbol{\varepsilon}_i,\mathbf{z}_i)p(\mathbf{z}_i)}{\int p(\mathbf{x}_i,\mathbf{y}_i|\boldsymbol{\varepsilon}_i,\mathbf{z}_i)p(\mathbf{z}_i)d\mathbf{z}_i}\\ &= \mathcal{N}\left(\mathbf{P}^{(k)}(\mathbf{W}^{(k)})^T\boldsymbol{\Sigma}^{(k)}\left(\tilde{\boldsymbol{x}}_i-\boldsymbol{\mu}^{(k)}-\tilde{\boldsymbol{\varepsilon}}_{\boldsymbol{i}}\right),\mathbf{P}^{(k)}\right), \quad (9)\end{aligned}$$

where $\mathbf{P}^{(k)} = \big(\mathbf{I}+(\mathbf{W}^{(k)})^T(\boldsymbol{\Sigma}^{(k)})^{-1}\mathbf{W}^{(k)}\big)^{-1}$. Then we can get

$$\begin{aligned}\ln p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i) \geq &-\frac{1}{2}\int q_2^\star(\mathbf{z}_i)\Big(\|\mathbf{x}_i-\boldsymbol{\mu}_1-\boldsymbol{\varepsilon}_i-\mathbf{W}_1\mathbf{z}_i\|^2_{\boldsymbol{\Sigma}_1^{-1}}\\ &+\|\mathbf{y}_i-\boldsymbol{\mu}_2-\mathbf{W}_2\mathbf{z}_i\|^2_{\boldsymbol{\Sigma}_2^{-1}}\Big)d\mathbf{z}_i-\frac{1}{2}\ln|\boldsymbol{\Sigma}_1|\\ &-\frac{1}{2}\ln|\boldsymbol{\Sigma}_2|+\mathrm{Const},\end{aligned}$$

where $\|\mathbf{x}\|_{\mathbf{M}} = \sqrt{\mathbf{x}^T\mathbf{M}\mathbf{x}}$ for a vector $\mathbf{x}$ and a square matrix $\mathbf{M}$ and $|\cdot|$ denotes the determinant of a square matrix. Then we can get the lower bound as

$$\begin{aligned}&2\sum_{i=1}^{n}\int q_1^\star(\boldsymbol{\varepsilon}_i)\ln p(\tilde{\mathbf{x}}_i|\boldsymbol{\varepsilon}_i)d\boldsymbol{\varepsilon}_i \geq -\sum_{i=1}^{n}\Big\{\|\mathbf{x}_i-\boldsymbol{\mu}_1\|^2_{\boldsymbol{\Sigma}_1^{-1}}\\ &+\mathrm{tr}(\boldsymbol{\Sigma}_1^{-1}\mathbb{E}[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T])+\mathrm{tr}(\mathbf{W}_1^T\boldsymbol{\Sigma}_1^{-1}\mathbf{W}_1\mathbb{E}[\mathbf{z}_i\mathbf{z}_i^T])+\ln|\boldsymbol{\Sigma}_1|\\ &-2(\mathbf{x}_i-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}\mathbf{W}_1\mathbb{E}[\mathbf{z}_i]+2\mathrm{tr}(\boldsymbol{\Sigma}_1^{-1}\mathbf{W}_1\mathbb{E}[\mathbf{z}_i\boldsymbol{\varepsilon}_i^T])\\ &+\mathrm{tr}(\mathbf{W}_2^T\boldsymbol{\Sigma}_2^{-1}\mathbf{W}_2\mathbb{E}[\mathbf{z}_i\mathbf{z}_i^T])-2(\mathbf{y}_i-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}_2^{-1}\mathbf{W}_2\mathbb{E}[\mathbf{z}_i]\\ &+\|\mathbf{y}_i-\boldsymbol{\mu}_2\|^2_{\boldsymbol{\Sigma}_2^{-1}}+\ln|\boldsymbol{\Sigma}_2|-2(\mathbf{x}_i-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}\mathbb{E}[\boldsymbol{\varepsilon}_i]\Big\}\\ &+\mathrm{Const},\end{aligned}$$

