Probabilistic Principal Component Analysis

Grégoire DHIMOÏLA
ENS Paris-Saclay

Clément DUMAS
ENS Paris-Saclay

March - August 2024

1 Introduction

Dimensionality reduction is a critical tool in data analysis, enabling the simplification of high-dimensional datasets while retaining their essential structure. Among the most widely used methods is Principal Component Analysis (PCA), which identifies directions of maximum variance in the data. However, the original formulation of PCA lacks a probabilistic framework, limiting its application in scenarios involving missing data, denoising, or the need for a generative model. To address these limitations, Probabilistic Principal Component Analysis (PPCA) was proposed [Tipping and Bishop, 1999b], introducing a probabilistic model to dimensionality reduction.

This probabilistic approach facilitates tasks such as sample generation, missing data imputation or more robust denoising. Furthermore, it can model uncertainty in a principled manner, enabling for example outlier detection. Building upon this probabilistic framework, Mixtures of Probabilistic Principal Component Analyzers (MPPCA) [Tipping and Bishop, 1999a] combines the strengths of PPCA with clustering capabilities akin to Gaussian Mixture Models (GMM), enabling the modeling of complex, multimodal data distributions. The original formulation of PCA can not be extended to such mixture models.

In this report, we delve into the theoretical foundations and practical applications of PPCA and MPPCA. We present a revised approach to the EM algorithm introduced by Tipping and Bishop [1999b] to explicitly handle missing data. We quantify their performance on various tasks such as missing data reconstruction, compression and denoising, comparing them to standard PCA and other baseline methods. Using the real-world dataset MNIST, we also show how PPCA and MPPCA can be used for data visualization.

2 Principal Component Analysis

Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction. Given a data matrix $\mathbf{X} \in \mathbb{R}^{N\times d}$, PCA aims to find a set of q orthogonal vectors $\mathbf{w}_1, \dots, \mathbf{w}_q$ such that the variance of the data projected onto these vectors is maximized. These vectors are called the *principal components* of the data.

To find these principal components, PCA uses the Singular Value Decomposition (SVD) of the data matrix $\mathbf{X} = \mathbf{U\Sigma V}^T$, where $\mathbf{U} \in \mathbb{R}^{N\times N}$ and $\mathbf{V} \in \mathbb{R}^{d\times d}$ are orthogonal matrices and $\mathbf{\Sigma} \in \mathbb{R}^{N\times d}$ is a diagonal matrix. The principal components are then the columns of $\mathbf{V}$. They correspond to eigenvectors of the empirical covariance matrix. This has the nice property of minimizing the reconstruction error of the data : $\|\mathbf{X} - \mathbf{VV}^T\mathbf{X}\|_F^2$.

3 Probabilistic Principal Component Analysis

3.1 Model

Let $\mathbf{X} \in \mathbb{R}^{N\times d}$ be the data matrix, where N is the number of samples and d is the dimension of the data, and $\mathbf{x} \in \mathbf{X}$ be a data point. We assume that the data is generated by the following model:

$$\mathbf{x} = \mathbf{Wz} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^q$ is the latent variable, $\mathbf{W} \in \mathbb{R}^{d \times q}$ is the *loading matrix*, $\boldsymbol{\mu} \in \mathbb{R}^d$ is the offset of the model and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is isotropic gaussian noise. We assume that the latent variables are independent and identically distributed, following a standard normal distribution : $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and that the noise is independent of the latent variables.

From 1 and the assumption on the noise ϵ we can write the conditional distribution of the data given the latent variables as:

$$\mathbf{x}|\mathbf{z} \sim \mathcal{N}(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I}) \tag{2}$$

and derive the marginal distribution of the data as:

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}) \tag{3}$$

Given this model, the data is thus supposed to be Gaussian. To get the latent variables given some observed data, we can use the conditional distribution of the latent variables given the data, which is also Gaussian:

$$\mathbf{z}|\mathbf{x} \sim \mathcal{N}(\mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}), \sigma^2\mathbf{M}^{-1}) \tag{4}$$

where $\mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}$, and take the mean of this distribution as the estimate of $\mathbf{z}$.

