Latent Variable Models for Dimensionality Reduction

Piyush Rai

Probabilistic Machine Learning (CS772A)

September 7, 2017

A Simple Additive Model for Data Compression

- Consider a set of observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, with $\boldsymbol{x}_n \in \mathbb{R}^D$

A Simple Additive Model for Data Compression

- Consider a set of observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, with $\boldsymbol{x}_n \in \mathbb{R}^D$
- Let's approximate each $\boldsymbol{x}_n$ by a linear combination of K vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K$ $(K \ll D)$

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

A Simple Additive Model for Data Compression

- Consider a set of observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, with $\boldsymbol{x}_n \in \mathbb{R}^D$
- Let's approximate each $\boldsymbol{x}_n$ by a linear combination of K vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K$ $(K \ll D)$

$$\boldsymbol{x}_n \approx \sum_{k=1}^{K} z_{nk} \boldsymbol{w}_k \qquad \text{or} \qquad \boldsymbol{x}_n \approx \mathbf{W} \boldsymbol{z}_n$$

A Simple Additive Model for Data Compression

- Consider a set of observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, with $\boldsymbol{x}_n \in \mathbb{R}^D$
- Let's approximate each $\boldsymbol{x}_n$ by a linear combination of K vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K$ $(K \ll D)$

$$\boldsymbol{x}_n \approx \sum_{k=1}^{K} z_{nk} \boldsymbol{w}_k \qquad \text{or} \qquad \boldsymbol{x}_n \approx \mathbf{W} \boldsymbol{z}_n$$

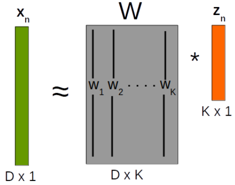
where $\mathbf{W} = [\boldsymbol{w}_1 \ldots \boldsymbol{w}_K]$ is $D \times K$, each $\boldsymbol{w}_k \in \mathbb{R}^D$, and $\boldsymbol{z}_n = [z_{n1} \ldots z_{nK}] \in \mathbb{R}^K$

A Simple Additive Model for Data Compression

- Consider a set of observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, with $\boldsymbol{x}_n \in \mathbb{R}^D$
- Let's approximate each $\boldsymbol{x}_n$ by a linear combination of K vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K$ $(K \ll D)$

$$\boldsymbol{x}_n \approx \sum_{k=1}^{K} z_{nk} \boldsymbol{w}_k \qquad \text{or} \qquad \boldsymbol{x}_n \approx \mathbf{W} \boldsymbol{z}_n$$

where $\mathbf{W} = [\boldsymbol{w}_1 \ldots \boldsymbol{w}_K]$ is $D \times K$, each $\boldsymbol{w}_k \in \mathbb{R}^D$, and $\boldsymbol{z}_n = [z_{n1} \ldots z_{nK}] \in \mathbb{R}^K$

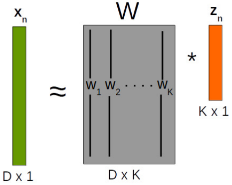


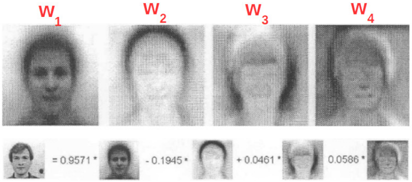
A Simple Additive Model for Data Compression

- Consider a set of observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, with $\boldsymbol{x}_n \in \mathbb{R}^D$
- Let's approximate each $\boldsymbol{x}_n$ by a linear combination of K vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K$ $(K \ll D)$

$$\boldsymbol{x}_n \approx \sum_{k=1}^{K} z_{nk} \boldsymbol{w}_k \qquad \text{or} \qquad \boldsymbol{x}_n \approx \mathbf{W} \boldsymbol{z}_n$$

where $\mathbf{W} = [\boldsymbol{w}_1 \ldots \boldsymbol{w}_K]$ is $D \times K$, each $\boldsymbol{w}_k \in \mathbb{R}^D$, and $\boldsymbol{z}_n = [z_{n1} \ldots z_{nK}] \in \mathbb{R}^K$



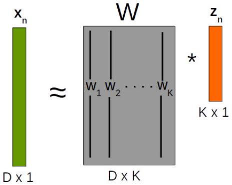


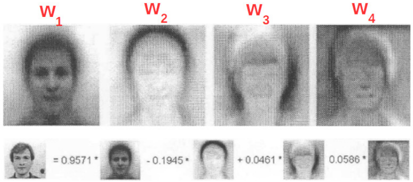
A Simple Additive Model for Data Compression

- Consider a set of observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, with $\boldsymbol{x}_n \in \mathbb{R}^D$
- Let's approximate each $\boldsymbol{x}_n$ by a linear combination of K vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K$ $(K \ll D)$

$$\boldsymbol{x}_n \approx \sum_{k=1}^{K} z_{nk} \boldsymbol{w}_k \qquad \text{or} \qquad \boldsymbol{x}_n \approx \mathbf{W} \boldsymbol{z}_n$$

where $\mathbf{W} = [\boldsymbol{w}_1 \ldots \boldsymbol{w}_K]$ is $D \times K$, each $\boldsymbol{w}_k \in \mathbb{R}^D$, and $\boldsymbol{z}_n = [z_{n1} \ldots z_{nK}] \in \mathbb{R}^K$





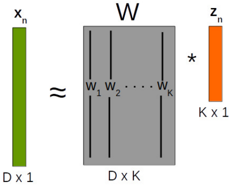
- z_{nk} tell us much of "component" $\boldsymbol{w}_k$ is present in the observation $\boldsymbol{x}_n$

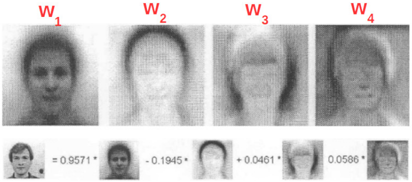
A Simple Additive Model for Data Compression

- Consider a set of observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, with $\boldsymbol{x}_n \in \mathbb{R}^D$
- Let's approximate each $\boldsymbol{x}_n$ by a linear combination of K vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K$ $(K \ll D)$

$$\boldsymbol{x}_n \approx \sum_{k=1}^{K} z_{nk} \boldsymbol{w}_k \qquad \text{or} \qquad \boldsymbol{x}_n \approx \mathbf{W} \boldsymbol{z}_n$$

where $\mathbf{W} = [\boldsymbol{w}_1 \ldots \boldsymbol{w}_K]$ is $D \times K$, each $\boldsymbol{w}_k \in \mathbb{R}^D$, and $\boldsymbol{z}_n = [z_{n1} \ldots z_{nK}] \in \mathbb{R}^K$

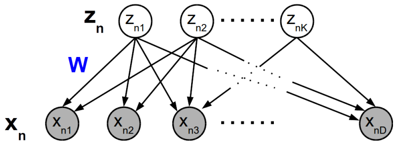




- z_{nk} tell us much of "component" $\boldsymbol{w}_k$ is present in the observation $\boldsymbol{x}_n$
- Can think of $\boldsymbol{z}_n \in \mathbb{R}^K$ as a "compressed" latent representation of $\boldsymbol{x}_n \in \mathbb{R}^D$ (would like to learn it)

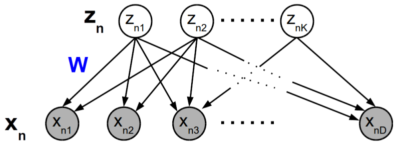
Zooming-in into the Model..

- Can think of latent $\boldsymbol{z}_n$ generating $\boldsymbol{x}_n$ via a linear mapping defined by $D \times K$ matrix $\mathbf{W}$



Zooming-in into the Model..

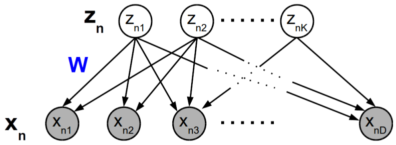
- Can think of latent $\boldsymbol{z}_n$ generating $\boldsymbol{x}_n$ via a linear mapping defined by $D \times K$ matrix $\mathbf{W}$



- Note: Also possible to define a nonlinear mapping from $\boldsymbol{z}_n$ to $\boldsymbol{x}_n$, e.g. Variational Auto-encoders (VAE), Gaussian Process Latent Variable Models (GPLVM)

Zooming-in into the Model..

