Generative Models for Dimensionality Reduction: Probabilistic PCA and Factor Analysis

Piyush Rai

Machine Learning (CS771A)

Oct 5, 2016

Generative Model for Dimensionality Reduction

- Assume the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$

Generative Model for Dimensionality Reduction

- Assume the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ $(K \ll D)$ as

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

Generative Model for Dimensionality Reduction

- Assume the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ $(K \ll D)$ as

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

  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as

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

Generative Model for Dimensionality Reduction

- Assume the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ ($K \ll D$) as
$$\boldsymbol{z}_n \sim \mathcal{N}(0, \mathbf{I}_K)$$
  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as
$$\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \sigma^2\mathbf{I}_D)$$

    where $\mathbf{W}$ is the $D \times K$ "factor loading matrix" or "dictionary"

Generative Model for Dimensionality Reduction

- Assume the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ $(K \ll D)$ as
  $$\boldsymbol{z}_n \sim \mathcal{N}(0, \mathbf{I}_K)$$
  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as
  $$\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \sigma^2\mathbf{I}_D)$$

  where $\mathbf{W}$ is the $D \times K$ "factor loading matrix" or "dictionary"
- $\boldsymbol{z}_n$ is K-dim latent features or latent factors or "coding" of $\boldsymbol{x}_n$ w.r.t. $\mathbf{W}$

Generative Model for Dimensionality Reduction

- Assume the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ $(K \ll D)$ as

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

  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as

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

  where $\mathbf{W}$ is the $D \times K$ "factor loading matrix" or "dictionary"

- $\boldsymbol{z}_n$ is K-dim latent features or latent factors or "coding" of $\boldsymbol{x}_n$ w.r.t. $\mathbf{W}$
- Note: Can also write $\boldsymbol{x}_n$ as a linear transformation of $\boldsymbol{z}_n$, plus Gaussian noise

$$\boldsymbol{x}_n = \mathbf{W}\boldsymbol{z}_n + \epsilon_n \qquad (\text{where } \epsilon_n \sim \mathcal{N}(0, \sigma^2\mathbf{I}_D))$$

Generative Model for Dimensionality Reduction

- Assume the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ ($K \ll D$) as

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

  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as

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

    where $\mathbf{W}$ is the $D \times K$ "factor loading matrix" or "dictionary"

- $\boldsymbol{z}_n$ is K-dim latent features or latent factors or "coding" of $\boldsymbol{x}_n$ w.r.t. $\mathbf{W}$
- Note: Can also write $\boldsymbol{x}_n$ as a linear transformation of $\boldsymbol{z}_n$, plus Gaussian noise

$$\boldsymbol{x}_n = \mathbf{W}\boldsymbol{z}_n + \epsilon_n \qquad (\text{where } \epsilon_n \sim \mathcal{N}(0, \sigma^2\mathbf{I}_D))$$

- This is "Probabilistic PCA" (PPCA) with Gaussian observation model

Generative Model for Dimensionality Reduction

- Assume the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ $(K \ll D)$ as

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

  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as

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

    where $\mathbf{W}$ is the $D \times K$ "factor loading matrix" or "dictionary"

- $\boldsymbol{z}_n$ is K-dim latent features or latent factors or "coding" of $\boldsymbol{x}_n$ w.r.t. $\mathbf{W}$
- Note: Can also write $\boldsymbol{x}_n$ as a linear transformation of $\boldsymbol{z}_n$, plus Gaussian noise

$$\boldsymbol{x}_n = \mathbf{W}\boldsymbol{z}_n + \epsilon_n \qquad (\text{where } \epsilon_n \sim \mathcal{N}(0, \sigma^2\mathbf{I}_D))$$

- This is "Probabilistic PCA" (PPCA) with Gaussian observation model
- Want to learn model parameters $\mathbf{W}, \sigma^2$ and latent factors $\{\boldsymbol{z}_n\}_{n=1}^N$

Generative Model for Dimensionality Reduction

- Assume the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ $(K \ll D)$ as

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

  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as

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

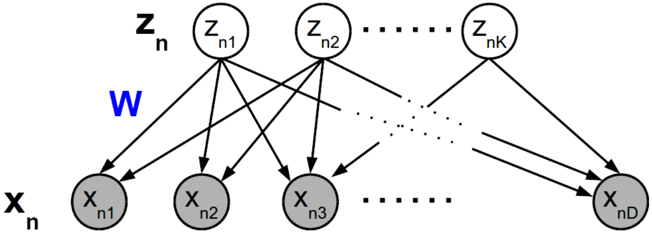
  where $\mathbf{W}$ is the $D \times K$ "factor loading matrix" or "dictionary"
- $\boldsymbol{z}_n$ is K-dim latent features or latent factors or "coding" of $\boldsymbol{x}_n$ w.r.t. $\mathbf{W}$
- Note: Can also write $\boldsymbol{x}_n$ as a linear transformation of $\boldsymbol{z}_n$, plus Gaussian noise

$$\boldsymbol{x}_n = \mathbf{W}\boldsymbol{z}_n + \epsilon_n \qquad (\text{where } \epsilon_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_D))$$

- This is "Probabilistic PCA" (PPCA) with Gaussian observation model
- Want to learn model parameters $\mathbf{W}, \sigma^2$ and latent factors $\{\boldsymbol{z}_n\}_{n=1}^N$
- When $\epsilon_n \sim \mathcal{N}(0, \Psi)$, Ψ is diagonal, it is called "Factor Analysis" (FA)

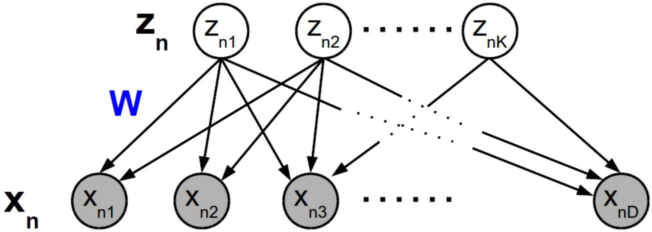
Generative Model for Dimensionality Reduction

- Zooming in at the relationship between each $\boldsymbol{x}_n \in \mathbb{R}^D$ and each $\boldsymbol{z}_n \in \mathbb{R}^K$