3.2 Maximum likelihood estimation

Closed form solutions for the maximum likelihood estimates of the parameters can be derived from this simple Gaussian model, as shown by Tipping and Bishop [1999b]. The maximum likelihood estimate of the offset $\boldsymbol{\mu}$ is the empirical mean of the data. The maximum likelihood estimate $\mathbf{W}_{ML}$ of the loading matrix $\mathbf{W}$ is the matrix of the q eigenvectors corresponding to the q largest eigenvalues of the empirical covariance matrix Σ of the data while σ^2_{ML} is the average of the remaining eigenvalues.

$$\boldsymbol{\mu}_{ML} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \tag{5}$$

$$\sigma^2_{ML} = \frac{1}{d-q}\sum_{i=q+1}^{d} \lambda_i \tag{6}$$

$$\mathbf{W}_{ML} = \mathbf{U}_q \left(\boldsymbol{\Lambda}_q - \sigma^2_{ML}\mathbf{I}\right)^{1/2}\mathbf{R} \tag{7}$$

where $\mathbf{U}_q$ is the matrix of the q eigenvectors of Σ, corresponding to the q largest eigenvalues in the diagonal matrix $\boldsymbol{\Lambda}_q$, and $\mathbf{R}$ is an arbitrary orthogonal matrix - most often ignored and set to $\mathbf{I}$. The λ_i are the eigenvalues of Σ, sorted in decreasing order.

3.3 Expectation-Maximization algorithm

The ability to account for missing data is one of the main advantages of using a probabilistic model instead of the standard PCA.

The algorithm presented here is our own adaptation of the EM algorithm given by Tipping and Bishop [1999b] to the case of missing data in $\mathbf{X}$. They give this algorithm in the case where no data is missing, then use it in an experiment with missing data without details on how to adapt the algorithm.

For all $\mathbf{x}$ in $\mathbf{X}$, we isolate

- the observed entries $\mathbf{x}^{(o)}$,
- the missing entries $\mathbf{x}^{(m)}$,

The idea is to estimate the latent variable given the observed data and our current best guess for the missing data, then use these to update the estimate of the missing data. Missing data is dealt with as if it was a latent variable.

Latent variables for this algorithm are both the actual latent variables $\mathbf{z}$ and the missing data $\mathbf{x}^{(m)}$, model parameters are $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2. We start the E-step with some estimate of the latent variables and the model parameters, which we use to update the values of the latent variables. Then in the M-step, we use the updated latent variables to update the model parameters.

E-step Use the data $\mathbf{x}$ - made of observed and estimated missing data - to estimate the mean and covariance of the latent variable $\mathbf{z}|\mathbf{x}$:

$$\mathbb{E}[\mathbf{z}|\mathbf{x}] = \mathbf{M}^{-1}\mathbf{W}^T\left(x - \boldsymbol{\mu}\right) \tag{8}$$
$$\mathbf{Cov}[\mathbf{z}|\mathbf{x}] = \sigma^2\mathbf{M}^{-1} + \mathbb{E}[\mathbf{z}|\mathbf{x}]\mathbb{E}[\mathbf{z}|\mathbf{x}]^T \tag{9}$$

where $\mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}$. The expectation of the latent variable given the observed data can then be used to estimate the missing data:

$$\mathbb{E}[\mathbf{x}^{(m)}|\mathbf{x}^{(o)}] = \mathbf{W}^{(m)}\mathbb{E}[\mathbf{z}|\mathbf{x}] + \boldsymbol{\mu}^{(m)} \tag{10}$$

