Dimensionality Reduction: Probabilistic PCA and Factor Analysis

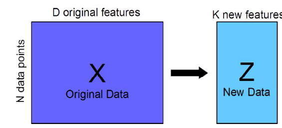
Piyush Rai
IIT Kanpur

Probabilistic Machine Learning (CS772A)

Feb 3, 2016

Dimensionality Reduction

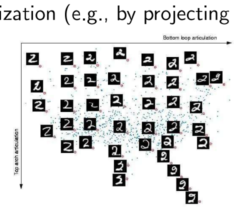
- Given: an $N \times D$ data matrix $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, with $\boldsymbol{x}_n \in \mathbb{R}^D$
- Want a lower-dim. rep. as an $N \times K$ matrix $\mathbf{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$, $\boldsymbol{z}_n \in \mathbb{R}^K$
- $K \ll D \Rightarrow$ dimensionality reduction

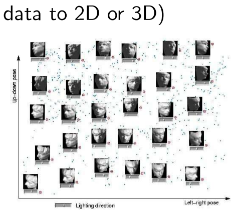


- Learns a new feature representation of data with reduced dimensionality
- Don't want to lose much information about $\mathbf{X}$ while doing this: want to preserve the interesting/useful information in $\mathbf{X}$ and discard the rest
- Various ways to quantify what "useful" is (depends on what we want to learn)

Why Dimensionality Reduction?

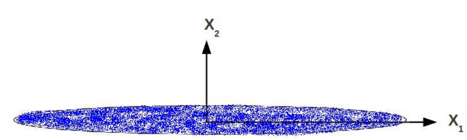
- To compress data by reducing dimensionality. E.g., representing each image in a large collection as a linear combination of a small set of "template" images
  - Also sometimes called dictionary learning (can also be used for other types of data, e.g., speech signals, text-documents, etc.)
- Visualization (e.g., by projecting high-dim data to 2D or 3D)





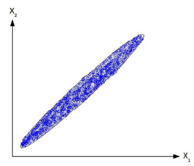
- To make learning algorithms run faster
- To reduce overfitting problem caused by high-dimensional data

Dimensionality Reduction: A Simple Illustration



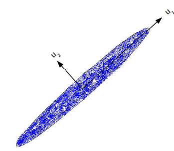
- Consider this 2 dimensional data
- Each example $\boldsymbol{x}$ has 2 features $\{x_1, x_2\}$
- Consider ignoring the feature x_2 for each example
- Each 2-dimensional example $\boldsymbol{x}$ now becomes 1-dimensional $\boldsymbol{x} = \{x_1\}$
- Are we losing much information by throwing away x_2?
- No. Most of the data spread is along x_1 (very little variance along x_2)

Dimensionality Reduction: A Simple Illustration



- Consider this 2 dimensional data
- Each example $\boldsymbol{x}$ has 2 features $\{x_1, x_2\}$
- Consider ignoring the feature x_2 for each example
- Each 2-dimensional example $\boldsymbol{x}$ now becomes 1-dimensional $\boldsymbol{x} = \{x_1\}$
- Are we losing much information by throwing away x_2?
- Yes. The data has substantial variance along both features (i.e., both axes)

Dimensionality Reduction: A Simple Illustration

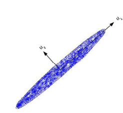


- Now consider a change of axes (the co-ordinate system)
- Each example $\boldsymbol{x}$ has 2 features $\{u_1, u_2\}$
- Consider ignoring the feature u_2 for each example
- Each 2-dimensional example $\boldsymbol{x}$ now becomes 1-dimensional $\boldsymbol{x} = \{u_1\}$
- Are we losing much information by throwing away u_2?
- No. Most of the data spread is along u_1 (very little variance along u_2)

Review: Principal Component Analysis

Principal Component Analysis (PCA)

- Based on identifying the **Principal Components** in the data
- Principal Components (PC): Directions of high variance in the data
- Roughly speaking, PCA does a change of axes that represent the data



- First PC: Direction of the highest variance
- Second PC: Direction of next highest variability (**orthogonal** to the first PC)
- Subsequent PCs: Other directions of highest variability (in decreasing order)
- Note: All principal components are orthogonal to each other
- PCA: Take top K PC's and project the data along those