where $\|\cdot\|_2$ denotes the 2-norm of a vector. According to the form of $q_2^\star(\mathbf{z}_i)$ in Eq. (9), we rewrite it as $q_2^\star(\mathbf{z}_i) = \mathcal{N}(\mathbf{a}_i - \mathbf{B}\boldsymbol{\varepsilon}_i, \mathbf{C})$ where $\mathbf{B}$ and $\mathbf{C}$ are shared by all data points. Then we can calculate the expectation terms in the lower bound above as

$$\begin{aligned}\mathbb{E}[\mathbf{z}_i] &= \mathbb{E}[\mathbf{a}_i-\mathbf{B}\boldsymbol{\varepsilon}_i] = \mathbf{a}_i-\mathbf{B}\mathbb{E}[\boldsymbol{\varepsilon}_i]\\ \mathbb{E}[\mathbf{z}_i\mathbf{z}_i^T] &= \mathbf{C}+\mathbf{a}_i\mathbf{a}_i^T+\mathbf{B}\mathbb{E}[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T]\mathbf{B}^T-\mathbf{a}_i\mathbb{E}[\boldsymbol{\varepsilon}_i]^T\mathbf{B}^T\\ &\quad -\mathbf{B}\mathbb{E}[\boldsymbol{\varepsilon}_i]\mathbf{a}_i^T\\ \mathbb{E}[\boldsymbol{\varepsilon}_i\mathbf{z}_i^T] &= \mathbb{E}[\boldsymbol{\varepsilon}_i(\mathbf{a}_i-\mathbf{B}\boldsymbol{\varepsilon}_i)^T] = \mathbb{E}[\boldsymbol{\varepsilon}_i]\mathbf{a}_i^T-\mathbb{E}[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T]\mathbf{B}^T.\end{aligned}$$

We set the derivative of this lower bound with respect to $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \mathbf{W}_1, \mathbf{W}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$ to zero and get the update rules as

$$\begin{aligned}\boldsymbol{\mu}_1^{(k+1)} =&\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i-\mathbb{E}[\boldsymbol{\varepsilon}_i]-\mathbf{W}_1^{(k)}\mathbb{E}[\mathbf{z}_i])\\ \boldsymbol{\mu}_2^{(k+1)} =&\frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i-\mathbf{W}_2^{(k)}\mathbb{E}[\mathbf{z}_i])\\ \mathbf{W}_1^{(k+1)} =&\Big[\sum_{i=1}^{n}\Big((\mathbf{x}_i-\boldsymbol{\mu}_1^{(k+1)})\mathbb{E}[\mathbf{z}_i]^T-\mathbb{E}[\boldsymbol{\varepsilon}_i\mathbf{z}_i^T]\Big)\Big]\times\\ &\Big(\sum_{i=1}^{n}\mathbb{E}[\mathbf{z}_i\mathbf{z}_i^T]\Big)^{-1}\\ \mathbf{W}_2^{(k+1)} =&\Big[\sum_{i=1}^{n}(\mathbf{y}_i-\boldsymbol{\mu}_2^{(k+1)})\mathbb{E}[\mathbf{z}_i]^T\Big]\Big(\sum_{i=1}^{n}\mathbb{E}[\mathbf{z}_i\mathbf{z}_i^T]\Big)^{-1}\\ \boldsymbol{\Sigma}_1^{(k+1)} =&\frac{1}{n}\sum_{i=1}^{n}\Big\{\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^T+\mathbf{W}_1^{(k+1)}\mathrm{cov}(\mathbf{z}_i)(\mathbf{W}_1^{(k+1)})^T\\ &-\mathbf{W}_1^{(k+1)}\mathbf{B}\mathrm{cov}(\boldsymbol{\varepsilon}_i)-\mathrm{cov}(\boldsymbol{\varepsilon}_i)\mathbf{B}^T(\mathbf{W}_1^{(k+1)})^T\\ &+\mathrm{cov}(\boldsymbol{\varepsilon}_i)\Big\}\\ \boldsymbol{\Sigma}_2^{(k+1)} =&\frac{1}{n}\sum_{i=1}^{n}\Big\{\tilde{\mathbf{y}}_i\tilde{\mathbf{y}}_i^T+\mathbf{W}_2^{(k+1)}\mathrm{cov}(\mathbf{z}_i)(\mathbf{W}_2^{(k+1)})^T\Big\},\end{aligned}$$