M-step Use these new values to update $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2:

$$\widetilde{\mathbf{x}} = \left[\begin{array}{c}\mathbf{x}^{(o)}\\ \mathbb{E}[\mathbf{x}^{(m)}|\mathbf{x}^{(o)}]\end{array}\right] \tag{11}$$

modulo some permutation of the entries of $\mathbf{x}$. We can then use the maximization step of the original EM algorithm to update the parameters of the model:

$$\widetilde{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N}\widetilde{\mathbf{x}}_n \tag{12}$$
$$\widetilde{\mathbf{W}} = \left[\sum_{n=1}^{N}\left(\widetilde{\mathbf{x}}_n - \widetilde{\boldsymbol{\mu}}\right)\mathbb{E}[\mathbf{z}|\mathbf{x}_n]^T\right]\left[\sum_{n=1}^{N}\mathbf{Cov}[\mathbf{z}|\mathbf{x}_n]\right] \tag{13}$$
$$\sigma^2 = \frac{1}{Nd}\sum_{n=1}^{N}\left\{\|\widetilde{\mathbf{x}}_n - \widetilde{\boldsymbol{\mu}}\|^2 - 2\mathbb{E}[\mathbf{z}|\mathbf{x}_n]^T\widetilde{\mathbf{W}}^T\left(\widetilde{\mathbf{x}}_n - \widetilde{\boldsymbol{\mu}}\right) + \mathrm{tr}\left(\mathbf{Cov}[\mathbf{z}|\mathbf{x}_n]\widetilde{\mathbf{W}}^T\widetilde{\mathbf{W}}\right)\right\} \tag{14}$$

3.4 Mixtures of PPCA

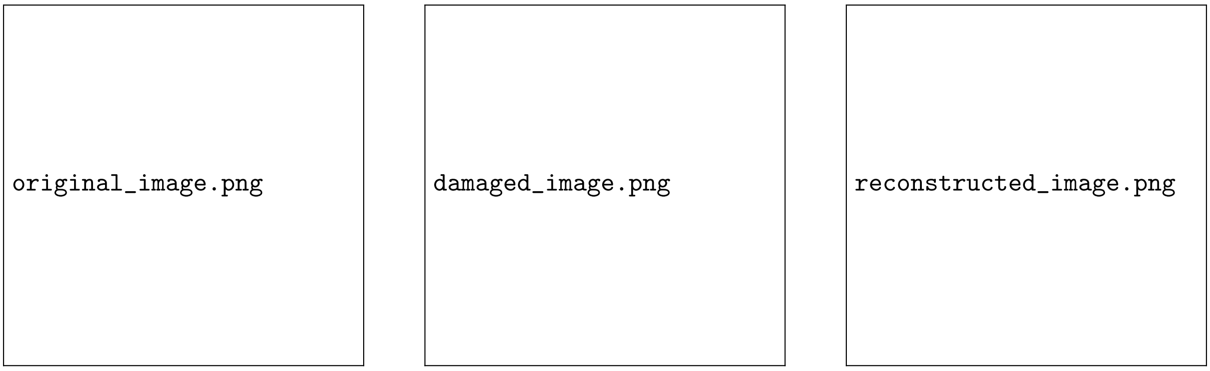
An advantage of using a probabilistic model is that it can be easily extended to a mixture model, unlike standard PCA. This is done by introducing M PPCA models, each with its own parameters $\mathbf{W}_m$, $\boldsymbol{\mu}_m$ and σ_m^2, and a mixing coefficient π_m for each model. This is very close to a Gaussian Mixture Model.

The EM algorithm for this model, described in Tipping and Bishop [1999a], updates π_m and $\boldsymbol{\mu}_m$ exactly as in a GMM formulation, and everything else follows naturally as described in that paper. To include missing data, we use the same modifications as in the single PPCA model described above. In practice, we never use the EM algorithm described for a single PPCA, we instead use a mixture of $M = 1$ components.

A full description of the algorithm is given in A.

4 Experiments

4.1 Missing data



(a) Original Image

(b) Damaged Image

(c) Reconstructed Image

Figure 1: Illustrative example of image reconstruction using PPCA. (a) Original image, (b) Image with 30% missing pixels, (c) Reconstructed image using PPCA.