Generative Model for Dimensionality Reduction

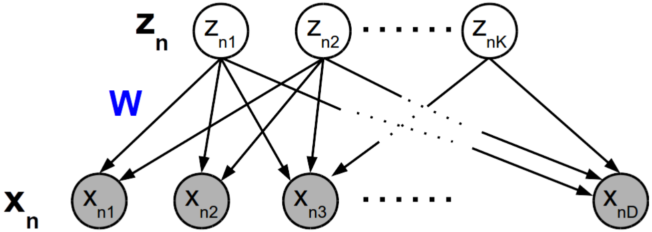
- Zooming in at the relationship between each $\boldsymbol{x}_n \in \mathbb{R}^D$ and each $\boldsymbol{z}_n \in \mathbb{R}^K$



- W_{dk} denotes the weight of relationship between feature d and latent factor k

Generative Model for Dimensionality Reduction

- Zooming in at the relationship between each $\boldsymbol{x}_n \in \mathbb{R}^D$ and each $\boldsymbol{z}_n \in \mathbb{R}^K$



- W_{dk} denotes the weight of relationship between feature d and latent factor k
- This view also helps in thinking about "deep" generative models that have many layers of latent variables or "hidden units"

Linear Gaussian Systems

- Note that PPCA and FA are special cases of linear Gaussian Systems which have the following general form

$$
\begin{aligned}
p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) \\
p(\boldsymbol{x}|\boldsymbol{z}) &= \mathcal{N}(\mathbf{W}\boldsymbol{z} + \mathbf{b}, \boldsymbol{\Sigma}_x)
\end{aligned}
$$

Linear Gaussian Systems

- Note that PPCA and FA are special cases of linear Gaussian Systems which have the following general form

$$
\begin{aligned}
p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) \\
p(\boldsymbol{x}|\boldsymbol{z}) &= \mathcal{N}(\mathbf{W}\boldsymbol{z} + \mathbf{b}, \boldsymbol{\Sigma}_x) \qquad (\boldsymbol{x} = \mathbf{W}\boldsymbol{z} + \mathbf{b} + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \boldsymbol{\Sigma}_x))
\end{aligned}
$$

Linear Gaussian Systems

- Note that PPCA and FA are special cases of linear Gaussian Systems which have the following general form

$$
\begin{aligned}
p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) \\
p(\boldsymbol{x}|\boldsymbol{z}) &= \mathcal{N}(\mathbf{W}\boldsymbol{z} + \mathbf{b}, \boldsymbol{\Sigma}_x) \qquad (\boldsymbol{x} = \mathbf{W}\boldsymbol{z} + \mathbf{b} + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \boldsymbol{\Sigma}_x))
\end{aligned}
$$

- A few nice properties of such systems (follow from properties of Gaussians):

Linear Gaussian Systems

- Note that PPCA and FA are special cases of linear Gaussian Systems which have the following general form

$$\begin{aligned} p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) \\ p(\boldsymbol{x}|\boldsymbol{z}) &= \mathcal{N}(\mathbf{W}\boldsymbol{z} + \mathbf{b}, \boldsymbol{\Sigma}_x) \qquad (\boldsymbol{x} = \mathbf{W}\boldsymbol{z} + \mathbf{b} + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \boldsymbol{\Sigma}_x)) \end{aligned}$$

- A few nice properties of such systems (follow from properties of Gaussians):
  - The marginal distribution of $\boldsymbol{x}$, i.e., $p(\boldsymbol{x})$, is Gaussian

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} = \int p(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z} = \mathcal{N}(\mathbf{W}\boldsymbol{\mu}_z + \mathbf{b}, \boldsymbol{\Sigma}_x + \mathbf{W}\boldsymbol{\Sigma}_z\mathbf{W}^\top)$$

Linear Gaussian Systems

- Note that PPCA and FA are special cases of linear Gaussian Systems which have the following general form

$$\begin{aligned} p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) \\ p(\boldsymbol{x}|\boldsymbol{z}) &= \mathcal{N}(\mathbf{W}\boldsymbol{z} + \mathbf{b}, \boldsymbol{\Sigma}_x) \qquad (\boldsymbol{x} = \mathbf{W}\boldsymbol{z} + \mathbf{b} + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \boldsymbol{\Sigma}_x)) \end{aligned}$$

- A few nice properties of such systems (follow from properties of Gaussians):
  - The marginal distribution of $\boldsymbol{x}$, i.e., $p(\boldsymbol{x})$, is Gaussian

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} = \int p(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z} = \mathcal{N}(\mathbf{W}\boldsymbol{\mu}_z + \mathbf{b}, \boldsymbol{\Sigma}_x + \mathbf{W}\boldsymbol{\Sigma}_z\mathbf{W}^\top)$$

  - The posterior distribution of $\boldsymbol{z}$, i.e., $p(\boldsymbol{z}|\boldsymbol{x}) \propto p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})$ is Gaussian

$$\begin{aligned} p(\boldsymbol{z}|\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ \boldsymbol{\Sigma}^{-1} &= \boldsymbol{\Sigma}_z^{-1} + \mathbf{W}^\top \boldsymbol{\Sigma}_x^{-1} \mathbf{W} \\ \boldsymbol{\mu} &= \boldsymbol{\Sigma}[\mathbf{W}^\top \boldsymbol{\Sigma}_x^{-1}(\boldsymbol{x} - \mathbf{b}) + \boldsymbol{\Sigma}_z^{-1}\boldsymbol{\mu}_z] \end{aligned}$$

Linear Gaussian Systems

- Note that PPCA and FA are special cases of linear Gaussian Systems which have the following general form

$$\begin{aligned} p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) \\ p(\boldsymbol{x}|\boldsymbol{z}) &= \mathcal{N}(\mathbf{W}\boldsymbol{z} + \mathbf{b}, \boldsymbol{\Sigma}_x) \qquad (\boldsymbol{x} = \mathbf{W}\boldsymbol{z} + \mathbf{b} + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \boldsymbol{\Sigma}_x)) \end{aligned}$$

- A few nice properties of such systems (follow from properties of Gaussians):
  - The marginal distribution of $\boldsymbol{x}$, i.e., $p(\boldsymbol{x})$, is Gaussian

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} = \int p(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z} = \mathcal{N}(\mathbf{W}\boldsymbol{\mu}_z + \mathbf{b}, \boldsymbol{\Sigma}_x + \mathbf{W}\boldsymbol{\Sigma}_z\mathbf{W}^\top)$$