where $\mathrm{cov}(\cdot)$ denotes the covariance operator, $\tilde{\mathbf{z}}_i = \mathbf{x}_i - \boldsymbol{\mu}_1^{(k+1)} - \mathbb{E}[\boldsymbol{\varepsilon}_i] - \mathbf{W}_1^{(k+1)}\mathbb{E}[\mathbf{z}_i]$ and $\tilde{\mathbf{y}}_i = \mathbf{y}_i - \boldsymbol{\mu}_2^{(k+1)} - \mathbf{W}_2^{(k+1)}\mathbb{E}[\mathbf{z}_i]$.

In summary, learning the model parameters involves a two-fold variational EM algorithm. In the E-step of the outer-fold EM, we first get a lower bound of the log-likelihood of the whole data set by introducing variational parameters $\{\eta_{ij}\}$ and then view $\boldsymbol{\varepsilon}_i$ as a hidden variable and calculate $q_1^\star(\boldsymbol{\varepsilon}_i)$. Then we update η_{ij} and b in the M-step of the outer-fold EM. For the inner-fold EM, we view $\mathbf{z}_i$ as a hidden variable and calculate $q_2^\star(\mathbf{z}_i)$ in the E-step and update $\boldsymbol{\mu}$, $\mathbf{W}$ and $\boldsymbol{\Sigma}$ in the M-step.

After learning the optimal model parameters, for each training data point $(\mathbf{x}_i, \mathbf{y}_i)$, we can approximate $p(\mathbf{x}_i, \mathbf{y}_i|\mathbf{z}_i)$ based on variational parameters $\{\boldsymbol{\eta}_i\}$ as

$$p(\mathbf{x}_i,\mathbf{y}_i|\mathbf{z}_i) \approx \mathcal{N}(\tilde{\mathbf{x}}_i|\boldsymbol{\mu}+\mathbf{W}\mathbf{z}_i, \boldsymbol{\Phi}_i),$$

where $\boldsymbol{\Phi}_i = \boldsymbol{\Sigma} + b^2\mathrm{diag}((\boldsymbol{\eta}_i, \mathbf{0}))$, and the posterior distri-

bution of the embedding $\{\mathbf{z}_i\}$ as

$$\begin{aligned} p(\mathbf{z}_i|\mathbf{x}_i,\mathbf{y}_i) &\approx \frac{\mathcal{N}(\tilde{\mathbf{x}}_i|\boldsymbol{\mu}+\mathbf{W}\mathbf{z}_i,\boldsymbol{\Phi}_i)p(\mathbf{z}_i)}{\int \mathcal{N}(\tilde{\mathbf{x}}_i|\boldsymbol{\mu}+\mathbf{W}\mathbf{z}_i,\boldsymbol{\Phi}_i)p(\mathbf{z}_i)d\mathbf{z}_i} \\ &= \mathcal{N}(\boldsymbol{\Psi}_i\mathbf{W}^T\boldsymbol{\Phi}_i^{-1}(\tilde{\mathbf{x}}_i-\boldsymbol{\mu}),\boldsymbol{\Psi}_i), \end{aligned}$$

where $\boldsymbol{\Psi}_i = \left(\mathbf{I}_d + \mathbf{W}^T\boldsymbol{\Phi}_i^{-1}\mathbf{W}\right)^{-1}$.