PCA: Finding the Principal Components

- Given: N examples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, each example $\boldsymbol{x}_n \in \mathbb{R}^D$
- Goal: Project the data from D dimensions to K dimensions ($K < D$)
- Want projection directions s.t. the projected data has maximum variance
  - Note: This is equivalent to minimizing the reconstruction error, i.e., error in reconstructing the original data from its projections
- Let $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_D$ be the principal components, assumed to be:
  - Orthogonal: $\boldsymbol{w}_i^\top \boldsymbol{w}_j = 0$ if $i \neq j$, Orthonormal: $\boldsymbol{w}_i^\top \boldsymbol{w}_i = 1$
- Each principal component is a vector of size $D \times 1$
- We want only the first K principal components

PCA: Finding the Principal Components

- Projection of a data point $\boldsymbol{x}_n$ along $\boldsymbol{w}_1$: $\boldsymbol{w}_1^\top \boldsymbol{x}_n$
- Projection of the mean $\bar{\boldsymbol{x}}$ along $\boldsymbol{w}_1$: $\boldsymbol{w}_1^\top \bar{\boldsymbol{x}}$ (where $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{n=1}^N \boldsymbol{x}_n$)
- Variance of the projected data (along projection direction $\boldsymbol{u}_1$):

$$\frac{1}{N}\sum_{n=1}^N \left\{ \boldsymbol{w}_1^\top \boldsymbol{x}_n - \boldsymbol{w}_1^\top \bar{\boldsymbol{x}} \right\}^2 = \boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1$$

where $\boldsymbol{S}$ is the data covariance matrix defined as

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^N (\boldsymbol{x}_n - \bar{\boldsymbol{x}})(\boldsymbol{x}_n - \bar{\boldsymbol{x}})^\top$$

- Want to have $\boldsymbol{w}_1$ that maximizes the projected data variance $\boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1$
  - Subject to the constraint: $\boldsymbol{w}_1^\top \boldsymbol{w}_1 = 1$
  - We will introduce a Lagrange multiplier λ_1 for this constraint

PCA: Finding the Principal Components

- Objective function: $\boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1 + \lambda_1(1 - \boldsymbol{w}_1^\top \boldsymbol{w}_1)$
- Taking derivative w.r.t. $\boldsymbol{u}_1$ and setting it to zero gives:

$$\boldsymbol{S}\boldsymbol{w}_1 = \lambda_1 \boldsymbol{w}_1$$

- This is the eigenvalue equation
  - $\boldsymbol{w}_1$ must be *an* eigenvector of $\boldsymbol{S}$ (and λ_1 the corresponding eigenvalue)
- But there are multiple eigenvectors of $\boldsymbol{S}$. Which one is $\boldsymbol{w}_1$?
- Consider $\boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1 = \boldsymbol{w}_1^\top \lambda_1 \boldsymbol{w}_1 = \lambda_1$ (using $\boldsymbol{w}_1^\top \boldsymbol{w}_1 = 1$)
- We know that the projected data variance $\boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1 = \lambda_1$ is maximum
  - Thus λ_1 should be the largest eigenvalue
  - Thus $\boldsymbol{w}_1$ is the first (top) eigenvector of $\boldsymbol{S}$ (with eigenvalue λ_1) $\Rightarrow$ the first principal component (direction of highest variance in the data)
- Subsequent PC's are given by the subsequent eigenvectors of $\boldsymbol{S}$

PCA: The Algorithm

- Compute the mean of the data

$$\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{n=1}^N \boldsymbol{x}_n$$

- Compute the sample covariance matrix (using the mean subtracted data)

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^N (\boldsymbol{x}_n - \bar{\boldsymbol{x}})(\boldsymbol{x}_n - \bar{\boldsymbol{x}})^\top$$

- Do the eigenvalue decomposition of the $D \times D$ matrix $\boldsymbol{S}$
- Take the top K eigenvectors (corresponding to the top K eigenvalues)
- Call these $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K$ (s.t. $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_{K-1} \geq \lambda_K$)
- $\mathbf{W} = [\boldsymbol{w}_1 \; \boldsymbol{w}_2 \; \ldots \; \boldsymbol{w}_K]$ is the projection matrix of size $D \times K$
- Projection of each example $\boldsymbol{x}_n$ is computed as $\boldsymbol{z}_n = \mathbf{W}^\top \boldsymbol{x}_n$
  - $\boldsymbol{z}_n$ is a $K \times 1$ vector (also called the embedding of $\boldsymbol{x}_n$)