  - The posterior distribution of $\boldsymbol{z}$, i.e., $p(\boldsymbol{z}|\boldsymbol{x}) \propto p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})$ is Gaussian

$$\begin{aligned} p(\boldsymbol{z}|\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ \boldsymbol{\Sigma}^{-1} &= \boldsymbol{\Sigma}_z^{-1} + \mathbf{W}^\top \boldsymbol{\Sigma}_x^{-1} \mathbf{W} \\ \boldsymbol{\mu} &= \boldsymbol{\Sigma}[\mathbf{W}^\top \boldsymbol{\Sigma}_x^{-1}(\boldsymbol{x} - \mathbf{b}) + \boldsymbol{\Sigma}_z^{-1} \boldsymbol{\mu}_z] \end{aligned}$$

(Chapter 4 of Murphy and Chapter 2 of Bishop have various useful results on properties of multivar. Gaussians)

PPCA or FA = Low-Rank Gaussian

- Suppose we're modeling D-dim data using a (say zero mean) Gaussian

$$p(\boldsymbol{x}) = \mathcal{N}(0, \boldsymbol{\Sigma})$$

where $\boldsymbol{\Sigma}$ is a $D \times D$ p.s.d. cov. matrix, $\mathcal{O}(D^2)$ parameters needed

PPCA or FA = Low-Rank Gaussian

- Suppose we're modeling D-dim data using a (say zero mean) Gaussian

$$p(\boldsymbol{x}) = \mathcal{N}(0, \boldsymbol{\Sigma})$$

  where $\boldsymbol{\Sigma}$ is a $D \times D$ p.s.d. cov. matrix, $\mathcal{O}(D^2)$ parameters needed

- Consider modeling the same data using the one-layer PPCA model

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\mathbf{W}\boldsymbol{z}, \sigma^2\mathbf{I}_D) \qquad \text{where } p(\boldsymbol{z}) = \mathcal{N}(0, \mathbf{I}_K)$$

PPCA or FA = Low-Rank Gaussian

- Suppose we're modeling D-dim data using a (say zero mean) Gaussian

$$p(\boldsymbol{x}) = \mathcal{N}(0, \boldsymbol{\Sigma})$$

where $\boldsymbol{\Sigma}$ is a $D \times D$ p.s.d. cov. matrix, $\mathcal{O}(D^2)$ parameters needed

- Consider modeling the same data using the one-layer PPCA model

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\mathbf{W}\boldsymbol{z}, \sigma^2\mathbf{I}_D) \qquad \text{where } p(\boldsymbol{z}) = \mathcal{N}(0, \mathbf{I}_K)$$

- For this Gaussian PPCA, the marginal distribution $p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}$ is

$$\boxed{p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)} \qquad \text{(using result from previous slide)}$$

PPCA or FA = Low-Rank Gaussian

- Suppose we're modeling D-dim data using a (say zero mean) Gaussian
$$p(\boldsymbol{x}) = \mathcal{N}(0, \boldsymbol{\Sigma})$$
where $\boldsymbol{\Sigma}$ is a $D \times D$ p.s.d. cov. matrix, $\mathcal{O}(D^2)$ parameters needed
- Consider modeling the same data using the one-layer PPCA model
$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\mathbf{W}\boldsymbol{z}, \sigma^2\mathbf{I}_D) \qquad \text{where } p(\boldsymbol{z}) = \mathcal{N}(0, \mathbf{I}_K)$$
- For this Gaussian PPCA, the marginal distribution $p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z})d\boldsymbol{z}$ is
$$\boxed{p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)} \qquad \text{(using result from previous slide)}$$
- Cov. matrix is close to low-rank. Also, only $(DK + 1)$ free params to learn

PPCA or FA = Low-Rank Gaussian

- Suppose we're modeling D-dim data using a (say zero mean) Gaussian

$$p(\boldsymbol{x}) = \mathcal{N}(0, \boldsymbol{\Sigma})$$

where $\boldsymbol{\Sigma}$ is a $D \times D$ p.s.d. cov. matrix, $\mathcal{O}(D^2)$ parameters needed

- Consider modeling the same data using the one-layer PPCA model

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\mathbf{W}\boldsymbol{z}, \sigma^2\mathbf{I}_D) \qquad \text{where } p(\boldsymbol{z}) = \mathcal{N}(0, \mathbf{I}_K)$$

- For this Gaussian PPCA, the marginal distribution $p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}$ is

$$\boxed{p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)} \qquad \text{(using result from previous slide)}$$

- Cov. matrix is close to low-rank. Also, only $(DK + 1)$ free params to learn
- Thus modeling data using a Gaussian PPCA instead of Gaussian with full cov. may be easier when we have very little but high-dim data (i.e., $D \gg N$)

PPCA or FA = Low-Rank Gaussian

- Suppose we're modeling D-dim data using a (say zero mean) Gaussian
$$p(\boldsymbol{x}) = \mathcal{N}(0, \boldsymbol{\Sigma})$$
where $\boldsymbol{\Sigma}$ is a $D \times D$ p.s.d. cov. matrix, $\mathcal{O}(D^2)$ parameters needed
- Consider modeling the same data using the one-layer PPCA model
$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\mathbf{W}\boldsymbol{z}, \sigma^2\mathbf{I}_D) \qquad \text{where } p(\boldsymbol{z}) = \mathcal{N}(0, \mathbf{I}_K)$$
- For this Gaussian PPCA, the marginal distribution $p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z})d\boldsymbol{z}$ is
$$\boxed{p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)} \qquad \text{(using result from previous slide)}$$
- Cov. matrix is close to low-rank. Also, only $(DK + 1)$ free params to learn
- Thus modeling data using a Gaussian PPCA instead of Gaussian with full cov. may be easier when we have very little but high-dim data (i.e., $D \gg N$)
- $p(\boldsymbol{x})$ is still a Gaussian but between two extremes (diagonal cov and full cov)

Parameter Estimation for PPCA

- Data: $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, latent vars: $\mathbf{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$, parameters: $\mathbf{W}, \sigma^2$

† Probabilistic Principal Component Analysis (Tipping and Bishop, 1999)

Parameter Estimation for PPCA

- Data: $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, latent vars: $\mathbf{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$, parameters: $\mathbf{W}, \sigma^2$
- Note: If we just want to estimate $\mathbf{W}$ and σ^2, we could do MLE directly[†] on incomplete data likelihood $p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I}_D)$