For a test data point $\mathbf{x}$, we first calculate the variational bound based on $\mathbf{x}$ only as

$$\begin{aligned} &\ln p(\mathbf{x}) \\ &= \ln \int p(\mathbf{x}|\boldsymbol{\varepsilon})p(\boldsymbol{\varepsilon})d\boldsymbol{\varepsilon} \\ &\geq \sum_i \ln\phi(\eta_i) + \ln\int p(\mathbf{x}|\boldsymbol{\varepsilon},\boldsymbol{\eta})p(\boldsymbol{\varepsilon}|\boldsymbol{\eta})d\boldsymbol{\varepsilon} \\ &\geq \sum_i \ln\phi(\eta_i) + \int q(\boldsymbol{\varepsilon})\ln\frac{p(\mathbf{x}|\boldsymbol{\varepsilon},\boldsymbol{\eta})p(\boldsymbol{\varepsilon}|\boldsymbol{\eta})}{q(\boldsymbol{\varepsilon})}d\boldsymbol{\varepsilon}, \end{aligned}$$

where $\boldsymbol{\eta} = (\eta_1,\ldots,\eta_D)^T$ denote the variational parameters. Let $\boldsymbol{\eta}^{(k)}$ denote the estimate of $\boldsymbol{\eta}$ in the kth iteration and $\boldsymbol{\Lambda}^{(k)} = \mathrm{diag}(\boldsymbol{\eta}^{(k)})$. The optimal $q^\star(\boldsymbol{\varepsilon})$ can be calculated as

$$\begin{aligned} q^\star(\boldsymbol{\varepsilon}) =& \frac{\mathcal{N}(\boldsymbol{\varepsilon}|\mathbf{0},b^2\boldsymbol{\Lambda}^{(k)})\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1+\boldsymbol{\varepsilon},\boldsymbol{\Sigma}_1+\mathbf{W}_1\mathbf{W}_1^T)}{\int \mathcal{N}(\boldsymbol{\varepsilon}|\mathbf{0},b^2\boldsymbol{\Lambda}^{(k)})\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1+\boldsymbol{\varepsilon},\boldsymbol{\Sigma}_1+\mathbf{W}_1\mathbf{W}_1^T)d\boldsymbol{\varepsilon}} \\ =& \mathcal{N}\Big(\boldsymbol{\Omega}^{(k)}(\mathbf{x}-\boldsymbol{\mu}_1),\boldsymbol{\Omega}^{(k)}\big(\mathbf{W}_1\mathbf{W}_1^T+\boldsymbol{\Sigma}_1\big)\Big), \end{aligned}$$

where $\boldsymbol{\Omega}^{(k)} = b^2\boldsymbol{\Lambda}^{(k)}\left(b^2\boldsymbol{\Lambda}^{(k)}+\mathbf{W}_1\mathbf{W}_1^T+\boldsymbol{\Sigma}_1\right)^{-1}$. We can then update $\boldsymbol{\eta} = (\eta_1,\ldots,\eta_D)^T$ as

$$\boldsymbol{\eta}^{(k+1)} = \frac{1}{b}\sqrt{\mathrm{diag}_m(\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T])}.$$