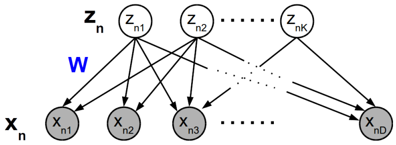
- Can think of latent $\boldsymbol{z}_n$ generating $\boldsymbol{x}_n$ via a linear mapping defined by $D \times K$ matrix $\mathbf{W}$



- Note: Also possible to define a nonlinear mapping from $\boldsymbol{z}_n$ to $\boldsymbol{x}_n$, e.g. Variational Auto-encoders (VAE), Gaussian Process Latent Variable Models (GPLVM)
  - Such models are much more flexible but inference becomes harder though

Zooming-in into the Model..

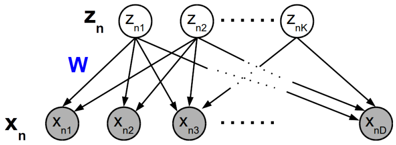
- Can think of latent $\boldsymbol{z}_n$ generating $\boldsymbol{x}_n$ via a linear mapping defined by $D \times K$ matrix $\mathbf{W}$



- Note: Also possible to define a nonlinear mapping from $\boldsymbol{z}_n$ to $\boldsymbol{x}_n$, e.g. Variational Auto-encoders (VAE), Gaussian Process Latent Variable Models (GPLVM)
  - Such models are much more flexible but inference becomes harder though
  - Will look at some of these when discussing deep generative models

Zooming-in into the Model..

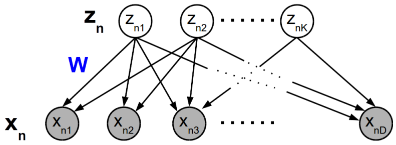
- Can think of latent $\boldsymbol{z}_n$ generating $\boldsymbol{x}_n$ via a linear mapping defined by $D \times K$ matrix $\mathbf{W}$



- Note: Also possible to define a nonlinear mapping from $\boldsymbol{z}_n$ to $\boldsymbol{x}_n$, e.g. Variational Auto-encoders (VAE), Gaussian Process Latent Variable Models (GPLVM)
  - Such models are much more flexible but inference becomes harder though
  - Will look at some of these when discussing deep generative models
- **Today's focus** will be on the linear case (will also look at a nonlinear extension via mixtures)

Zooming-in into the Model..

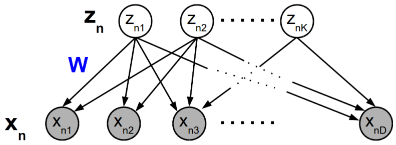
- Can think of latent $\boldsymbol{z}_n$ generating $\boldsymbol{x}_n$ via a linear mapping defined by $D \times K$ matrix $\mathbf{W}$



- Note: Also possible to define a nonlinear mapping from $\boldsymbol{z}_n$ to $\boldsymbol{x}_n$, e.g. Variational Auto-encoders (VAE), Gaussian Process Latent Variable Models (GPLVM)
  - Such models are much more flexible but inference becomes harder though
  - Will look at some of these when discussing deep generative models
- **Today's focus** will be on the linear case (will also look at a nonlinear extension via mixtures)
  - Linear models are simple but very powerful. Very easy to do inference in such models (e.g., using EM)

Zooming-in into the Model..

- Can think of latent $\boldsymbol{z}_n$ generating $\boldsymbol{x}_n$ via a linear mapping defined by $D \times K$ matrix $\mathbf{W}$



- Note: Also possible to define a nonlinear mapping from $\boldsymbol{z}_n$ to $\boldsymbol{x}_n$, e.g. Variational Auto-encoders (VAE), Gaussian Process Latent Variable Models (GPLVM)
  - Such models are much more flexible but inference becomes harder though
  - Will look at some of these when discussing deep generative models
- **Today's focus** will be on the linear case (will also look at a nonlinear extension via mixtures)
  - Linear models are simple but very powerful. Very easy to do inference in such models (e.g., using EM)
  - Nice interpretability. E.g., columns of $[\boldsymbol{w}_1 \ldots \boldsymbol{w}_K]$ are like K "latent parts" that compose $\boldsymbol{x}_n$

Latent Factor Models for Dimensionality Reduction

- In the linear model, we represented $\boldsymbol{x}_n$ approximately as $\boldsymbol{x}_n \approx \mathbf{W}\boldsymbol{z}_n$

Latent Factor Models for Dimensionality Reduction

- In the linear model, we represented $\boldsymbol{x}_n$ approximately as $\boldsymbol{x}_n \approx \mathbf{W}\boldsymbol{z}_n$
- Let's assume the data-point specific Gaussian noise or "residual" $\epsilon_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_D)$ and write

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

Latent Factor Models for Dimensionality Reduction

- In the linear model, we represented $\boldsymbol{x}_n$ approximately as $\boldsymbol{x}_n \approx \mathbf{W}\boldsymbol{z}_n$
- Let's assume the data-point specific Gaussian noise or "residual" $\epsilon_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_D)$ and write

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

- The distribution of $\boldsymbol{x}_n$, conditioned on $\boldsymbol{z}_n$ will be

Latent Factor Models for Dimensionality Reduction

- In the linear model, we represented $\boldsymbol{x}_n$ approximately as $\boldsymbol{x}_n \approx \mathbf{W}\boldsymbol{z}_n$
- Let's assume the data-point specific Gaussian noise or "residual" $\epsilon_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_D)$ and write

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

- The distribution of $\boldsymbol{x}_n$, conditioned on $\boldsymbol{z}_n$ will be

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

Latent Factor Models for Dimensionality Reduction

- In the linear model, we represented $\boldsymbol{x}_n$ approximately as $\boldsymbol{x}_n \approx \mathbf{W}\boldsymbol{z}_n$
- Let's assume the data-point specific Gaussian noise or "residual" $\epsilon_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_D)$ and write

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

- The distribution of $\boldsymbol{x}_n$, conditioned on $\boldsymbol{z}_n$ will be

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

- If $\boldsymbol{z}_n$ is *a priori* Gaussian distributed, e.g., $p(\boldsymbol{z}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$

Latent Factor Models for Dimensionality Reduction

- In the linear model, we represented $\boldsymbol{x}_n$ approximately as $\boldsymbol{x}_n \approx \mathbf{W}\boldsymbol{z}_n$
- Let's assume the data-point specific Gaussian noise or "residual" $\epsilon_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_D)$ and write

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

- The distribution of $\boldsymbol{x}_n$, conditioned on $\boldsymbol{z}_n$ will be

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

- If $\boldsymbol{z}_n$ is *a priori* Gaussian distributed, e.g., $p(\boldsymbol{z}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I}_K) \Rightarrow$ Linear Gaussian Model (LGM)

Latent Factor Models for Dimensionality Reduction

- In the linear model, we represented $\boldsymbol{x}_n$ approximately as $\boldsymbol{x}_n \approx \mathbf{W}\boldsymbol{z}_n$
- Let's assume the data-point specific Gaussian noise or "residual" $\epsilon_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_D)$ and write

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

- The distribution of $\boldsymbol{x}_n$, conditioned on $\boldsymbol{z}_n$ will be

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

- If $\boldsymbol{z}_n$ is *a priori* Gaussian distributed, e.g., $p(\boldsymbol{z}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I}_K) \Rightarrow$ Linear Gaussian Model (LGM)
- Note: Recall LGMs have nice properties, e.g., easy to compute $p(\boldsymbol{z}|\boldsymbol{x})$ or $p(\boldsymbol{x})$ (both Gaussians!)

Latent Factor Models for Dimensionality Reduction

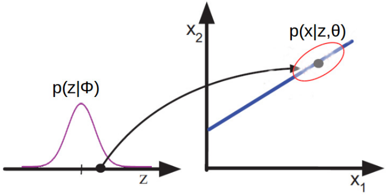
- In the linear model, we represented $\boldsymbol{x}_n$ approximately as $\boldsymbol{x}_n \approx \mathbf{W}\boldsymbol{z}_n$
- Let's assume the data-point specific Gaussian noise or "residual" $\epsilon_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_D)$ and write

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

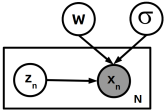
- The distribution of $\boldsymbol{x}_n$, conditioned on $\boldsymbol{z}_n$ will be

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

- If $\boldsymbol{z}_n$ is *a priori* Gaussian distributed, e.g., $p(\boldsymbol{z}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I}_K) \Rightarrow$ Linear Gaussian Model (LGM)
- Note: Recall LGMs have nice properties, e.g., easy to compute $p(\boldsymbol{z}|\boldsymbol{x})$ or $p(\boldsymbol{x})$ (both Gaussians!)