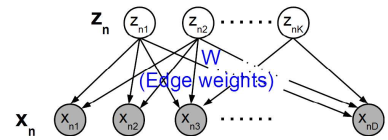
Now on to Probabilistic PCA

Probabilistic PCA (PPCA)

- Assume the following generative model for each observation $\boldsymbol{x}_n$

$$\boldsymbol{x}_n = \mathbf{W}\boldsymbol{z}_n + \epsilon_n$$

- Note: We'll assume data to be centered, otherwise $\boldsymbol{x}_n = \boldsymbol{\mu} + \mathbf{W}\boldsymbol{z}_n + \epsilon_n$
- Think of it as low dimensional $\boldsymbol{z}_n \in \mathbb{R}^K$ "generating" a higher-dimensional $\boldsymbol{x}_n \in \mathbb{R}^D$ via a mapping matrix $\mathbf{W} \in \mathbb{R}^{D\times K}$, plus some noise $\epsilon_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_D)$



- Intuitively, this generative model is "inverse" of what the traditional PCA does. Here we assume a latent low-dim $\boldsymbol{z}_n$ that "generates" the high-dim $\boldsymbol{x}_n$ via the mapping $\mathbf{W}$ (plus adding some noise)
- A directed graphical model linking $\boldsymbol{z}_n$ and $\boldsymbol{x}_n$ via "edge weights" $\mathbf{W}$

Interpreting Probabilistic PCA

- Can also write $\boldsymbol{x}_n = \mathbf{W}\boldsymbol{z}_n + \epsilon_n$ as each example $\boldsymbol{x}_n$ being a linear comb. of columns of $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$, plus some example-specific random noise ϵ_n

$$\boldsymbol{x}_n = \sum_{k=1}^{K} \boldsymbol{w}_k z_{nk} + \epsilon_n$$

- The K columns of $\mathbf{W}$ (each $\mathbb{R}^D$) are like "prototype vectors" shared by all examples. Each $\boldsymbol{x}_n$ is a linear combination of these vectors (the combination coefficients are given by $\boldsymbol{z}_n \in \mathbb{R}^K$ which is basically the low-dim rep. of $\boldsymbol{x}_n$).
- Some examples:
  - In case of images, columns of $\mathbf{W}$ would correspond to "basis images"
  - In case of text documents, columns of $\mathbf{W}$ (with non-negativity imposed on it) would correspond to "topics" in the corpus

Probabilistic PCA

- Since noise $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$ is Gaussian, the conditional distrib. of $\boldsymbol{x}_n$

$$p(\boldsymbol{x}_n | \boldsymbol{z}_n, \mathbf{W}, \sigma^2) = \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \sigma^2 \mathbf{I}_D)$$

- Given a set of observations $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, the goal is to learn $\mathbf{W}$ and the low-dim. representation of data, i.e., $\mathbf{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$
- Assume a Gaussian prior on the low-dimensional latent representation, i.e.,

$$p(\boldsymbol{z}_n) = \mathcal{N}(0, \mathbf{I}_K)$$

- Using the equation for marginal of Gaussians (lecture-2 and PRML 2.115), the marginal distribution of $\boldsymbol{x}_n$ (after integrating out latent variables $\boldsymbol{z}_n$)

$$p(\boldsymbol{x}_n) \text{ or } p(\boldsymbol{x}_n | \mathbf{W}, \sigma^2) = \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I}_D)$$

- Note: Cov. matrix of $p(\boldsymbol{x}_n)$ now has DK parameters instead of $D(D-1)/2$
- Thus PPCA also allows a more parsimonious parameterization of $p(\boldsymbol{x}_n)$

Probabilistic PCA

- Consider the marginal likelihood

$$p(\boldsymbol{x}_n|\mathbf{W},\sigma^2) = \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)$$