After learning the optimal $\boldsymbol{\eta}$, we approximate $p(\mathbf{z}|\mathbf{x})$ as

$$\begin{aligned} p(\mathbf{z}|\mathbf{x}) &\approx \frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1+\mathbf{W}_1\mathbf{z},\boldsymbol{\Sigma}_1+b^2\boldsymbol{\Lambda})p(\mathbf{z})}{\int \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1+\mathbf{W}_1\mathbf{z},\boldsymbol{\Sigma}_1+b^2\boldsymbol{\Lambda})p(\mathbf{z})d\mathbf{z}} \\ &= \mathcal{N}(\boldsymbol{\Psi}\mathbf{W}_1^T(\boldsymbol{\Sigma}_1+b^2\boldsymbol{\Lambda})^{-1}(\mathbf{x}-\boldsymbol{\mu}_1),\boldsymbol{\Psi}), \end{aligned}$$

where $\boldsymbol{\Psi} = \left(\mathbf{I}_d+\mathbf{W}_1^T\left(\boldsymbol{\Sigma}_1+b^2\boldsymbol{\Lambda}\right)^{-1}\mathbf{W}_1\right)^{-1}$. Then we can use information from $p(\mathbf{z}|\mathbf{x})$, i.e., its mean, to make prediction on the label of $\mathbf{x}$.

Moreover, we can also consider the inference problem in model (2) in which there is only sparse noise in the data. In this case, we set $\boldsymbol{\Sigma}_1$ to $\xi\mathbf{I}_D$ (e.g., $\xi = 10^{-4}$) instead of learning it. The learning procedure for the other model parameters remains unchanged.

Related Work

To the best of our knowledge, all the existing methods (Gao 2008; Wright et al. 2009; Xu, Caramanis, and Sanghavi 2010; Candes et al. 2011; Agarwal, Negahban, and Wainwright 2011) that consider sparse noise focus on the unsupervised setting exclusively. Extending these methods to incorporate label information appears to require nontrivial effort. On the other hand, it is very easy and natural to incorporate label information into our SPRE model. Gao (Gao 2008) has proposed the L1-PCA model with sparse noise and utilized variational method for inference. The methods in (Wright et al. 2009; Agarwal, Negahban, and Wainwright 2011; Candes et al. 2011) also consider sparse noise in the data generation process. Formulated under the regularization framework, these methods use l_1 regularization to identify the sparse noise and recover the low-rank data. Another method (Xu, Caramanis, and Sanghavi 2010) makes an additional assumption that if a data point is an outlier, then all its features are likely to be corrupted by sparse noise. An $l_{2,1}$ regularizer is used to model this 'group sparsity' phenomenon.

It is well known that the t distribution has heavy tail and hence is more robust than the Gaussian distribution in dealing with outliers. Archambeau et al. (Archambeau, Delannay, and Verleysen 2006) proposed a robust version of probabilistic principal component analysis (PPCA) and PLDA via utilizing the t distribution in the noise model instead of the Gaussian distribution and then extended it to a mixture model in (Archambeau, Delannay, and Verleysen 2008). Similar to (Archambeau, Delannay, and Verleysen 2006), Chen et al. (Chen, Martin, and Montague 2009) proposed a robust version of PPCA by utilizing the t distribution to deal with the missing data problem. Unlike these methods, we use the Laplace distribution here for sparse noise modeling to achieve model robustness.

Experiments

In this section, we evaluate our method empirically on some classification problems.

After embedding the data to a lower-dimensional space, we use a 1-nearest-neighbor classifier for classification. Similar to PLDA, the reduced dimensionality is set to $C-1$ where C is the number of classes in a classification problem. We compare SPRE with both PLDA (Bach and Jordan 2005) and robust PLDA (rPLDA) (Archambeau, Delannay, and Verleysen 2006). While the MLE solution of PLDA is identical to that of LDA, rPLDA improves over PLDA by using the t distribution instead of the Gaussian distribution to model noise. We consider two variants of SPRE, with SPRE-1 referring to the model in which there exist both Gaussian and sparse noises in the data, i.e., model (1), and SPRE-2 referring to the model in which there is only sparse noise in the data, i.e., model (2).