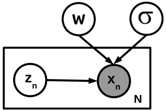


Latent Factor Models for Dimensionality Reduction



- Can think of the data generation process via the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$

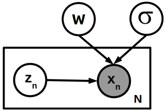
Latent Factor Models for Dimensionality Reduction



- Can think of the data generation process via the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ as

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

Latent Factor Models for Dimensionality Reduction



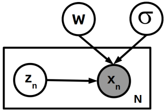
- Can think of the data generation process via the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ as

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

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

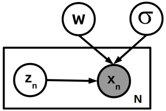
$$\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \sigma^2 \mathbf{I}_D)$$

Latent Factor Models for Dimensionality Reduction



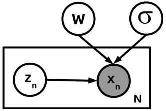
- Can think of the data generation process via the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ as
  $$\boldsymbol{z}_n \sim \mathcal{N}(0, \mathbf{I}_K)$$
  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as
  $$\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \sigma^2 \mathbf{I}_D)$$
- $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$ is the $D \times K$ "factor loading matrix" (also known as "dictionary")

Latent Factor Models for Dimensionality Reduction



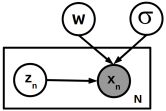
- Can think of the data generation process via the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ as
$$\boldsymbol{z}_n \sim \mathcal{N}(0, \mathbf{I}_K)$$
  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as
$$\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \sigma^2\mathbf{I}_D)$$
- $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$ is the $D \times K$ "factor loading matrix" (also known as "dictionary")
- $\boldsymbol{z}_n$ called latent features or latent factors or factor scores of $\boldsymbol{x}_n$ w.r.t. dictionary $\mathbf{W}$

Latent Factor Models for Dimensionality Reduction



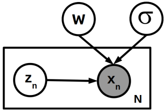
- Can think of the data generation process via the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ as
$$\boldsymbol{z}_n \sim \mathcal{N}(0, \mathbf{I}_K)$$
  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as
$$\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \sigma^2\mathbf{I}_D)$$
- $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$ is the $D \times K$ "factor loading matrix" (also known as "dictionary")
- $\boldsymbol{z}_n$ called latent features or latent factors or factor scores of $\boldsymbol{x}_n$ w.r.t. dictionary $\mathbf{W}$
- This model is popularly known as Probabilistic PCA (PPCA)

Latent Factor Models for Dimensionality Reduction



- Can think of the data generation process via the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ as
  $$\boldsymbol{z}_n \sim \mathcal{N}(0, \mathbf{I}_K)$$
  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as
  $$\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \sigma^2\mathbf{I}_D)$$
- $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$ is the $D \times K$ "factor loading matrix" (also known as "dictionary")
- $\boldsymbol{z}_n$ called latent features or latent factors or factor scores of $\boldsymbol{x}_n$ w.r.t. dictionary $\mathbf{W}$
- This model is popularly known as Probabilistic PCA (PPCA)
- **Note:** The Gaussians on $\boldsymbol{z}$ and $\boldsymbol{x}$ can be replaced by other distributions (e.g., Exp. Family)

Latent Factor Models for Dimensionality Reduction



- Can think of the data generation process via the following generative story for each $\boldsymbol{x}_n \in \mathbb{R}^D$
  - Generate latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ as
    $$\boldsymbol{z}_n \sim \mathcal{N}(0, \mathbf{I}_K)$$
  - Generate data $\boldsymbol{x}_n$ conditioned on $\boldsymbol{z}_n$ as
    $$\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \sigma^2\mathbf{I}_D)$$
- $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$ is the $D \times K$ "factor loading matrix" (also known as "dictionary")
- $\boldsymbol{z}_n$ called latent features or latent factors or factor scores of $\boldsymbol{x}_n$ w.r.t. dictionary $\mathbf{W}$
- This model is popularly known as Probabilistic PCA (PPCA)
- **Note:** The Gaussians on $\boldsymbol{z}$ and $\boldsymbol{x}$ can be replaced by other distributions (e.g., Exp. Family)
- In the Gaussian case, if $p(\epsilon) = \mathcal{N}(\mathbf{0}, \Psi)$ where Ψ is diagonal, it's called Factor Analysis (FA)

PPCA: Marginal and Posterior Distributions

- 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: Marginal and Posterior Distributions

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

PPCA: Marginal and Posterior Distributions

- 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 PPCA: $p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\mathbf{W}\boldsymbol{z}, \sigma^2\mathbf{I}_D)$, $p(\boldsymbol{z}) = \mathcal{N}(0, \mathbf{I}_K)$
- For this Gaussian PPCA, the marginal distribution $p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}$ is

$$\boxed{p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)} \qquad \text{(recall the Gaussian marginal result; verify)}$$

PPCA: Marginal and Posterior Distributions

- 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 PPCA: $p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\mathbf{W}\boldsymbol{z}, \sigma^2\mathbf{I}_D)$, $p(\boldsymbol{z}) = \mathcal{N}(0, \mathbf{I}_K)$
- For this Gaussian PPCA, the marginal distribution $p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z})d\boldsymbol{z}$ is

$$\boxed{p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)} \qquad \text{(recall the Gaussian marginal result; verify)}$$

- Cov. matrix is close to low-rank as $\sigma^2 \to 0$. Only $(DK + 1)$ parameters needed (nice when $D \gg N$)

PPCA: Marginal and Posterior Distributions

- 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 PPCA: $p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\mathbf{W}\boldsymbol{z}, \sigma^2\mathbf{I}_D)$, $p(\boldsymbol{z}) = \mathcal{N}(0, \mathbf{I}_K)$
- For this Gaussian PPCA, the marginal distribution $p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z})d\boldsymbol{z}$ is

$$\boxed{p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)} \qquad \text{(recall the Gaussian marginal result; verify)}$$

- Cov. matrix is close to low-rank as $\sigma^2 \to 0$. Only $(DK + 1)$ parameters needed (nice when $D \gg N$)
  - PPCA = Low-rank Gaussian. Fewer parameters to learn; less chance of overfitting

PPCA: Marginal and Posterior Distributions

- 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 PPCA: $p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\mathbf{W}\boldsymbol{z}, \sigma^2\mathbf{I}_D)$, $p(\boldsymbol{z}) = \mathcal{N}(0, \mathbf{I}_K)$
- For this Gaussian PPCA, the marginal distribution $p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z})d\boldsymbol{z}$ is

$$\boxed{p(\boldsymbol{x}) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)} \qquad \text{(recall the Gaussian marginal result; verify)}$$

- Cov. matrix is close to low-rank as $\sigma^2 \to 0$. Only $(DK + 1)$ parameters needed (nice when $D \gg N$)
  - PPCA = Low-rank Gaussian. Fewer parameters to learn; less chance of overfitting
- Posterior of $\boldsymbol{z}$ is also Gaussian (recall the Gaussian posterior result of linear Gaussian model)

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

Identifiability

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

Identifiability

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

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

Identifiability

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

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

Identifiability

- Note that for PPCA, we have $p(\boldsymbol{x}_n) = \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)$
- If we replace $\mathbf{W}$ by $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ for some $K \times K$ 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 for PPCA, we have $p(\boldsymbol{x}_n) = \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I}_D)$
- If we replace $\mathbf{W}$ by $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ for some $K \times K$ 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 solution isn't unique (for every $\mathbf{W}$, there is a $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ that gives the same solution)

Identifiability

- Note that for PPCA, we have $p(\boldsymbol{x}_n) = \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I}_D)$
- If we replace $\mathbf{W}$ by $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ for some $K \times K$ 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 solution isn't unique (for every $\mathbf{W}$, there is a $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ that gives the same solution)
- Thus the PPCA/FA model is not uniquely identifiable

Identifiability

- Note that for PPCA, we have $p(\boldsymbol{x}_n) = \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)$
- If we replace $\mathbf{W}$ by $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ for some $K \times K$ 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 solution isn't unique (for every $\mathbf{W}$, there is a $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ that gives the same solution)
- Thus the PPCA/FA model is not uniquely identifiable
- Usually this is not a problem, unless we want to very strictly interpret $\mathbf{W}$