- Can do MLE on this, or use EM with latent variables $\boldsymbol{z}_n$ (is simpler)
- To do EM, we will need the posterior over latent vars. $\boldsymbol{z}_n$ in the E step
- The posterior over $\boldsymbol{z}_n$ (using result from lecture-2 and PRML 2.116)

$$p(\boldsymbol{z}_n|\boldsymbol{x}_n,\mathbf{W}) = \mathcal{N}(\mathbf{M}^{-1}\mathbf{W}^\top\boldsymbol{x}_n, \sigma^2\mathbf{M}^{-1})$$

where $\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}_K$ (a $K \times K$ matrix)

EM for Probabilistic PCA

- Observed data: $\mathbf{X} = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$, latent variable: $\mathbf{Z} = \{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_N\}$
- Parameters: $\mathbf{W},\sigma^2$
- The complete data log -likelihood

$$\begin{aligned}\log p(\mathbf{X},\mathbf{Z}|\mathbf{W},\sigma^2) = \log\prod_{n=1}^N p(\boldsymbol{x}_n,\boldsymbol{z}_n|\mathbf{W},\sigma^2) &= \log\prod_{n=1}^N p(\boldsymbol{x}_n|\boldsymbol{z}_n,\mathbf{W},\sigma^2)p(\boldsymbol{z}_n) \\ &= \sum_{n=1}^N\{\log p(\boldsymbol{x}_n|\boldsymbol{z}_n,\mathbf{W},\sigma^2) + \log p(\boldsymbol{z}_n)\}\end{aligned}$$

- Plugging in, simplifying and ignoring the constants, we get

$$\log p(\mathbf{X},\mathbf{Z}|\mathbf{W},\sigma^2) = \sum_{n=1}^N\left\{\frac{D}{2}\log\sigma^2 + \frac{1}{2\sigma^2}||\boldsymbol{x}_n||^2 - \frac{1}{\sigma^2}\boldsymbol{z}_n^\top\mathbf{W}^\top\boldsymbol{x}_n + \frac{1}{2\sigma^2}\mathrm{tr}(\boldsymbol{z}_n\boldsymbol{z}_n^\top\mathbf{W}^\top\mathbf{W}) + \frac{1}{2}\mathrm{tr}(\boldsymbol{z}_n\boldsymbol{z}_n^\top)\right\}$$

- The expected complete data log-likelihood $\mathbb{E}[\log p(\mathbf{X},\mathbf{Z}|\mathbf{W},\sigma^2)]$

$$= \sum_{n=1}^N\left\{\frac{D}{2}\log\sigma^2 + \frac{1}{2\sigma^2}||\boldsymbol{x}_n||^2 - \frac{1}{\sigma^2}\mathbb{E}[\boldsymbol{z}_n]^\top\mathbf{W}^\top\boldsymbol{x}_n + \frac{1}{2\sigma^2}\mathrm{tr}(\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]\mathbf{W}^\top\mathbf{W}) + \frac{1}{2}\mathrm{tr}(\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top])\right\}$$

EM for Probabilistic PCA

- The expected complete data log-likelihood

$$\sum_{n=1}^N\left\{\frac{D}{2}\log\sigma^2 + \frac{1}{2\sigma^2}||\boldsymbol{x}_n||^2 - \frac{1}{\sigma^2}\mathbb{E}[\boldsymbol{z}_n]^\top\mathbf{W}^\top\boldsymbol{x}_n + \frac{1}{2\sigma^2}\mathrm{tr}(\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]\mathbf{W}^\top\mathbf{W}) + \frac{1}{2}\mathrm{tr}(\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top])\right\}$$

- We need two terms: $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$
- We have already seen that

$$p(\boldsymbol{z}_n|\boldsymbol{x}_n,\mathbf{W}) = \mathcal{N}(\mathbf{M}^{-1}\mathbf{W}^\top\boldsymbol{x}_n, \sigma^2\mathbf{M}^{-1}) \qquad \text{where } \mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}_K$$

- From posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n,\mathbf{W})$, we can easily compute the required expectations

$$\begin{aligned}\mathbb{E}[\boldsymbol{z}_n] &= \mathbf{M}^{-1}\mathbf{W}^\top\boldsymbol{x}_n \\ \mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top] &= \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \mathrm{cov}(\boldsymbol{z}_n) = \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \sigma^2\mathbf{M}^{-1}\end{aligned}$$