In this experiment, we evaluate the capacity of PPCA to reconstruct missing data. We measure the proximity of the reconstructed data to ground truth and compare it to a baseline consisting of replacing missing data with their mean value. We proceed as follows:

Let $\mathbf{X}$ be the $N \times d$ data matrix, and $\mathbf{M}_p$ be a $N \times d$ binary mask matrix indicating missing values. Each entry of $\mathbf{M}_p$ is set to 0 with probability p. We then compute $\mathbf{X}_{\text{baseline}}$ and $\mathbf{X}_{\text{PPCA}}$ following the baseline and PPCA methods respectively. We evaluate the reconstruction error as the mean squared error between the reconstructed data and the ground truth:

$$\text{MSE} = \frac{1}{N \cdot d} \sum_{n=1}^{N} \sum_{i=1}^{d} \left(\mathbf{X}_{ni} - \mathbf{X}_{\text{reconstructed},ni}\right)^2 \tag{15}$$

with $\mathbf{X}_{\text{reconstructed}}$ being either $\mathbf{X}_{\text{baseline}}$ or $\mathbf{X}_{\text{PPCA}}$. We repeat this experiment for different proportions of missing data p and use a mixture of 10 PPCA models on the MNIST dataset. We report the results in Figure 2. Figure 1 shows an illustrative example of missing value reconstruction using PPCA.

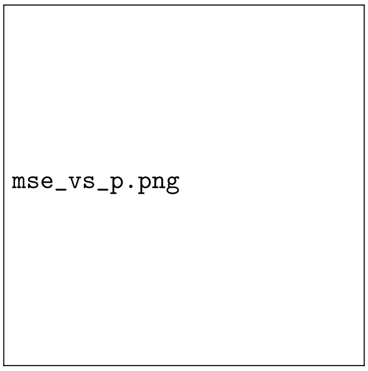
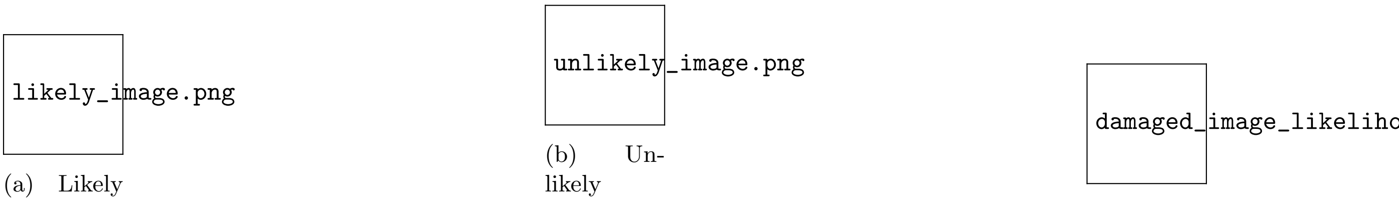


Figure 2: Mean Squared Error (MSE) with respect to the proportion of missing data p.

4.2 Outliers



(a) Likely images Likelihood: -1200, -1250

(b) Unlikely images Likelihood: -3000, -3100

(c) Damaged image Likelihood: -5000

Figure 3: Likelihood of MNIST images under the mixture of 10 PPCA models. (a) shows images with high likelihood scores, (b) images with low likelihood scores, and (e) is a severely damaged image with added noise.

One of the advantages of using a probabilistic model is having the ability to model uncertainty, which can be used for outlier detection. Given a new data point $\mathbf{x}$, we can compute its likelihood under the PPCA model using Equation (17) - and the corresponding formula in the mixture setting. If the likelihood is sufficiently low - e.g. by setting a threshold - the data point can be considered an outlier.

In our experiments, we evaluate the outlier detection performance of PPCA on a synthetic dataset. We generate a dataset with one cluster along with some outliers. We then fit a PPCA model to the data and compute the log-likelihood of each data point. We report the results in Figure 4. Figure 3 shows the likelihood of MNIST images under a mixture of 10 PPCA models as well as the likelihood of a damaged image.

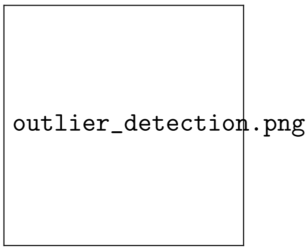


Figure 4: Outlier detection using PPCA. The red points are the detected outliers.