Identifiability

- Note that for PPCA, we have $p(\boldsymbol{x}_n) = \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)$
- If we replace $\mathbf{W}$ by $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ for some $K \times K$ 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 solution isn't unique (for every $\mathbf{W}$, there is a $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ that gives the same solution)
- Thus the PPCA/FA model is not uniquely identifiable
- Usually this is not a problem, unless we want to very strictly interpret $\mathbf{W}$
- To ensure identifiability, there are some alternatives

Identifiability

- Note that for PPCA, we have $p(\boldsymbol{x}_n) = \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)$
- If we replace $\mathbf{W}$ by $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ for some $K \times K$ 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 solution isn't unique (for every $\mathbf{W}$, there is a $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ that gives the same solution)
- Thus the PPCA/FA model is not uniquely identifiable
- Usually this is not a problem, unless we want to very strictly interpret $\mathbf{W}$
- To ensure identifiability, there are some alternatives
    - Constrain $\mathbf{W}$ it to be a lower-triangular or sparse matrix

Identifiability

- Note that for PPCA, we have $p(\boldsymbol{x}_n) = \mathcal{N}(\mathbf{0}, \mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I}_D)$
- If we replace $\mathbf{W}$ by $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ for some $K \times K$ 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 solution isn't unique (for every $\mathbf{W}$, there is a $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ that gives the same solution)
- Thus the PPCA/FA model is not uniquely identifiable
- Usually this is not a problem, unless we want to very strictly interpret $\mathbf{W}$
- To ensure identifiability, there are some alternatives
  - Constrain $\mathbf{W}$ it to be a lower-triangular or sparse matrix
  - Use Independent Component Analysis (ICA): ICA uses a non-Gaussian prior on $\boldsymbol{z}$ to get identifiability

$$
p(\boldsymbol{z}) = \prod_{k=1}^{K} p_k(z_k) \quad \text{(each } p_k \text{ is a non-Gaussian distr. like Laplace)}
$$

Parameter Estimation for PPCA via MLE

- Given data $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, estimate latent vars: $\mathbf{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$ and parameters $\Theta = (\mathbf{W}, \sigma^2)$

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

Parameter Estimation for PPCA via MLE

- Given data $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, estimate latent vars: $\mathbf{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$ and parameters $\Theta = (\mathbf{W}, \sigma^2)$
- If we only want $\Theta = (\mathbf{W}$ and $\sigma^2)$, we can do MLE directly on $p(\mathbf{X}|\Theta) = \mathcal{N}(0, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)$

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

Parameter Estimation for PPCA via MLE

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

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

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

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

Parameter Estimation for PPCA via MLE

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

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

  where $\mathbf{S}$ is the data covariance matrix, $\mathbf{C}^{-1} = \sigma^{-1}\mathbf{I} - \sigma^{-1}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^\top$and $\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}$
- The MLE solution is given by

$$\begin{aligned} \mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma^2_{ML}\mathbf{I})^{1/2}\mathbf{R} \\ \sigma^2_{ML} &= \frac{1}{D-K}\sum_{k=K+1}^{D}\lambda_k \end{aligned}$$

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

Parameter Estimation for PPCA via MLE

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

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

  where $\mathbf{S}$ is the data covariance matrix, $\mathbf{C}^{-1} = \sigma^{-1}\mathbf{I} - \sigma^{-1}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^\top$and $\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}$
- The MLE solution is given by

$$\begin{aligned} \mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma_{ML}^2\mathbf{I})^{1/2}\mathbf{R} \\ \sigma_{ML}^2 &= \frac{1}{D-K}\sum_{k=K+1}^{D} \lambda_k \end{aligned}$$

  where $\mathbf{U}_K$ is $D \times K$ matrix of top K eigvecs of $\mathbf{S}$,

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

Parameter Estimation for PPCA via MLE

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

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

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

- The MLE solution is given by

$$\begin{aligned}
\mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma^2_{ML}\mathbf{I})^{1/2}\mathbf{R} \\
\sigma^2_{ML} &= \frac{1}{D-K}\sum_{k=K+1}^{D}\lambda_k
\end{aligned}$$

where $\mathbf{U}_K$ is $D \times K$ matrix of top K eigvecs of $\mathbf{S}$, $\mathbf{L}_K$: $K \times K$ diagonal matrix of top K eigvals $\lambda_1, \ldots, \lambda_K$,

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

Parameter Estimation for PPCA via MLE

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

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

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

- The MLE solution is given by

$$\begin{aligned} \mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma_{ML}^2\mathbf{I})^{1/2}\mathbf{R} \\ \sigma_{ML}^2 &= \frac{1}{D-K}\sum_{k=K+1}^{D} \lambda_k \end{aligned}$$

where $\mathbf{U}_K$ is $D \times K$ matrix of top K eigvecs of $\mathbf{S}$, $\mathbf{L}_K$: $K \times K$ diagonal matrix of top K eigvals $\lambda_1, \ldots, \lambda_K$, $\mathbf{R}$ is a $K \times K$ arbitrary rotation matrix

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

Parameter Estimation for PPCA via MLE

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

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

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

- The MLE solution is given by

$$\begin{aligned} \mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma_{ML}^2\mathbf{I})^{1/2}\mathbf{R} \\ \sigma_{ML}^2 &= \frac{1}{D-K}\sum_{k=K+1}^{D}\lambda_k \end{aligned}$$

where $\mathbf{U}_K$ is $D \times K$ matrix of top K eigvecs of $\mathbf{S}$, $\mathbf{L}_K$: $K \times K$ diagonal matrix of top K eigvals $\lambda_1, \ldots, \lambda_K$, $\mathbf{R}$ is a $K \times K$ arbitrary rotation matrix (equivalent to PCA for $\mathbf{R} = \mathbf{I}$ and $\sigma^2 \to 0$)

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

Parameter Estimation for PPCA via EM

- First of all, why bother doing EM here? Didn't MLE already give us a closed-form solution?

Parameter Estimation for PPCA via EM

- First of all, why bother doing EM here? Didn't MLE already give us a closed-form solution?

$$
\begin{aligned}
\mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma^2_{ML}\mathbf{I})^{1/2}\mathbf{R} \\
\sigma^2_{ML} &= \frac{1}{D-K}\sum_{k=K+1}^{D}\lambda_k
\end{aligned}
$$

Parameter Estimation for PPCA via EM

- First of all, why bother doing EM here? Didn't MLE already give us a closed-form solution?

$$\begin{aligned}\mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma^2_{ML}\mathbf{I})^{1/2}\mathbf{R} \\ \sigma^2_{ML} &= \frac{1}{D-K}\sum_{k=K+1}^{D}\lambda_k\end{aligned}$$

- Well, several reasons that make EM appealing for PPCA

Parameter Estimation for PPCA via EM

- First of all, why bother doing EM here? Didn't MLE already give us a closed-form solution?

$$\begin{aligned} \mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma^2_{ML}\mathbf{I})^{1/2}\mathbf{R} \\ \sigma^2_{ML} &= \frac{1}{D-K}\sum_{k=K+1}^{D}\lambda_k \end{aligned}$$

- Well, several reasons that make EM appealing for PPCA
  - MLE for PPCA becomes much easier (and computationally efficient) when using EM

Parameter Estimation for PPCA via EM

- First of all, why bother doing EM here? Didn't MLE already give us a closed-form solution?

$$\begin{aligned}\mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma_{ML}^2\mathbf{I})^{1/2}\mathbf{R} \\ \sigma_{ML}^2 &= \frac{1}{D-K}\sum_{k=K+1}^{D}\lambda_k\end{aligned}$$

- Well, several reasons that make EM appealing for PPCA
  - MLE for PPCA becomes much easier (and computationally efficient) when using EM
  - EM allows directly inferring the latent vars. $\boldsymbol{z}_n$'s (MLE only gives parameters Θ)

Parameter Estimation for PPCA via EM

- First of all, why bother doing EM here? Didn't MLE already give us a closed-form solution?