- Taking the derivative of $\mathbb{E}[\log p(\mathbf{X},\mathbf{Z}|\mathbf{W},\sigma^2)]$ w.r.t. $\mathbf{W}$ and setting to zero

$$\mathbf{W} = \left[\sum_{n=1}^N\boldsymbol{x}_n\mathbb{E}[\boldsymbol{z}_n]^\top\right]\left[\sum_{n=1}^N\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]\right]^{-1} = \left[\sum_{n=1}^N\boldsymbol{x}_n\mathbb{E}[\boldsymbol{z}_n]^\top\right]\left[\sum_{n=1}^N\mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \sigma^2\mathbf{M}^{-1}\right]^{-1}$$

The Full EM Algorithm

- Initialize $\mathbf{W}$ and σ^2
- **E step:** Compute the exp. complete data log-lik. using current $\mathbf{W}$ and σ^2

$$\sum_{n=1}^N\left\{\frac{D}{2}\log\sigma^2 + \frac{1}{2\sigma^2}||\boldsymbol{x}_n||^2 - \frac{1}{\sigma^2}\mathbb{E}[\boldsymbol{z}_n]^\top\mathbf{W}^\top\boldsymbol{x}_n + \frac{1}{2\sigma^2}\mathrm{tr}(\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]\mathbf{W}^\top\mathbf{W}) + \frac{1}{2}\mathrm{tr}(\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top])\right\}$$

where

$$\begin{aligned}\mathbb{E}[\boldsymbol{z}_n] &= (\mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}_K)^{-1}\mathbf{W}^\top\boldsymbol{x}_n = \mathbf{M}^{-1}\mathbf{W}^\top\boldsymbol{x}_n \\ \mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top] &= \mathrm{cov}(\boldsymbol{z}_n) + \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top = \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \sigma^2\mathbf{M}^{-1}\end{aligned}$$

- **M step:** Re-estimate $\mathbf{W}$ and σ^2 (taking derivatives w.r.t. $\mathbf{W}$ and σ^2, respectively)

$$\begin{aligned}\mathbf{W}_{new} &= \left[\sum_{n=1}^N\boldsymbol{x}_n\mathbb{E}[\boldsymbol{z}_n]^\top\right]\left[\sum_{n=1}^N\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]\right]^{-1} = \left[\sum_{n=1}^N\boldsymbol{x}_n\mathbb{E}[\boldsymbol{z}_n]^\top\right]\left[\sum_{n=1}^N\mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \sigma^2\mathbf{M}^{-1}\right]^{-1} \\ \sigma^2_{new} &= \frac{1}{ND}\sum_{n=1}^N\left\{||\boldsymbol{x}_n||^2 - 2\mathbb{E}[\boldsymbol{z}_n]^\top\mathbf{W}_{new}^\top\boldsymbol{x}_n + \mathrm{tr}\left(\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]\mathbf{W}_{new}^\top\mathbf{W}_{new}\right)\right\}\end{aligned}$$

- Set $\mathbf{W} = \mathbf{W}_{new}$ and $\sigma^2 = \sigma^2_{new}$
- If not converged, go back to E step.

EM for PPCA to a PCA-like Algorithm

- Let's see what happens if the noise variance σ^2 goes to 0
- Let's first look at the E step

$$\mathbb{E}[z_n] = (\mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}_K)^{-1}\mathbf{W}^\top x_n = (\mathbf{W}^\top\mathbf{W})^{-1}\mathbf{W}^\top x_n$$

No need to compute $\mathbb{E}[z_n z_n^\top]$ since it will simply be equal to $\mathbb{E}[z_n]\mathbb{E}[z_n]^\top$

- Thus, in this case, E step computes an orthogonal projection of x_n
- Let's now look at the M step

$$\mathbf{W}_{new} = \left[\sum_{n=1}^{N} x_n \mathbb{E}[z_n]^\top\right]\left[\sum_{n=1}^{N}\mathbb{E}[z_n]\mathbb{E}[z_n]^\top\right]^{-1} = \mathbf{X}^\top\Omega(\Omega^\top\Omega)^{-1}$$

where $\mathbf{\Omega} = \mathbb{E}[\mathbf{Z}]$ is an $N \times K$ matrix with row n equal to $\mathbb{E}[z_n]$