4.3 Data visualization

Data visualization is an important application of PCA. By projecting high-dimensional data onto a lower-dimensional space, we can gain insights into the structure and relationships within the data, and the visualization of the mean and principal components of the model can also provide useful insights. In this section, we demonstrate how PPCA can be used for data visualization and how MPPCA is better for visualizing multimodal data.

We use the MNIST dataset for this experiment. The MNIST dataset consists of images of handwritten digits, each of size 28x28 pixels. We first fit a PPCA model with 2 principal components to the dataset and project the data onto the 2-dimensional space spanned by these components. The resulting scatter plot is shown in Figure 5, where points correspond to images and are colored by their digit label.

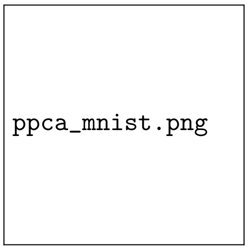
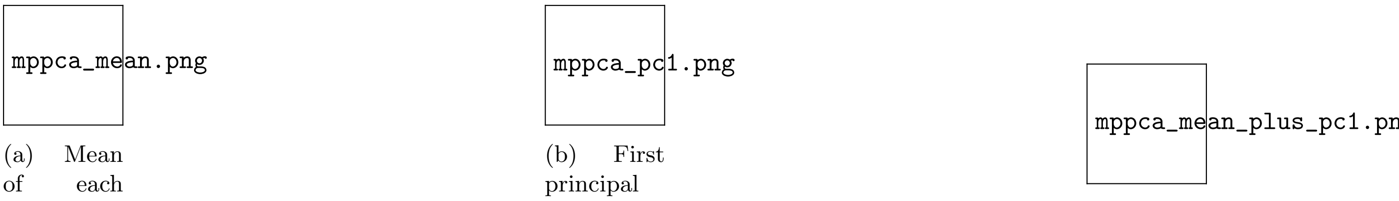


Figure 5: 2D visualization of the MNIST dataset using PPCA. Each point represents an image, colored by its digit label.

We then fit a mixture of 10 PPCA models (MPPCA) to the dataset, with each component having 10 principal components. We visualise the mean of each component in Figure 6a and the first principal components in ??. We compare it to the same visualisation using a simple PCA in ??. TODO : analyse the results.

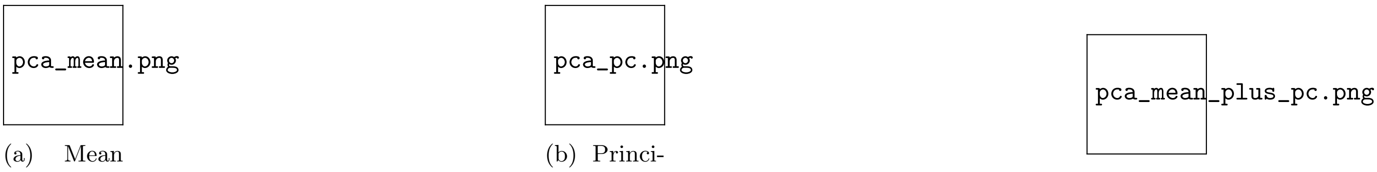


(a) Mean of each component $\boldsymbol{\mu}_m$

(b) First principal component $\mathbf{W}_m^1$

(c) $\boldsymbol{\mu}_m$ + $\mathbf{W}_m^1$

Figure 6: Visualization of the mean, first principal component, and mean + first principal component for each component in the MPPCA model.



(a) Mean of the data $\boldsymbol{\mu}$

(b) Principal components $\mathbf{W}^C$

(c) $\boldsymbol{\mu}_m$ + $\mathbf{W}_m^1$

Figure 7: Visualization of the mean, principal components, and mean + principal components for a PCA of the MNIST dataset.