$$
\begin{aligned}
\mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma^2_{ML}\mathbf{I})^{1/2}\mathbf{R} \\
\sigma^2_{ML} &= \frac{1}{D-K}\sum_{k=K+1}^{D}\lambda_k
\end{aligned}
$$

- Well, several reasons that make EM appealing for PPCA
  - MLE for PPCA becomes much easier (and computationally efficient) when using EM
  - EM allows directly inferring the latent vars. $\boldsymbol{z}_n$'s (MLE only gives parameters Θ)
  - No need to construct $\mathbf{S}$ or do eigen-decomposition of $\mathbf{S}$ (can be expensive for large D and N)

Parameter Estimation for PPCA via EM

- First of all, why bother doing EM here? Didn't MLE already give us a closed-form solution?

$$\begin{aligned} \mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma^2_{ML}\mathbf{I})^{1/2}\mathbf{R} \\ \sigma^2_{ML} &= \frac{1}{D-K}\sum_{k=K+1}^{D}\lambda_k \end{aligned}$$

- Well, several reasons that make EM appealing for PPCA
  - MLE for PPCA becomes much easier (and computationally efficient) when using EM
  - EM allows directly inferring the latent vars. $\boldsymbol{z}_n$'s (MLE only gives parameters Θ)
  - No need to construct $\mathbf{S}$ or do eigen-decomposition of $\mathbf{S}$ (can be expensive for large D and N)
  - Closed-form MLE not even possible for other PPCA variant (e.g., FA where $p(\epsilon) = \mathcal{N}(\mathbf{0}, \Psi)$)

Parameter Estimation for PPCA via EM

- First of all, why bother doing EM here? Didn't MLE already give us a closed-form solution?

$$\begin{aligned} \mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma^2_{ML}\mathbf{I})^{1/2}\mathbf{R} \\ \sigma^2_{ML} &= \frac{1}{D-K}\sum_{k=K+1}^{D}\lambda_k \end{aligned}$$

- Well, several reasons that make EM appealing for PPCA
  - MLE for PPCA becomes much easier (and computationally efficient) when using EM
  - EM allows directly inferring the latent vars. $\boldsymbol{z}_n$'s (MLE only gives parameters Θ)
  - No need to construct $\mathbf{S}$ or do eigen-decomposition of $\mathbf{S}$ (can be expensive for large D and N)
  - Closed-form MLE not even possible for other PPCA variant (e.g., FA where $p(\epsilon) = \mathcal{N}(\mathbf{0}, \Psi)$)
  - EM can properly handle missing data, e.g., by computing $p(\boldsymbol{x}_n^{missing}|\boldsymbol{x}_n^{obs})$; often doable easily

Parameter Estimation for PPCA via EM

- First of all, why bother doing EM here? Didn't MLE already give us a closed-form solution?

$$
\begin{aligned}
\mathbf{W}_{ML} &= \mathbf{U}_K(\mathbf{L}_K - \sigma^2_{ML}\mathbf{I})^{1/2}\mathbf{R} \\
\sigma^2_{ML} &= \frac{1}{D-K}\sum_{k=K+1}^{D}\lambda_k
\end{aligned}
$$

- Well, several reasons that make EM appealing for PPCA
  - MLE for PPCA becomes much easier (and computationally efficient) when using EM
  - EM allows directly inferring the latent vars. $\boldsymbol{z}_n$'s (MLE only gives parameters Θ)
  - No need to construct $\mathbf{S}$ or do eigen-decomposition of $\mathbf{S}$ (can be expensive for large D and N)
  - Closed-form MLE not even possible for other PPCA variant (e.g., FA where $p(\epsilon) = \mathcal{N}(\mathbf{0}, \Psi)$)
  - EM can properly handle missing data, e.g., by computing $p(\boldsymbol{x}_n^{missing}|\boldsymbol{x}_n^{obs})$; often doable easily
  - EM can easily be made online (enables handling large D and large N)

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

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

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

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

  - M Step: Maximize the expected complete data log-lik. (CLL) $\mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)]$ w.r.t. Θ

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

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

  - M Step: Maximize the expected complete data log-lik. (CLL) $\mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)]$ w.r.t. Θ
- The CLL (and expected CLL) for PPCA has a simple expression.

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

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

  - M Step: Maximize the expected complete data log-lik. (CLL) $\mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)]$ w.r.t. Θ
- The CLL (and expected CLL) for PPCA has a simple expression. The CLL is

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

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

  - M Step: Maximize the expected complete data log-lik. (CLL) $\mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)]$ w.r.t. Θ
- The CLL (and expected CLL) for PPCA has a simple expression. The CLL is

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

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

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

  - M Step: Maximize the expected complete data log-lik. (CLL) $\mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)]$ w.r.t. Θ
- The CLL (and expected CLL) for PPCA has a simple expression. The CLL is

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

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

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

  - M Step: Maximize the expected complete data log-lik. (CLL) $\mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)]$ w.r.t. Θ
- The CLL (and expected CLL) for PPCA has a simple expression. The CLL is

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

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

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

  - M Step: Maximize the expected complete data log-lik. (CLL) $\mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)]$ w.r.t. Θ
- The CLL (and expected CLL) for PPCA has a simple expression. The CLL is

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

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

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

  - M Step: Maximize the expected complete data log-lik. (CLL) $\mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)]$ w.r.t. Θ
- The CLL (and expected CLL) for PPCA has a simple expression. The CLL is

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

- Using $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left[-\frac{(\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)^\top(\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)}{2\sigma^2}\right]$

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

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

  - M Step: Maximize the expected complete data log-lik. (CLL) $\mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)]$ w.r.t. Θ
- The CLL (and expected CLL) for PPCA has a simple expression. The CLL is

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

- Using $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left[-\frac{(\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)^\top(\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)}{2\sigma^2}\right]$ and $p(\boldsymbol{z}_n) \propto \exp\left[-\frac{\boldsymbol{z}_n^\top \boldsymbol{z}_n}{2}\right]$

Parameter Estimation for PPCA via EM

- EM for PPCA iterates between the following two steps
  - E Step: Infer the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ given current estimate of $\Theta = (\mathbf{W}, \sigma^2)$ (needed for expectations)

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

  - M Step: Maximize the expected complete data log-lik. (CLL) $\mathbb{E}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)]$ w.r.t. Θ
- The CLL (and expected CLL) for PPCA has a simple expression. The CLL is

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

- Using $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \mathbf{W}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left[-\frac{(\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)^\top(\boldsymbol{x}_n - \mathbf{W}\boldsymbol{z}_n)}{2\sigma^2}\right]$ and $p(\boldsymbol{z}_n) \propto \exp\left[-\frac{\boldsymbol{z}_n^\top\boldsymbol{z}_n}{2}\right]$ and simplifying

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

Parameter Estimation for PPCA via EM

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

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

Parameter Estimation for PPCA via EM

- 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

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

(Exercise: verify; can also be done "online")

Parameter Estimation for PPCA via EM

- 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

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

(Exercise: verify; can also be done "online")

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

Parameter Estimation for PPCA via EM

- 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

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

(Exercise: verify; can also be done "online")

- To compute $\mathbf{W}$, we need two posterior expectations $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$
- These can be easily obtained from the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ computed in E step

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

Parameter Estimation for PPCA via EM

- 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

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

(Exercise: verify; can also be done "online")

- To compute $\mathbf{W}$, we need two posterior expectations $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$
- These can be easily obtained from the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ computed in E step

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

Parameter Estimation for PPCA via EM

- 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

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

(Exercise: verify; can also be done "online")

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

- These can be easily obtained from the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ computed in E step

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

Parameter Estimation for PPCA via EM

- 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

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

(Exercise: verify; can also be done "online")

- To compute $\mathbf{W}$, we need two posterior expectations $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$
- These can be easily obtained from the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ computed in E step

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

Parameter Estimation for PPCA via EM

- 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

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

(Exercise: verify; can also be done "online")

- To compute $\mathbf{W}$, we need two posterior expectations $\mathbb{E}[\boldsymbol{z}_n]$ and $\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$
- These can be easily obtained from the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ computed in E step

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

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

Summary: The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data ($\boldsymbol{x}_n = \boldsymbol{x}_n - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n$)

Summary: The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data ($\boldsymbol{x}_n = \boldsymbol{x}_n - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n$)
- **E step:** For each n, compute $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ using current $\mathbf{W}$ and σ^2. Compute exp. for the M step