We first do experiments on three image databases, namely, the face databases PIE (Sim, Baker, and Bsat 2003) and AR and the object database COIL (Nene, Nayar, and Murase 1996). These databases are known to contain sparse noise due to the conditions under which the images were captured. While the sparse noise in PIE comes from variations in illumination, that in AR is due to occlusion as shown in Figure 1 while that in COIL is due to object rotation. For each configuration, we randomly select 50% of the data for training and the rest for testing. This random selection is repeated 10 times to obtain the mean classification error and standard deviation as reported in Table 1. Paired t-test at 5% significance level is applied to compare different methods and the best results are shown in bold for clarity. From the results, we can see that SPRE-1 consistently outperforms PLDA

and rPLDA, showing that the Laplace prior is effective in capturing sparse noise. We also note that SPRE-2 outperforms PLDA, showing the superiority of the Laplace prior over the Gaussian prior in modeling sparse noise. Comparing SPRE-2 with SPRE-1, SPRE-1 is generally better for these image datasets probably due to the existence of Gaussian noise in the data in addition to sparse noise.

Table 1: Average classification errors on the face and object datasets. The first row of each dataset records the mean and the second one is the standard deviation.

Dataset	PLDA	rPLDA	SPRE-1	SPRE-2
PIE	0.3592	0.1504	**0.1131**	0.2178
	0.0137	0.0178	0.0199	0.0198
AR	0.5165	0.2705	**0.2191**	0.4230
	0.0114	0.0241	0.0214	0.0192
COIL	0.2400	0.1446	0.0832	**0.0464**
	0.0142	0.0185	0.0120	0.0057

We further do more experiments on some widely used UCI datasets (Asuncion and Newman 2007). Unlike the image databases, we do not know whether sparse noise exists in the data of these datasets. The experimental settings are identical to those of the above experiments. Table 2 summarizes the results obtained. For some datasets, SPRE-2 is inferior to PLDA. A possible reason is that Gaussian noise is more common than sparse noise in these datasets. Moreover, SPRE-1, which models both Gaussian and sparse noises, is often comparable to or even better than rPLDA. This may suggest that using a combination of Gaussian and Laplace distributions is a favorable alternative to the t distribution for modeling noise in real data.

Table 2: Average classification errors on several UCI datasets. The first row of each dataset records the mean and the second one is the standard deviation.

Dataset	PLDA	rPLDA	SPRE-1	SPRE-2
sonar	0.4219	0.4205	**0.2644**	0.5041
	0.0689	0.0709	0.0078	0.0316
balance-scale	0.1849	0.3388	**0.1155**	0.6164
	0.0211	0.0362	0.0283	0.0574
ecoli	**0.2026**	0.2683	**0.2000**	**0.1974**
	0.0307	0.0417	0.0207	0.0101
glass	0.4188	0.4242	0.4577	**0.3785**
	0.0263	0.0372	0.0464	0.0390
hayes-roth	0.4179	0.4821	**0.3768**	0.4839
	0.0330	0.0188	0.0132	0.0530
iris	0.1086	0.1886	**0.0495**	0.0876
	0.0257	0.0528	0.0395	0.0183
mfeat-karhunen	0.0943	0.0897	**0.0593**	0.1607
	0.0089	0.0142	0.0092	0.0053
mfeat-pixel	0.2540	0.1264	0.1010	**0.0626**
	0.0161	0.0148	0.0070	0.0070
soybean	0.3188	**0.1893**	**0.1924**	0.2766
	0.0162	0.0099	0.0077	0.0115

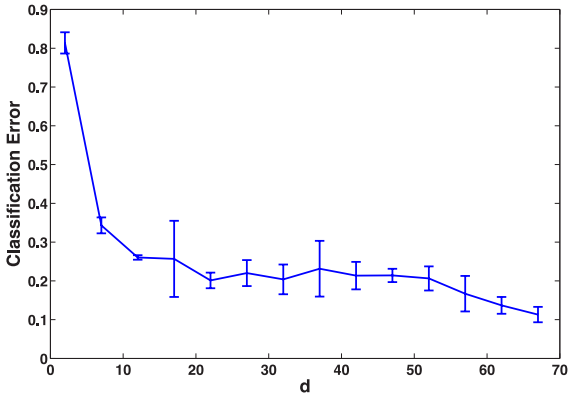

Figure 2: Average classification error against d from 2 to $C-1$ where C is the number of classes.