[†] Probabilistic Principal Component Analysis (Tipping and Bishop, 1999)

Parameter Estimation for PPCA

- Data: $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, latent vars: $\mathbf{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$, parameters: $\mathbf{W}, \sigma^2$
- Note: If we just want to estimate $\mathbf{W}$ and σ^2, we could do MLE directly† on incomplete data likelihood $p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)$
- Closed-form solution† can be obtained for $\mathbf{W}$ and σ^2 by maximizing

$$\log p(\mathbf{X}) = -\frac{N}{2}(D\log 2\pi + \log|\mathbf{C}| + \text{trace}(\mathbf{C}^{-1}\mathbf{S}))$$

where $\mathbf{S}$ is the data cov. matrix and $\mathbf{C}^{-1} = \sigma^{-1}\mathbf{I} - \sigma^{-1}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^\top$and $\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}$

† Probabilistic Principal Component Analysis (Tipping and Bishop, 1999)

Parameter Estimation for PPCA

- Data: $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, latent vars: $\mathbf{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$, parameters: $\mathbf{W}, \sigma^2$
- Note: If we just want to estimate $\mathbf{W}$ and σ^2, we could do MLE directly† on incomplete data likelihood $p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)$
- Closed-form solution† can be obtained for $\mathbf{W}$ and σ^2 by maximizing

$$\log p(\mathbf{X}) = -\frac{N}{2}(D \log 2\pi + \log |\mathbf{C}| + \text{trace}(\mathbf{C}^{-1}\mathbf{S}))$$

where $\mathbf{S}$ is the data cov. matrix and $\mathbf{C}^{-1} = \sigma^{-1}\mathbf{I} - \sigma^{-1}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^\top$ and $\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}$

- But this method isn't usually preferred because

† Probabilistic Principal Component Analysis (Tipping and Bishop, 1999)

Parameter Estimation for PPCA

- Data: $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, latent vars: $\mathbf{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$, parameters: $\mathbf{W}, \sigma^2$
- Note: If we just want to estimate $\mathbf{W}$ and σ^2, we could do MLE directly† on incomplete data likelihood $p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I}_D)$
- Closed-form solution† can be obtained for $\mathbf{W}$ and σ^2 by maximizing

$$\log p(\mathbf{X}) = -\frac{N}{2}(D \log 2\pi + \log |\mathbf{C}| + \text{trace}(\mathbf{C}^{-1}\mathbf{S}))$$

where $\mathbf{S}$ is the data cov. matrix and $\mathbf{C}^{-1} = \sigma^{-1}\mathbf{I} - \sigma^{-1}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^\top$ and $\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}$

- But this method isn't usually preferred because
  - It is expensive (have to work with cov. matrices and their eig-decomp)

† Probabilistic Principal Component Analysis (Tipping and Bishop, 1999)

Parameter Estimation for PPCA

- Data: $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, latent vars: $\mathbf{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$, parameters: $\mathbf{W}, \sigma^2$
- Note: If we just want to estimate $\mathbf{W}$ and σ^2, we could do MLE directly† on incomplete data likelihood $p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)$
- Closed-form solution† can be obtained for $\mathbf{W}$ and σ^2 by maximizing

$$\log p(\mathbf{X}) = -\frac{N}{2}(D \log 2\pi + \log|\mathbf{C}| + \text{trace}(\mathbf{C}^{-1}\mathbf{S}))$$

  where $\mathbf{S}$ is the data cov. matrix and $\mathbf{C}^{-1} = \sigma^{-1}\mathbf{I} - \sigma^{-1}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^\top$and $\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}$
- But this method isn't usually preferred because
  - It is expensive (have to work with cov. matrices and their eig-decomp)
  - A closed-form solution may not even be possible for more general models (e.g. Factor Analysis where $\sigma^2\mathbf{I}$ is replace by diagonal matrix, or mixture of PPCA)

† Probabilistic Principal Component Analysis (Tipping and Bishop, 1999)

Parameter Estimation for PPCA

- Data: $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, latent vars: $\mathbf{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$, parameters: $\mathbf{W}, \sigma^2$
- Note: If we just want to estimate $\mathbf{W}$ and σ^2, we could do MLE directly[†] on incomplete data likelihood $p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I}_D)$
- Closed-form solution[†] can be obtained for $\mathbf{W}$ and σ^2 by maximizing

$$\log p(\mathbf{X}) = -\frac{N}{2}(D \log 2\pi + \log |\mathbf{C}| + \text{trace}(\mathbf{C}^{-1}\mathbf{S}))$$

where $\mathbf{S}$ is the data cov. matrix and $\mathbf{C}^{-1} = \sigma^{-1}\mathbf{I} - \sigma^{-1}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^\top$and $\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}$

- But this method isn't usually preferred because
  - It is expensive (have to work with cov. matrices and their eig-decomp)
  - A closed-form solution may not even be possible for more general models (e.g. Factor Analysis where $\sigma^2\mathbf{I}$ is replace by diagonal matrix, or mixture of PPCA)
  - Won't be possible to learn the latent variables $\mathbf{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$

[†] Probabilistic Principal Component Analysis (Tipping and Bishop, 1999)

EM based Parameter Estimation for PPCA

- We will instead go the EM route and work with the complete data log-lik.

$$\log p(\mathbf{X}, \mathbf{Z}|\mathbf{W}, \sigma^2)$$

EM based Parameter Estimation for PPCA

- We will instead go the EM route and work with the complete data log-lik.

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

EM based Parameter Estimation for PPCA

- We will instead go the EM route and work with the complete data log-lik.

$$\log p(\mathbf{X}, \mathbf{Z}|\mathbf{W}, \sigma^2) = \log \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\mathbf{W}, \sigma^2) \quad = \quad \log \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2)p(\boldsymbol{z}_n)$$

EM based Parameter Estimation for PPCA

- We will instead go the EM route and work with the complete data log-lik.

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

$$= \sum_{n=1}^{N} \{\log p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) + \log p(\boldsymbol{z}_n)\}$$

EM based Parameter Estimation for PPCA

- We will instead go the EM route and work with the complete data log-lik.

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

- As we'll see, it leads to much simpler expressions and efficient solutions

EM based Parameter Estimation for PPCA

- We will instead go the EM route and work with the complete data log-lik.