Summary: The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data ($\boldsymbol{x}_n = \boldsymbol{x}_n - \frac{1}{N}\sum_{n=1}^N \boldsymbol{x}_n$)
- **E step:** For each n, compute $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ using current $\mathbf{W}$ and σ^2. Compute exp. for the M step

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

Summary: The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data ($\boldsymbol{x}_n = \boldsymbol{x}_n - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n$)
- **E step:** For each n, compute $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ using current $\mathbf{W}$ and σ^2. Compute exp. for the M step

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

Summary: The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data ($\boldsymbol{x}_n = \boldsymbol{x}_n - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n$)
- **E step:** For each n, compute $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ using current $\mathbf{W}$ and σ^2. Compute exp. for the M step

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

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

Summary: The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data ($\boldsymbol{x}_n = \boldsymbol{x}_n - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n$)
- **E step:** For each n, compute $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ using current $\mathbf{W}$ and σ^2. Compute exp. for the M step

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

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

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

Summary: The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data ($\boldsymbol{x}_n = \boldsymbol{x}_n - \frac{1}{N}\sum_{n=1}^N \boldsymbol{x}_n$)
- **E step:** For each n, compute $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ using current $\mathbf{W}$ and σ^2. Compute exp. for the M step

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

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

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

Summary: The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data ($\boldsymbol{x}_n = \boldsymbol{x}_n - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n$)
- **E step:** For each n, compute $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ using current $\mathbf{W}$ and σ^2. Compute exp. for the M step

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

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

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

- Set $\mathbf{W} = \mathbf{W}_{new}$ and $\sigma^2 = \sigma^2_{new}$. If not converged (monitor $p(\mathbf{X}|\Theta)$), go back to E step

Summary: The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data ($\boldsymbol{x}_n = \boldsymbol{x}_n - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n$)
- **E step:** For each n, compute $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ using current $\mathbf{W}$ and σ^2. Compute exp. for the M step

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

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

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

- Set $\mathbf{W} = \mathbf{W}_{new}$ and $\sigma^2 = \sigma^2_{new}$. If not converged (monitor $p(\mathbf{X}|\Theta)$), go back to E step
- **Note:** This EM algorithm can easily be made online (Exercise: Think how!)

Summary: The Full EM Algorithm for PPCA

- Specify K, initialize $\mathbf{W}$ and σ^2 randomly. Also center the data ($\boldsymbol{x}_n = \boldsymbol{x}_n - \frac{1}{N}\sum_{n=1}^N \boldsymbol{x}_n$)
- **E step:** For each n, compute $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ using current $\mathbf{W}$ and σ^2. Compute exp. for the M step

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

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

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

- Set $\mathbf{W} = \mathbf{W}_{new}$ and $\sigma^2 = \sigma^2_{new}$. If not converged (monitor $p(\mathbf{X}|\Theta)$), go back to E step
- **Note:** This EM algorithm can easily be made online (Exercise: Think how!)
- **Note:** For $\sigma^2 = 0$, this EM algorithm can also be used to efficiently solve standard PCA

EM for Factor Analysis

- Similar to PPCA except that $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a diagonal matrix

EM for Factor Analysis

- Similar to PPCA except that $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, 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 $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a diagonal matrix
- EM for Factor Analysis is very similar to that for PPCA
  - In the E step, for each n, compute the posterior over $\boldsymbol{z}_n$ and the required expectations

$$\begin{aligned} p(\boldsymbol{z}_n|\boldsymbol{x}_n, \mathbf{W}, \Psi) &= \mathcal{N}(\mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n, \mathbf{G}) \qquad \text{(Exercise: Verify)} \\ \mathbb{E}[\boldsymbol{z}_n] &= \mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n \\ \mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top] &= \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \mathbf{G}^{-1} \end{aligned}$$

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

EM for Factor Analysis

- Similar to PPCA except that $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a diagonal matrix
- EM for Factor Analysis is very similar to that for PPCA
    - In the E step, for each n, compute the posterior over $\boldsymbol{z}_n$ and the required expectations

$$
\begin{aligned}
p(\boldsymbol{z}_n|\boldsymbol{x}_n, \mathbf{W}, \Psi) &= \mathcal{N}(\mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n, \mathbf{G}) \qquad \text{(Exercise: Verify)} \\
\mathbb{E}[\boldsymbol{z}_n] &= \mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n \\
\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top] &= \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \mathbf{G}^{-1}
\end{aligned}
$$

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

EM for Factor Analysis

- Similar to PPCA except that $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a diagonal matrix
- EM for Factor Analysis is very similar to that for PPCA
    - In the E step, for each n, compute the posterior over $\boldsymbol{z}_n$ and the required expectations

$$
\begin{aligned}
p(\boldsymbol{z}_n|\boldsymbol{x}_n, \mathbf{W}, \Psi) &= \mathcal{N}(\mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n, \mathbf{G}) \qquad \text{(Exercise: Verify)} \\
\mathbb{E}[\boldsymbol{z}_n] &= \mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n \\
\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top] &= \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \mathbf{G}^{-1}
\end{aligned}
$$

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

    - In the M step, updates for $\mathbf{W}_{new}$ have the exact same for as PPCA

EM for Factor Analysis

- Similar to PPCA except that $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a diagonal matrix
- EM for Factor Analysis is very similar to that for PPCA
  - In the E step, for each n, compute the posterior over $\boldsymbol{z}_n$ and the required expectations

$$\begin{aligned}
p(\boldsymbol{z}_n|\boldsymbol{x}_n, \mathbf{W}, \Psi) &= \mathcal{N}(\mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n, \mathbf{G}) \qquad \text{(Exercise: Verify)} \\
\mathbb{E}[\boldsymbol{z}_n] &= \mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n \\
\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top] &= \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \mathbf{G}^{-1}
\end{aligned}$$

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

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

EM for Factor Analysis

- Similar to PPCA except that $\boldsymbol{x}_n \sim \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a diagonal matrix
- EM for Factor Analysis is very similar to that for PPCA
  - In the E step, for each n, compute the posterior over $\boldsymbol{z}_n$ and the required expectations

$$\begin{aligned} p(\boldsymbol{z}_n|\boldsymbol{x}_n, \mathbf{W}, \Psi) &= \mathcal{N}(\mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n, \mathbf{G}) \qquad \text{(Exercise: Verify)} \\ \mathbb{E}[\boldsymbol{z}_n] &= \mathbf{G}^{-1}\mathbf{W}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n \\ \mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top] &= \mathbb{E}[\boldsymbol{z}_n]\mathbb{E}[\boldsymbol{z}_n]^\top + \mathbf{G}^{-1} \end{aligned}$$

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

  - In the M step, updates for $\mathbf{W}_{new}$ have the exact same for as PPCA
  - In the M step, updates for $\boldsymbol{\Psi}$ are

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

- This EM algorithm too can be easily made online!

How to Set “K”?

- Several option to select the “best” K, e.g.,

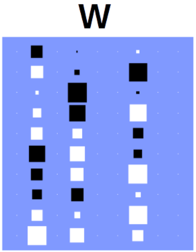
How to Set “K”?

- Several option to select the “best” K, e.g.,
  - Compute the marginal likelihood (or its approximation) for each K and choose the best model

How to Set "K"?

- Several option to select the "best" K, e.g.,
  - Compute the marginal likelihood (or its approximation) for each K and choose the best model
  - Use sparsity inducing priors on $\mathbf{W}$ and/or $\boldsymbol{z}_n$ (set K to some large value; the unnecessary columns of $\mathbf{W}$ will "turn off" automatically as they will be shrunk to zero during inference)

Using sparsity-inducing Prior (e.g., Automatic Relevance Determination) on **W**

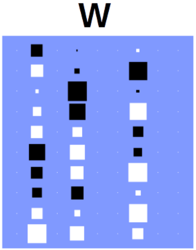


Effect: Only few columns of **W** will have entries with significant magnitudes

How to Set "K"?