Additionally, we can use the PPCA models to generate new samples, since in this models, latents are supposed to follow standard normal distributions. By sampling from the latent space and transforming the samples back to the original space, we can generate new images that resemble the original data. Figure 8 show examples of images generated using the MPPCA from the previous visualization.

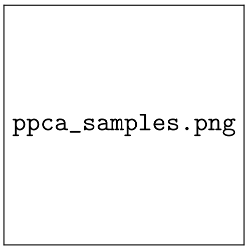


Figure 8: Images sampled from the MPPCA models.

These generated samples further illustrate the capability of PPCA to model the underlying structure of the data.

In summary, PPCA and MPPCA are powerful tools for data visualization and sample generation. They provide insights into the structure of high-dimensional data and enable the generation of new samples that follow the original data distribution.

4.4 Data compression

Data compression is a key application of dimensionality reduction methods. We evaluate the performance of PPCA and MPPCA on the task of data compression and compare it to standard PCA. We measure the reconstruction error as a function of the compression rate and report the results in Figure 9. We use the MNIST dataset for this experiment.

Figure 9: Reconstruction error as a function of the compression rate for PCA, PPCA and MPPCA.

4.5 Denoising

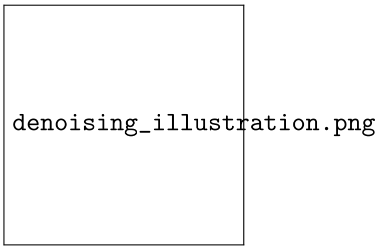


Figure 10: Illustrative example of image denoising using PPCA.

In this experiment, we evaluate the performance of PPCA on the task of data denoising. We compare the performance of these methods to standard PCA and other baseline methods. We proceed as follows:

- Generate a noisy version of the MNIST dataset by adding isotropic Gaussian noise to the images.
- Fit a PPCA model to the noisy data and reconstruct the images. We use a mixture of 10 PPCA models with a high number of components $q = \frac{d}{2}$.
- Measure the reconstruction error and compare it to the baseline methods.

We report reconstruction error as a function of the strenght of the noise σ_{noise} in Figure 11. Figure 10 shows an illustrative example of image denoising using PPCA.

The intuition behind why PPCA is better than PCA on this task is that the denoising capabilities of PCA arise from the fact that low variance directions can be considered as containing noise while all the interesting signal is stored in high variance directions. However, if the noise is full rank, the first principal components will also contain noise. PPCA can separate the noise from the signal by modeling the noise explicitly in the model.

denoising.png

Figure 11: Reconstruction error as a function of the noise strength for PCA, PPCA and a mixture of 10 PPCA on the MNIST dataset.

5 Discussion

This report has presented a comprehensive overview of Probabilistic Principal Component Analysis (PPCA) and its extension to Mixtures of Probabilistic Principal Component Analyzers (MPPCA). We have presented both the theoretical framework and practical applications of these methods, particularly in handling noisy or missing data, outlier detection and data visualization.

Strengths of PPCA and MPPCA One of the principal strengths of PPCA is that it extends the utility of standard PCA to applications requiring a probabilistic framework. This probabilistic basis enables PPCA to:

- Handle noisy or missing data seamlessly through the explicit modeling of noise and the Expectation-Maximization (EM) algorithm,
- Quantify uncertainty in data, which is beneficial for outlier detection and robustness analysis,
- Provide a solid foundation for further extensions, such as MPPCA, which can model complex, multimodal data distributions,
- Generate new samples from the model.

The flexibility of MPPCA, akin to Gaussian Mixture Models, allows clustering of data while simultaneously reducing dimensionality. This capability is particularly valuable for tasks involving high-dimensional and heterogeneous datasets, as demonstrated in our experiments on MNIST.

Challenges and Limitations Despite its advantages, PPCA and its mixture-based extension are not without limitations. Key challenges include:

- **Model Assumptions:** The assumption of Gaussianity in PPCA may limit its applicability to non-Gaussian data. Extending the model to accommodate non-Gaussian latent variables could improve its versatility.
- **Parameter Sensitivity:** The choice of the number of principal components to consider q and the number of mixture components M significantly affects performance, necessitating careful model selection and validation.