- Thus, in this case, M step finds $\mathbf{W}$ that minimizes the reconstruction error

$$\mathbf{W}_{new} = \arg\min_{\mathbf{W}} ||\mathbf{X} - \mathbb{E}[\mathbf{Z}]\mathbf{W}||^2 = \arg\min_{\mathbf{W}} ||\mathbf{X} - \mathbf{\Omega}\mathbf{W}||^2$$

Benefits of PPCA over PCA

- Can handle missing data (can treat it as latent variable in E step)
- Doesn't require computing the $D \times D$ cov. matrix of data and doing expensive eigen-decomposition. When K is small (i.e., we only want few eigen vectors), this is especially nice because only inverting $K \times K$ is required
- Easy to "plug-in" PPCA as part of more complex problems, e.g., mixtures of PPCA models for doing nonlinear dimensionality reduction, or subspace clustering (i.e., clustering when data in each cluster lives on a lower dimensional subspace).
- Possible to give it a fully Bayesian treatment (which has many benefits such as inferring K)

Identifiability

- Note that $p(x_n) = \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)$
- If we replace $\mathbf{W}$ by $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ for some orthogonal rotation matrix $\mathbf{R}$ then

$$\begin{aligned} p(x_n) &= \mathcal{N}(\mathbf{0}, \tilde{\mathbf{W}}\tilde{\mathbf{W}}^\top + \sigma^2\mathbf{I}_D) \\ &= \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{R}\mathbf{R}^\top\mathbf{W}^\top + \sigma^2\mathbf{I}_D) \\ &= \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D) \end{aligned}$$

- Thus PPCA doesn't give a unique solution (for every $\mathbf{W}$, there is another $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ that gives the same solution)
- Thus the PPCA model is not uniquely identifiable
- Usually this is not a problem, unless we want to interpret $\mathbf{W}$
- To ensure identifiability, we can impose certain structure on $\mathbf{W}$, e.g., constrain it to be a lower-triangular or sparse matrix

Factor Analysis

- Similar to PPCA except that the Gaussian conditional distribution $p(x_n|z_n)$ has diagonal instead of spherical covariance

$$x_n \sim \mathcal{N}(\mathbf{W}z_n, \mathbf{\Psi})$$

where $\mathbf{\Psi}$ is a diagonal matrix

- In Factor Analysis, the projection matrix $\mathbf{W}$ is also called the Factor Loading Matrix and z_n is called the factor scores for example n
- EM for Factor Analysis is same as that for PPCA except
  - The required expectations in the E step :

$$\begin{aligned} \mathbb{E}[z_n] &= \mathbf{G}^{-1}\mathbf{W}^\top\mathbf{\Psi}^{-1}x_n \\ \mathbb{E}[z_n z_n^\top] &= \mathbb{E}[z_n]\mathbb{E}[z_n]^\top + \mathbf{G} \end{aligned}$$

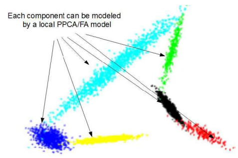
  where $\mathbf{G} = (\mathbf{W}^\top\mathbf{\Psi}^{-1}\mathbf{W} + \mathbf{I}_K)^{-1}$

  - In the M step, in addition to $\mathbf{W}_{new}$, we also need to estimate $\mathbf{\Psi}$

$$\mathbf{\Psi}_{new} = \text{diag}\left\{\mathbf{S} - \mathbf{W}_{new}\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[z_n]x_n^\top\right\}$$

Mixture of PPCAs/Mixture of Factor Analyzers

- PPCA and FA learn a linear projection of the data (i.e., are linear dimensionality reduction methods)
- Can use mixture of PPCAs or mixture of FAs to learn nonlinear projections (i.e., nonlinear dimensionality reduction)



- Similar to mixture of Gaussians, except that now each Gaussian is replaced by a PPCA or FA model
- Note: Unline PPCA/FA, can't have $\boldsymbol{\mu} = 0$: Each PPCA/FA based mixture component k will have a nonzero mean $\boldsymbol{\mu}_k$ and projection matrix $\mathbf{W}_k$
- We will later look at another nonlinear dim. red. model - Gaussian Process Latent Variable Models (GPLVM)

For details, check out "Mixtures of Probabilistic Principal Component Analysers" by Tipping and Bishop, and "The EM Algorithm for Mixtures of Factor Analyzers" by Ghahramani and Hinton