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

$$= \sum_{n=1}^{N} \{\log p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) + \log p(\boldsymbol{z}_n)\}$$

- As we'll see, it leads to much simpler expressions and efficient solutions
- Recall that $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp(-\frac{(\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)^\top (\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)}{2\sigma^2})$ and $p(\boldsymbol{z}_n) \propto \exp(-\frac{\boldsymbol{z}_n^\top \boldsymbol{z}_n}{2})$

EM based Parameter Estimation for PPCA

- We will instead go the EM route and work with the complete data log-lik.

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

$$= \sum_{n=1}^{N} \{\log p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) + \log p(\boldsymbol{z}_n)\}$$

- As we'll see, it leads to much simpler expressions and efficient solutions
- Recall that $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp(-\frac{(\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)^\top (\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)}{2\sigma^2})$ and $p(\boldsymbol{z}_n) \propto \exp(-\frac{\boldsymbol{z}_n^\top \boldsymbol{z}_n}{2})$
- Plugging in, simplifying, using the trace trick, and ignoring constants, we get the following expression for complete data log-likelihood $\log p(\mathbf{X}, \mathbf{Z}|\mathbf{W}, \sigma^2)$

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

EM based Parameter Estimation for PPCA

- We will instead go the EM route and work with the complete data log-lik.

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

$$= \sum_{n=1}^{N}\{\log p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) + \log p(\boldsymbol{z}_n)\}$$

- As we'll see, it leads to much simpler expressions and efficient solutions
- Recall that $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp(-\frac{(\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)^\top(\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)}{2\sigma^2})$ and $p(\boldsymbol{z}_n) \propto \exp(-\frac{\boldsymbol{z}_n^\top \boldsymbol{z}_n}{2})$
- Plugging in, simplifying, using the trace trick, and ignoring constants, we get the following expression for complete data log-likelihood $\log p(\mathbf{X}, \mathbf{Z}|\mathbf{W}, \sigma^2)$

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

- We will need the expected value of this quantity in M step of EM

EM based Parameter Estimation for PPCA

- We will instead go the EM route and work with the complete data log-lik.

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

$$= \sum_{n=1}^{N} \{\log p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) + \log p(\boldsymbol{z}_n)\}$$

- As we'll see, it leads to much simpler expressions and efficient solutions
- Recall that $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp(-\frac{(\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)^\top (\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)}{2\sigma^2})$ and $p(\boldsymbol{z}_n) \propto \exp(-\frac{\boldsymbol{z}_n^\top \boldsymbol{z}_n}{2})$
- Plugging in, simplifying, using the trace trick, and ignoring constants, we get the following expression for complete data log-likelihood $\log p(\mathbf{X}, \mathbf{Z}|\mathbf{W}, \sigma^2)$

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

- We will need the expected value of this quantity in M step of EM
  - This requires computing the posterior distribution of $\boldsymbol{z}_n$ in E step (which is Gaussian; recall the result from earlier slide on linear Gaussian systems)

EM based Parameter Estimation for PPCA

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

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

EM based Parameter Estimation for PPCA

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

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

- 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

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

EM based Parameter Estimation for PPCA

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

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

- 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

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

- To compute $\mathbf{W}$, we also need two expectations $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$

EM based Parameter Estimation for PPCA

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

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

- 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

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

- To compute $\mathbf{W}$, we also need two expectations $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$
- These can be obtained in E step by computing posterior over $\boldsymbol{z}_n$, which, using the results of Gaussian posterior for linear Gaussian models, is

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

EM based Parameter Estimation for PPCA

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

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

- 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

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

- To compute $\mathbf{W}$, we also need two expectations $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n \boldsymbol{z}_n^\top]$
- These can be obtained in E step by computing posterior over $\boldsymbol{z}_n$, which, using the results of Gaussian posterior for linear Gaussian models, is

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

- The required expectations can be easily obtained from the Gaussian posterior

EM based Parameter Estimation for PPCA

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

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

- 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

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

- To compute $\mathbf{W}$, we also need two expectations $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n \boldsymbol{z}_n^\top]$
- These can be obtained in E step by computing posterior over $\boldsymbol{z}_n$, which, using the results of Gaussian posterior for linear Gaussian models, is

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

- The required expectations can be easily obtained from the Gaussian posterior

$$\mathbb{E}[\boldsymbol{z}_n] = \mathbf{M}^{-1}\mathbf{W}^\top \boldsymbol{x}_n$$

EM based Parameter Estimation for PPCA

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

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

- 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

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

- To compute $\mathbf{W}$, we also need two expectations $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$
- These can be obtained in E step by computing posterior over $\boldsymbol{z}_n$, which, using the results of Gaussian posterior for linear Gaussian models, is

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

- The required expectations can be easily obtained from the Gaussian posterior

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

EM based Parameter Estimation for PPCA

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

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

- 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

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

- To compute $\mathbf{W}$, we also need two expectations $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$
- These can be obtained in E step by computing posterior over $\boldsymbol{z}_n$, which, using the results of Gaussian posterior for linear Gaussian models, is

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

- The required expectations can be easily obtained from the Gaussian posterior

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

EM based Parameter Estimation for PPCA

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

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

- 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

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

- To compute $\mathbf{W}$, we also need two expectations $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n \boldsymbol{z}_n^\top]$
- These can be obtained in E step by computing posterior over $\boldsymbol{z}_n$, which, using the results of Gaussian posterior for linear Gaussian models, is

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

- The required expectations can be easily obtained from the Gaussian posterior

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

- Note: The noise variance σ^2 can also be estimated (take deriv., set to zero..)