- Several option to select the "best" K, e.g.,
  - Compute the marginal likelihood (or its approximation) for each K and choose the best model
  - Use sparsity inducing priors on $\mathbf{W}$ and/or $\boldsymbol{z}_n$ (set K to some large value; the unnecessary columns of $\mathbf{W}$ will "turn off" automatically as they will be shrunk to zero during inference)

Using sparsity-inducing Prior (e.g., Automatic Relevance Determination) on **W**



Effect: Only few columns of **W** will have entries with significant magnitudes

  - Nonparametric Bayesian methods (allow K to grow with data)

Some Applications of PPCA/FA

- Compression/dimensionality reduction is a natural application (use $\boldsymbol{z}_n$ instead of $\boldsymbol{x}_n$)

Some Applications of PPCA/FA

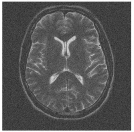
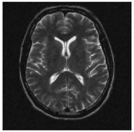
- Compression/dimensionality reduction is a natural application (use $\boldsymbol{z}_n$ instead of $\boldsymbol{x}_n$)
- Also used for learning low-dim. "good" features $\boldsymbol{z}_n$ from high-dim noisy features $\boldsymbol{x}_n$

Some Applications of PPCA/FA

- Compression/dimensionality reduction is a natural application (use $\boldsymbol{z}_n$ instead of $\boldsymbol{x}_n$)
- Also used for learning low-dim. "good" features $\boldsymbol{z}_n$ from high-dim noisy features $\boldsymbol{x}_n$
  - Note that this is different from feature selection ($\boldsymbol{z}_n$ is a transformed version of $\boldsymbol{x}_n$, not a subset)

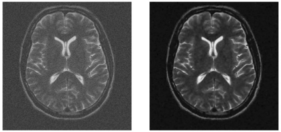
Some Applications of PPCA/FA

- Compression/dimensionality reduction is a natural application (use $\boldsymbol{z}_n$ instead of $\boldsymbol{x}_n$)
- Also used for learning low-dim. "good" features $\boldsymbol{z}_n$ from high-dim noisy features $\boldsymbol{x}_n$
  - Note that this is different from feature selection ($\boldsymbol{z}_n$ is a transformed version of $\boldsymbol{x}_n$, not a subset)
- Learning the noise variance enables "image denoising": $\boldsymbol{x}_n = \mathbf{W}\boldsymbol{z}_n + \epsilon_n$; $\mathbf{W}\boldsymbol{z}_n$ is the "clean" part

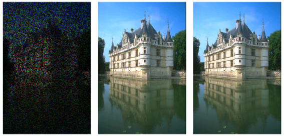
 

Some Applications of PPCA/FA

- Compression/dimensionality reduction is a natural application (use $\boldsymbol{z}_n$ instead of $\boldsymbol{x}_n$)
- Also used for learning low-dim. "good" features $\boldsymbol{z}_n$ from high-dim noisy features $\boldsymbol{x}_n$
    - Note that this is different from feature selection ($\boldsymbol{z}_n$ is a transformed version of $\boldsymbol{x}_n$, not a subset)
- Learning the noise variance enables "image denoising": $\boldsymbol{x}_n = \mathbf{W}\boldsymbol{z}_n + \epsilon_n$; $\mathbf{W}\boldsymbol{z}_n$ is the "clean" part



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



Mixture of PPCA and Mixture of FA

- May be appropriate if data also exists in clusters (suppose $M > 1$ clusters)

Mixture of PPCA and Mixture of FA

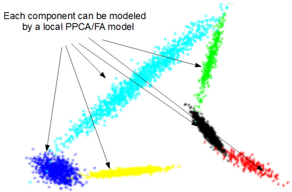
- May be appropriate if data also exists in clusters (suppose $M > 1$ clusters)
- Data in each cluster (say m) can have its own "local" PPCA/FA model defined by $\{\mathbf{W}_m, \sigma_m^2\}$

Mixture of PPCA and Mixture of FA

- May be appropriate if data also exists in clusters (suppose $M > 1$ clusters)
- Data in each cluster (say m) can have its own "local" PPCA/FA model defined by $\{\mathbf{W}_m, \sigma_m^2\}$
- Can use M such PPCA/FA models $\{\mathbf{W}_m, \sigma_m^2\}_{m=1}^M$ (one per cluster) for the entire data

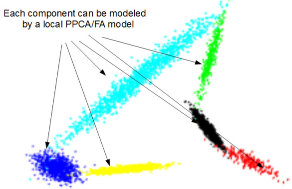
Mixture of PPCA and Mixture of FA

- May be appropriate if data also exists in clusters (suppose $M > 1$ clusters)
- Data in each cluster (say m) can have its own "local" PPCA/FA model defined by $\{\mathbf{W}_m, \sigma_m^2\}$
- Can use M such PPCA/FA models $\{\mathbf{W}_m, \sigma_m^2\}_{m=1}^M$ (one per cluster) for the entire data



Mixture of PPCA and Mixture of FA

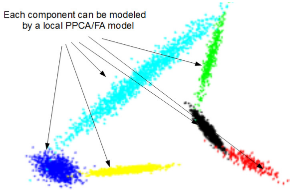
- May be appropriate if data also exists in clusters (suppose $M > 1$ clusters)
- Data in each cluster (say m) can have its own "local" PPCA/FA model defined by $\{\mathbf{W}_m, \sigma_m^2\}$
- Can use M such PPCA/FA models $\{\mathbf{W}_m, \sigma_m^2\}_{m=1}^M$ (one per cluster) for the entire data



- Mixtures of PPCA/FA can be seen as playing several roles

Mixture of PPCA and Mixture of FA

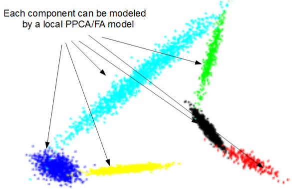
- May be appropriate if data also exists in clusters (suppose $M > 1$ clusters)
- Data in each cluster (say m) can have its own "local" PPCA/FA model defined by $\{\mathbf{W}_m, \sigma_m^2\}$
- Can use M such PPCA/FA models $\{\mathbf{W}_m, \sigma_m^2\}_{m=1}^M$ (one per cluster) for the entire data



- Mixtures of PPCA/FA can be seen as playing several roles
  - Jointly learning clustering and dimensionality reduction

Mixture of PPCA and Mixture of FA

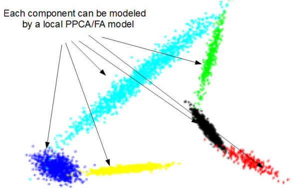
- May be appropriate if data also exists in clusters (suppose $M > 1$ clusters)
- Data in each cluster (say m) can have its own "local" PPCA/FA model defined by $\{\mathbf{W}_m, \sigma_m^2\}$
- Can use M such PPCA/FA models $\{\mathbf{W}_m, \sigma_m^2\}_{m=1}^M$ (one per cluster) for the entire data



- Mixtures of PPCA/FA can be seen as playing several roles
  - Jointly learning clustering and dimensionality reduction
  - Nonlinear dimensionality reduction

Mixture of PPCA and Mixture of FA

- May be appropriate if data also exists in clusters (suppose $M > 1$ clusters)
- Data in each cluster (say m) can have its own "local" PPCA/FA model defined by $\{\mathbf{W}_m, \sigma_m^2\}$
- Can use M such PPCA/FA models $\{\mathbf{W}_m, \sigma_m^2\}_{m=1}^{M}$ (one per cluster) for the entire data



- Mixtures of PPCA/FA can be seen as playing several roles
  - Jointly learning clustering and dimensionality reduction
  - Nonlinear dimensionality reduction
  - A flexible probability density model: Mixture of low-rank Gaussians

Mixture of PPCA and Mixture of FA

- For mixture of PPCA/FA, the generative story for each observation $\boldsymbol{x}_n$ is as follows

Mixture of PPCA and Mixture of FA

- For mixture of PPCA/FA, the generative story for each observation $\boldsymbol{x}_n$ is as follows
  - Generate its cluster id as

$$\boldsymbol{c}_n \sim \text{multinoulli}(\pi_1, \ldots, \pi_M)$$

Mixture of PPCA and Mixture of FA

- For mixture of PPCA/FA, the generative story for each observation $\boldsymbol{x}_n$ is as follows
  - Generate its cluster id as
$$\boldsymbol{c}_n \sim \text{multinoulli}(\pi_1, \ldots, \pi_M)$$
  - Generate latent variable $\boldsymbol{z}_n \in \mathbb{R}^K$ as
$$\boldsymbol{z}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$$