Moreover, we also conduct an experiment on the PIE dataset for the sensitivity analysis of the performance of the SPRE-1 model with respect to the reduced dimensionality. By varying the reduced dimensionality d from 2 to $C-1$ at an interval of 5, we record the average classification errors in Figure 2. From the results, we can see that the performance becomes better when the reduced dimensionality becomes larger, which suggests that setting d to $C-1$ is a good choice.

Conclusion

In this paper, we have proposed a supervised probabilistic robust embedding model that takes sparse noise into consideration under the supervised learning setting. Model learning is based on a two-fold variational EM algorithm in which the update of model parameters has analytical solution and hence has an advantage over gradient-based methods. One possible direction to extend the current work is to devise a mixture model for probabilistic robust embedding, as in (Tipping and Bishop 1999; Archambeau, Delannay, and Verleysen 2008), which is useful for more complex data. Another interesting direction is to incorporate group sparsity into our models by using Dirichlet process mixture to learn the group information.

Acknowledgment

Yu Zhang and Dit-Yan Yeung have been supported by General Research Fund 621310 from the Research Grants Council of Hong Kong.

References

Agarwal, A.; Negahban, S.; and Wainwright, M. J. 2011. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *Arxiv.1102.4807*.

Antoniak, C. E. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics* 2:1152–1174.

Archambeau, C.; Delannay, N.; and Verleysen, M. 2006. Robust probabilistic projections. In *Proceedings of the Twenty-*

Third International Conference on Machine Learning, 33–40.

Archambeau, C.; Delannay, N.; and Verleysen, M. 2008. Mixtures of robust probabilistic principal component analyzers. *Neurocomputing* 71(7–9):1274–1282.

Asuncion, A., and Newman, D. 2007. UCI machine learning repository.

Bach, F. R., and Jordan, M. I. 2005. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. New York: Springer.

Candes, E.; Li, X.; Ma, Y.; and Wright, J. 2011. Robust principal component analysis? *Journal of the ACM* 58(3):article 11.

Chen, T.; Martin, E.; and Montague, G. 2009. Robust probabilistic PCA with missing data and contribution analysis for outlier detection. *Computational Statistics & Data Analysis* 53(10):3706–3716.

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

Gao, J. 2008. Robust L1 principal component analysis and its Bayesian variational inference. *Neural Computation* 20(2):555–572.

Jordan, M. I.; Ghahramani, Z.; Jaakkola, T.; and Saul, L. K. 1999. An introduction to variational methods for graphical models. *Machine Learning* 37(2):183–233.

Martínez, A. M., and Benavente, R. 1998. The AR face database. Technical Report 24, CVC.

Nene, S. A.; Nayar, S. K.; and Murase, H. 1996. Columbia object image library (COIL-20). Technical Report 005, CUCS.

Sim, T.; Baker, S.; and Bsat, M. 2003. The CMU pose, illumination and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(12):1615–1618.

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

Wright, J.; Ganesh, A.; Rao, S.; Peng, Y.; and Ma, Y. 2009. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In Bengio, Y.; Schuurmans, D.; Lafferty, J.; Williams, C. K. I.; and Culotta, A., eds., *Advances in Neural Information Processing Systems 22*, 2080–2088.

Xu, H.; Caramanis, C.; and Sanghavi, S. 2010. Robust PCA via outlier pursuit. In Lafferty, J.; Williams, C. K. I.; Shawe-Taylor, J.; Zemel, R. S.; and Culotta, A., eds., *Advances in Neural Information Processing Systems 23*, 2496–2504.

Yuan, M., and Lin, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B* 68(1):49–67.