The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data

The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data
- **E step:** Compute the expectations required in M step. For each data point

The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data
- **E step:** Compute the expectations required in M step. For each data point

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

The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data
- **E step:** Compute the expectations required in M step. For each data point

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

The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data
- **E step:** Compute the expectations required in M step. For each data point

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

- **M step:** Re-estimate $\mathbf{W}$ and σ^2

The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data
- **E step:** Compute the expectations required in M step. For each data point

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

- **M step:** Re-estimate $\mathbf{W}$ and σ^2

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

The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data
- **E step:** Compute the expectations required in M step. For each data point

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

- **M step:** Re-estimate $\mathbf{W}$ and σ^2

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

The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data
- **E step:** Compute the expectations required in M step. For each data point

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

- **M step:** Re-estimate $\mathbf{W}$ and σ^2

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

- Set $\mathbf{W} = \mathbf{W}_{new}$ and $\sigma^2 = \sigma^2_{new}$

The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data
- **E step:** Compute the expectations required in M step. For each data point

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

- **M step:** Re-estimate $\mathbf{W}$ and σ^2

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

- Set $\mathbf{W} = \mathbf{W}_{new}$ and $\sigma^2 = \sigma^2_{new}$
- If not converged, go back to E step (can monitor the incomplete/complete log-likelihood to assess convergence)

EM for Factor Analysis

- Similar to PPCA except that the Gaussian conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n)$ has diagonal instead of spherical covariance, i.e., $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a diagonal matrix

EM for Factor Analysis

- Similar to PPCA except that the Gaussian conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n)$ has diagonal instead of spherical covariance, i.e., $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a diagonal matrix
- EM for Factor Analysis is very similar to that for PPCA

EM for Factor Analysis

- Similar to PPCA except that the Gaussian conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n)$ has diagonal instead of spherical covariance, i.e., $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a diagonal matrix
- EM for Factor Analysis is very similar to that for PPCA
  - The required expectations in the E step :

$$
\begin{aligned}
\mathbb{E}[\boldsymbol{z}_n] &= \mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n \\
\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top] &= \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \mathbf{G}
\end{aligned}
$$

  where $\mathbf{G} = (\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\mathbf{W} + \mathbf{I}_K)^{-1}$. Note that if $\boldsymbol{\Psi} = \sigma^2\mathbf{I}_D$, we get the same equations as in PPCA

EM for Factor Analysis

- Similar to PPCA except that the Gaussian conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n)$ has diagonal instead of spherical covariance, i.e., $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a diagonal matrix
- EM for Factor Analysis is very similar to that for PPCA
  - The required expectations in the E step :

$$
\begin{aligned}
\mathbb{E}[\boldsymbol{z}_n] &= \mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n \\
\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top] &= \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \mathbf{G}
\end{aligned}
$$

    where $\mathbf{G} = (\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\mathbf{W} + \mathbf{I}_K)^{-1}$. Note that if $\boldsymbol{\Psi} = \sigma^2\mathbf{I}_D$, we get the same equations as in PPCA
  - In the M step, updates for $\mathbf{W}_{new}$ are the same as PPCA

EM for Factor Analysis

- Similar to PPCA except that the Gaussian conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{z}_n)$ has diagonal instead of spherical covariance, i.e., $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a diagonal matrix
- EM for Factor Analysis is very similar to that for PPCA
  - The required expectations in the E step :

$$\begin{aligned}\mathbb{E}[\boldsymbol{z}_n] &= \mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n \\ \mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top] &= \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \mathbf{G}\end{aligned}$$

    where $\mathbf{G} = (\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\mathbf{W} + \mathbf{I}_K)^{-1}$. Note that if $\boldsymbol{\Psi} = \sigma^2\mathbf{I}_D$, we get the same equations as in PPCA
  - In the M step, updates for $\mathbf{W}_{new}$ are the same as PPCA
  - In the M step, updates for $\boldsymbol{\Psi}$ are

$$\boldsymbol{\Psi}_{new} = \text{diag}\left\{\mathbf{S} - \mathbf{W}_{new}\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[\boldsymbol{z}_n]\boldsymbol{x}_n^\top\right\} \qquad (\mathbf{S} \text{ is the cov. matrix of data})$$

Some Aspects about PPCA/FA

- Can also handle missing data as additional latent variables in E step. Just write each data point as $\boldsymbol{x}_n = [\boldsymbol{x}_n^{obs} \;\; \boldsymbol{x}_n^{miss}]$ and treat $\boldsymbol{x}_n^{miss}$ as latent vars.

Some Aspects about PPCA/FA

- Can also handle missing data as additional latent variables in E step. Just write each data point as $\boldsymbol{x}_n = [\boldsymbol{x}_n^{obs} \ \boldsymbol{x}_n^{miss}]$ and treat $\boldsymbol{x}_n^{miss}$ as latent vars.
  - Usually the posterior $p(\boldsymbol{x}_n^{miss}|\boldsymbol{x}_n^{obs})$ over missing data can be computed

Some Aspects about PPCA/FA

- Can also handle missing data as additional latent variables in E step. Just write each data point as $\boldsymbol{x}_n = [\boldsymbol{x}_n^{obs} \ \boldsymbol{x}_n^{miss}]$ and treat $\boldsymbol{x}_n^{miss}$ as latent vars.
  - Usually the posterior $p(\boldsymbol{x}_n^{miss}|\boldsymbol{x}_n^{obs})$ over missing data can be computed
  - Note: Ability to handle missing data is the property of EM in general and can be used in other models as well (e.g., GMM)

Some Aspects about PPCA/FA

- Can also handle missing data as additional latent variables in E step. Just write each data point as $\boldsymbol{x}_n = [\boldsymbol{x}_n^{obs} \;\; \boldsymbol{x}_n^{miss}]$ and treat $\boldsymbol{x}_n^{miss}$ as latent vars.
  - Usually the posterior $p(\boldsymbol{x}_n^{miss}|\boldsymbol{x}_n^{obs})$ over missing data can be computed
  - Note: Ability to handle missing data is the property of EM in general and can be used in other models as well (e.g., GMM)
- Can learn other model params such as noise variance σ^2 using MLE/MAP

Some Aspects about PPCA/FA

- Can also handle missing data as additional latent variables in E step. Just write each data point as $\boldsymbol{x}_n = [\boldsymbol{x}_n^{obs} \ \boldsymbol{x}_n^{miss}]$ and treat $\boldsymbol{x}_n^{miss}$ as latent vars.
  - Usually the posterior $p(\boldsymbol{x}_n^{miss}|\boldsymbol{x}_n^{obs})$ over missing data can be computed
  - Note: Ability to handle missing data is the property of EM in general and can be used in other models as well (e.g., GMM)
- Can learn other model params such as noise variance σ^2 using MLE/MAP
- Also more efficient than the naïve PCA. Doesn't require computing the $D \times D$ cov. matrix of data and doing expensive eigen-decomposition