References

Michael E Tipping and Christopher M Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999a.

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999b.

A EM algorithm for MPPCA with missing data

We present here the full EM algorithm for Mixtures of Probabilistic Principal Component Analyzers (MPPCA) with missing data.

Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ be the data matrix, where each $\mathbf{x}_n$ is a data point separated into observed entries $\mathbf{x}_n^{(o)}$ and missing entries $\mathbf{x}_n^{(m)}$. Let M be the number of PPCA models in the mixture, $\mathbf{Z}^m \in \mathbb{R}^{N \times q}$ be the latent variable matrix for the m-th model, $\mathbf{W}_m \in \mathbb{R}^{d \times q}$ be the loading matrix, $\boldsymbol{\mu}_m \in \mathbb{R}^d$ be the offset, $\sigma_m^2 \in \mathbb{R}^M$ be the noise variance, and π_m be the mixing coefficient for the m-th model.

Following the single PPCA model, we write the conditional distribution of the data given the m-th model as:

$$\mathbf{x}|m \sim \mathcal{N}(\mathbf{W}_m \mathbf{z}^m + \boldsymbol{\mu}_m, \sigma_m^2 \mathbf{I}) \tag{16}$$

and the overall distribution of the data as:

$$p(\mathbf{x}_n) = \sum_{m=1}^{M} \pi_m p(\mathbf{x}_n|m) \tag{17}$$

Let R_{nm} be the posterior responsibility of the m-th component for the generation of the n-th data point:

$$R_{nm} = \frac{p(\mathbf{x}_n|m)\pi_m}{p(\mathbf{x}_n)} \tag{18}$$

Contrary to the single PPCA model, we will use a two stage EM algorithm. If we follow the same procedure as in Section 3.3, we will get coupled equations for the updates of $\boldsymbol{\mu}_m$ and $\mathbf{W}_m$, which makes the algorithm intractable. Instead, we will use a two stage EM algorithm adapted from Tipping and Bishop [1999a], condensed as follows:

First, update the responsibilities R_{nm} using Equation (18). Then, do the E-step from the single PPCA model to update the estimate of the missing data :

$$\mathbb{E}[\mathbf{z}^m|\mathbf{x}] = \mathbf{M}_m^{-1}\mathbf{W}_m^T (x - \boldsymbol{\mu}_m) \tag{19}$$

$$\mathbf{Cov}[\mathbf{z}^m|\mathbf{x}] = \sigma_m^2 \mathbf{M}_m^{-1} + \mathbb{E}[\mathbf{z}^m|\mathbf{x}]\mathbb{E}[\mathbf{z}^m|\mathbf{x}]^T \tag{20}$$

where $\mathbf{M}_m = \mathbf{W}_m^T \mathbf{W}_m + \sigma_m^2 \mathbf{I}$. The expectation of the latent variable given the observed data can then be used to estimate the missing data:

$$\mathbb{E}[\mathbf{x}_n^{(m)}|\mathbf{x}_n^{(o)}] = \sum_{m=1}^{M} R_{nm} \left(\mathbf{W}_m^{(m)} \mathbb{E}[\mathbf{z}^m|\mathbf{x}] + \boldsymbol{\mu}_m^{(m)} \right) \tag{21}$$

With this new values for the data, update π_m and $\boldsymbol{\mu}_m$ as follows:

$$\pi_m = \frac{1}{N} \sum_{n=1}^{N} R_{nm} \tag{22}$$

$$\boldsymbol{\mu}_m = \frac{\sum_{n=1}^{N} R_{nm} \mathbf{x}_n}{\sum_{n=1}^{N} R_{nm}} \tag{23}$$

in order to compute the following covariance matrix:

$$\Sigma_m = \frac{1}{\pi_m N} \sum_{n=1}^{N} R_{nm} \left(\mathbf{x}_n - \boldsymbol{\mu}_m\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_m\right)^T \tag{24}$$

which can finally be used to update the loading matrix and the noise variance with Equations (6) and (7) respectively.