Some Aspects about PPCA/FA

- Can also handle missing data as additional latent variables in E step. Just write each data point as $\boldsymbol{x}_n = [\boldsymbol{x}_n^{obs} \ \boldsymbol{x}_n^{miss}]$ and treat $\boldsymbol{x}_n^{miss}$ as latent vars.
  - Usually the posterior $p(\boldsymbol{x}_n^{miss}|\boldsymbol{x}_n^{obs})$ over missing data can be computed
  - Note: Ability to handle missing data is the property of EM in general and can be used in other models as well (e.g., GMM)
- Can learn other model params such as noise variance σ^2 using MLE/MAP
- Also more efficient than the naïve PCA. Doesn't require computing the $D \times D$ cov. matrix of data and doing expensive eigen-decomposition
- Can learn the model very efficiently using "online EM"

Some Aspects about PPCA/FA

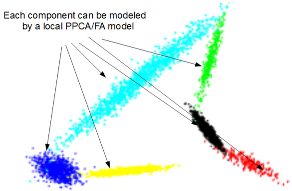
- Can also handle missing data as additional latent variables in E step. Just write each data point as $\boldsymbol{x}_n = [\boldsymbol{x}_n^{obs} \; \boldsymbol{x}_n^{miss}]$ and treat $\boldsymbol{x}_n^{miss}$ as latent vars.
  - Usually the posterior $p(\boldsymbol{x}_n^{miss}|\boldsymbol{x}_n^{obs})$ over missing data can be computed
  - Note: Ability to handle missing data is the property of EM in general and can be used in other models as well (e.g., GMM)
- Can learn other model params such as noise variance σ^2 using MLE/MAP
- Also more efficient than the naïve PCA. Doesn't require computing the $D \times D$ cov. matrix of data and doing expensive eigen-decomposition
- Can learn the model very efficiently using "online EM"
- Possible to give it a fully Bayesian treatment (which has many other benefits such as inferring K using nonparametric Bayesian modeling)

Some Aspects about PPCA/FA

- Provides a framework that could be extended to build more complex models

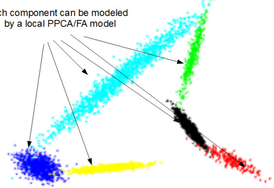
Some Aspects about PPCA/FA

- Provides a framework that could be extended to build more complex models
- Mixture of PPCA/FA models (joint clust. + dim. red., or nonlin. dim. red.)

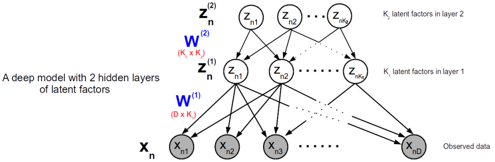


Some Aspects about PPCA/FA

- Provides a framework that could be extended to build more complex models
- Mixture of PPCA/FA models (joint clust. + dim. red., or nonlin. dim. red.)

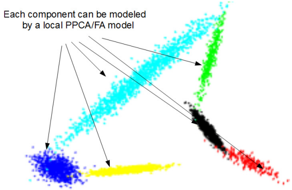


- Deep models for feature learning and dimensionality reduction

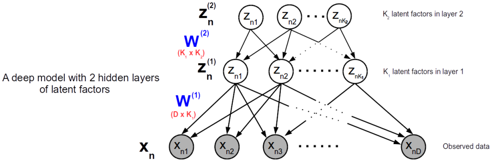


Some Aspects about PPCA/FA

- Provides a framework that could be extended to build more complex models
- Mixture of PPCA/FA models (joint clust. + dim. red., or nonlin. dim. red.)



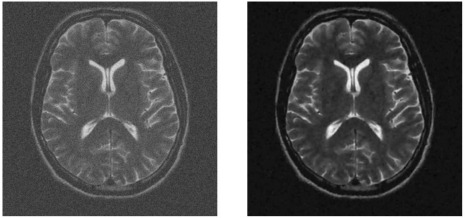
- Deep models for feature learning and dimensionality reduction



- Supervised extensions, e.g., by jointly modeling labels y_n as conditioned on latent factors, i.e., $p(y_n = 1|\boldsymbol{z}_n, \theta)$ using a logistic model with weights $\theta \in \mathbb{R}^K$

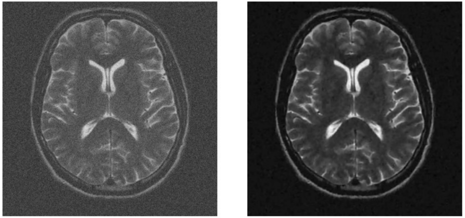
Some Applications of PPCA

- Learning the noise variance allows "image denoising"

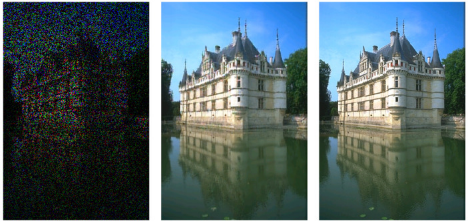


Some Applications of PPCA

- Learning the noise variance allows "image denoising"



- Ability to fill-in missing data allows "image inpainting" (left: image with 80% missing data, middle: reconstructed, right: original)



Using EM for (efficiently) solving standard PCA

- Let's see what happens if the noise variance σ^2 goes to 0

Using EM for (efficiently) solving standard PCA

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

$$\mathbb{E}[\boldsymbol{z}_n] = (\mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}_K)^{-1}\mathbf{W}^\top\boldsymbol{x}_n = (\mathbf{W}^\top\mathbf{W})^{-1}\mathbf{W}^\top\boldsymbol{x}_n$$

Using EM for (efficiently) solving standard PCA

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

$$\mathbb{E}[\boldsymbol{z}_n] = (\mathbf{W}^\top \mathbf{W} + \sigma^2 \mathbf{I}_K)^{-1} \mathbf{W}^\top \boldsymbol{x}_n = (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \boldsymbol{x}_n$$

(no need to compute $\mathbb{E}[\boldsymbol{z}_n \boldsymbol{z}_n^\top]$ since it will simply be equal to $\mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top$)

Using EM for (efficiently) solving standard PCA

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

$$\mathbb{E}[\boldsymbol{z}_n] = (\mathbf{W}^\top \mathbf{W} + \sigma^2 \mathbf{I}_K)^{-1} \mathbf{W}^\top \boldsymbol{x}_n = (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \boldsymbol{x}_n$$

(no need to compute $\mathbb{E}[\boldsymbol{z}_n \boldsymbol{z}_n^\top]$ since it will simply be equal to $\mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top$)

- Let's now look at the M step

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

Using EM for (efficiently) solving standard PCA

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

$$\mathbb{E}[\boldsymbol{z}_n] = (\mathbf{W}^\top \mathbf{W} + \sigma^2 \mathbf{I}_K)^{-1} \mathbf{W}^\top \boldsymbol{x}_n = (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \boldsymbol{x}_n$$

(no need to compute $\mathbb{E}[\boldsymbol{z}_n \boldsymbol{z}_n^\top]$ since it will simply be equal to $\mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top$)

- Let's now look at the M step

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

Using EM for (efficiently) solving standard PCA

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

$$\mathbb{E}[\boldsymbol{z}_n] = (\mathbf{W}^\top \mathbf{W} + \sigma^2 \mathbf{I}_K)^{-1} \mathbf{W}^\top \boldsymbol{x}_n = (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \boldsymbol{x}_n$$

(no need to compute $\mathbb{E}[\boldsymbol{z}_n \boldsymbol{z}_n^\top]$ since it will simply be equal to $\mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top$)

- Let's now look at the M step

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

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

Using EM for (efficiently) solving standard PCA

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

$$\mathbb{E}[\boldsymbol{z}_n] = (\mathbf{W}^\top \mathbf{W} + \sigma^2 \mathbf{I}_K)^{-1} \mathbf{W}^\top \boldsymbol{x}_n = (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \boldsymbol{x}_n$$

(no need to compute $\mathbb{E}[\boldsymbol{z}_n \boldsymbol{z}_n^\top]$ since it will simply be equal to $\mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top$)

- Let's now look at the M step

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

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

- Note that M step is equivalent to finding $\mathbf{W}$ that minimizes the recon. error

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

Using EM for (efficiently) solving standard PCA

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

$$\mathbb{E}[\boldsymbol{z}_n] = (\mathbf{W}^\top \mathbf{W} + \sigma^2 \mathbf{I}_K)^{-1} \mathbf{W}^\top \boldsymbol{x}_n = (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \boldsymbol{x}_n$$

(no need to compute $\mathbb{E}[\boldsymbol{z}_n \boldsymbol{z}_n^\top]$ since it will simply be equal to $\mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top$)

- Let's now look at the M step

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

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

- Note that M step is equivalent to finding $\mathbf{W}$ that minimizes the recon. error

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

- Thus EM can also be used to efficiently solve the standard non-probabilistic PCA without doing eigendecomposition

Identifiability

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

Identifiability

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

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

Identifiability

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

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

- 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)

Identifiability

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

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

- 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

Identifiability

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

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

- 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 very strictly interpret $\mathbf{W}$

Identifiability

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

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

- 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 very strictly interpret $\mathbf{W}$
- To ensure identifiability, we can impose some more structure on $\mathbf{W}$, e.g., constrain it to be a lower-triangular or sparse matrix

Some Concluding Thoughts

- Discussed the basic idea of generative models for doing unsupervised learning

Some Concluding Thoughts

- Discussed the basic idea of generative models for doing unsupervised learning
- Looked at a way (EM) to perform parameter estimation in such models
  - EM is a general framework for parameter estimation in latent variable models

Some Concluding Thoughts

- Discussed the basic idea of generative models for doing unsupervised learning
- Looked at a way (EM) to perform parameter estimation in such models
    - EM is a general framework for parameter estimation in latent variable models
- Looked at two types of unsupervised learning problems

Some Concluding Thoughts

- Discussed the basic idea of generative models for doing unsupervised learning
- Looked at a way (EM) to perform parameter estimation in such models
    - EM is a general framework for parameter estimation in latent variable models
- Looked at two types of unsupervised learning problems
    - Mixture models: Clustering

Some Concluding Thoughts

- Discussed the basic idea of generative models for doing unsupervised learning
- Looked at a way (EM) to perform parameter estimation in such models
  - EM is a general framework for parameter estimation in latent variable models
- Looked at two types of unsupervised learning problems
  - Mixture models: Clustering
  - Latent factor models: Dimensionality reduction

Some Concluding Thoughts

- Discussed the basic idea of generative models for doing unsupervised learning
- Looked at a way (EM) to perform parameter estimation in such models
  - EM is a general framework for parameter estimation in latent variable models
- Looked at two types of unsupervised learning problems
  - Mixture models: Clustering
  - Latent factor models: Dimensionality reduction
  - Both these models can also be used for estimating the prob. density $p(\boldsymbol{x})$

Some Concluding Thoughts

- Discussed the basic idea of generative models for doing unsupervised learning
- Looked at a way (EM) to perform parameter estimation in such models
  - EM is a general framework for parameter estimation in latent variable models
- Looked at two types of unsupervised learning problems
  - Mixture models: Clustering
  - Latent factor models: Dimensionality reduction
  - Both these models can also be used for estimating the prob. density $p(\boldsymbol{x})$
- More sophisticated models are usually built on these basic principles

Some Concluding Thoughts

- Discussed the basic idea of generative models for doing unsupervised learning
- Looked at a way (EM) to perform parameter estimation in such models
  - EM is a general framework for parameter estimation in latent variable models
- Looked at two types of unsupervised learning problems
  - Mixture models: Clustering
  - Latent factor models: Dimensionality reduction
  - Both these models can also be used for estimating the prob. density $p(\boldsymbol{x})$
- More sophisticated models are usually built on these basic principles
  - E.g., Hidden Markov Models and Kalman Filters can be seen as generalization of mixture models and Gaussian latent factor models, respectively, for sequential data ($\boldsymbol{z}_n$ correspond to the "state" of $\boldsymbol{x}_n$)

Some Concluding Thoughts

- Discussed the basic idea of generative models for doing unsupervised learning
- Looked at a way (EM) to perform parameter estimation in such models
  - EM is a general framework for parameter estimation in latent variable models
- Looked at two types of unsupervised learning problems
  - Mixture models: Clustering
  - Latent factor models: Dimensionality reduction
  - Both these models can also be used for estimating the prob. density $p(\boldsymbol{x})$
- More sophisticated models are usually built on these basic principles
  - E.g., Hidden Markov Models and Kalman Filters can be seen as generalization of mixture models and Gaussian latent factor models, respectively, for sequential data ($\boldsymbol{z}_n$ correspond to the "state" of $\boldsymbol{x}_n$)
  - We will look at these and other related models (e.g., LSTM) when talking about learning from seqential data