Mixture of PPCA and Mixture of FA

- For mixture of PPCA/FA, the generative story for each observation $\boldsymbol{x}_n$ is as follows
  - Generate its cluster id as
$$\boldsymbol{c}_n \sim \text{multinoulli}(\pi_1, \ldots, \pi_M)$$
  - Generate latent variable $\boldsymbol{z}_n \in \mathbb{R}^K$ as
$$\boldsymbol{z}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$$
  - Generate obervation $\boldsymbol{x}_n \in \mathbb{R}^D$ from the $\boldsymbol{c}_n^{th}$ PPCA/FA model
$$\boldsymbol{x}_n \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{c}_n} + \mathbf{W}_{\boldsymbol{c}_n}\boldsymbol{z}_n, \sigma^2_{\boldsymbol{c}_n}\mathbf{I}_D)$$

Mixture of PPCA and Mixture of FA

- For mixture of PPCA/FA, the generative story for each observation $\boldsymbol{x}_n$ is as follows
  - Generate its cluster id as
$$\boldsymbol{c}_n \sim \text{multinoulli}(\pi_1, \ldots, \pi_M)$$
  - Generate latent variable $\boldsymbol{z}_n \in \mathbb{R}^K$ as
$$\boldsymbol{z}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$$
  - Generate obervation $\boldsymbol{x}_n \in \mathbb{R}^D$ from the $\boldsymbol{c}_n^{th}$ PPCA/FA model
$$\boldsymbol{x}_n \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{c}_n} + \mathbf{W}_{\boldsymbol{c}_n}\boldsymbol{z}_n, \sigma^2_{\boldsymbol{c}_n}\mathbf{I}_D)$$
- Each PPCA/FA model has its separate mean $\boldsymbol{\mu}_{\boldsymbol{c}_n}$ (not needed when $M = 1$ if data is centered)

Mixture of PPCA and Mixture of FA

- For mixture of PPCA/FA, the generative story for each observation $\boldsymbol{x}_n$ is as follows
  - Generate its cluster id as
  $$\boldsymbol{c}_n \sim \text{multinoulli}(\pi_1, \ldots, \pi_M)$$
  - Generate latent variable $\boldsymbol{z}_n \in \mathbb{R}^K$ as
  $$\boldsymbol{z}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$$
  - Generate obervation $\boldsymbol{x}_n \in \mathbb{R}^D$ from the $\boldsymbol{c}_n^{th}$ PPCA/FA model
  $$\boldsymbol{x}_n \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{c}_n} + \mathbf{W}_{\boldsymbol{c}_n}\boldsymbol{z}_n, \sigma^2_{\boldsymbol{c}_n}\mathbf{I}_D)$$
- Each PPCA/FA model has its separate mean $\boldsymbol{\mu}_{\boldsymbol{c}_n}$ (not needed when $M = 1$ if data is centered)
- **Exercise:** What will be the marginal distribution of $\boldsymbol{x}_n$, i.e.. $p(\boldsymbol{x}_n|\Theta)$?

Mixture of PPCA and Mixture of FA

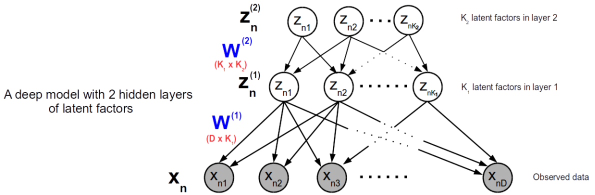
- For mixture of PPCA/FA, the generative story for each observation $\boldsymbol{x}_n$ is as follows
  - Generate its cluster id as
$$\boldsymbol{c}_n \sim \text{multinoulli}(\pi_1, \ldots, \pi_M)$$
  - Generate latent variable $\boldsymbol{z}_n \in \mathbb{R}^K$ as
$$\boldsymbol{z}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$$
  - Generate obervation $\boldsymbol{x}_n \in \mathbb{R}^D$ from the $\boldsymbol{c}_n^{th}$ PPCA/FA model
$$\boldsymbol{x}_n \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{c}_n} + \mathbf{W}_{\boldsymbol{c}_n}\boldsymbol{z}_n, \sigma^2_{\boldsymbol{c}_n}\mathbf{I}_D)$$
- Each PPCA/FA model has its separate mean $\boldsymbol{\mu}_{\boldsymbol{c}_n}$ (not needed when $M = 1$ if data is centered)
- **Exercise:** What will be the marginal distribution of $\boldsymbol{x}_n$, i.e.. $p(\boldsymbol{x}_n|\Theta)$?
- EM can be used in this model to learn the parameters and latent variables

PPCA/FA Extensions

- Already saw the mixture of PPCA/FA

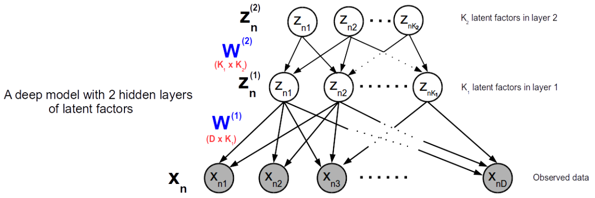
PPCA/FA Extensions

- Already saw the mixture of PPCA/FA
- Can stack PPCA/FA models to construct deep generative models (more on this later)

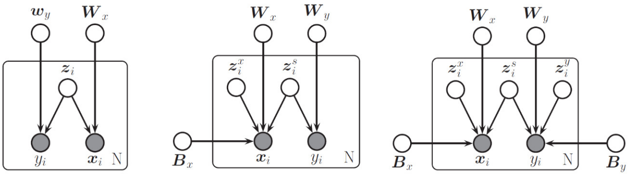


PPCA/FA Extensions

- Already saw the mixture of PPCA/FA
- Can stack PPCA/FA models to construct deep generative models (more on this later)



- Many other extensions for supervised dim-red, multi-modality data such as image+caption, etc.



Some Concluding Thoughts

- Latent variable models for dim-red have many advantages over non-probabilistic counterparts

Some Concluding Thoughts

- Latent variable models for dim-red have many advantages over non-probabilistic counterparts
  - Brings in all the benefits of probabilistic modeling with diverse likelihood/priors

Some Concluding Thoughts

- Latent variable models for dim-red have many advantages over non-probabilistic counterparts
  - Brings in all the benefits of probabilistic modeling with diverse likelihood/priors
  - Can be easily extended to learning more complex patterns (e.g., mixture of PPCA)

Some Concluding Thoughts

- Latent variable models for dim-red have many advantages over non-probabilistic counterparts
  - Brings in all the benefits of probabilistic modeling with diverse likelihood/priors
  - Can be easily extended to learning more complex patterns (e.g., mixture of PPCA)
  - Often more efficient too (e.g., using PPCA with EM can be faster than PCA with eigen-decomp.)

Some Concluding Thoughts

- Latent variable models for dim-red have many advantages over non-probabilistic counterparts
  - Brings in all the benefits of probabilistic modeling with diverse likelihood/priors
  - Can be easily extended to learning more complex patterns (e.g., mixture of PPCA)
  - Often more efficient too (e.g., using PPCA with EM can be faster than PCA with eigen-decomp.)
  - Handle missing data in a principled way

Some Concluding Thoughts

- Latent variable models for dim-red have many advantages over non-probabilistic counterparts
  - Brings in all the benefits of probabilistic modeling with diverse likelihood/priors
  - Can be easily extended to learning more complex patterns (e.g., mixture of PPCA)
  - Often more efficient too (e.g., using PPCA with EM can be faster than PCA with eigen-decomp.)
  - Handle missing data in a principled way
  - Nonlinear maps from $\boldsymbol{z}$ to $\boldsymbol{x}$ possible (e.g., using deep neural nets or Gaussian Processes)

Some Concluding Thoughts

- Latent variable models for dim-red have many advantages over non-probabilistic counterparts
  - Brings in all the benefits of probabilistic modeling with diverse likelihood/priors
  - Can be easily extended to learning more complex patterns (e.g., mixture of PPCA)
  - Often more efficient too (e.g., using PPCA with EM can be faster than PCA with eigen-decomp.)
  - Handle missing data in a principled way
  - Nonlinear maps from $\boldsymbol{z}$ to $\boldsymbol{x}$ possible (e.g., using deep neural nets or Gaussian Processes)
- These models can also be used for estimating the probability density $p(\boldsymbol{x})$

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z},\theta)p(\boldsymbol{z}|\phi)d\boldsymbol{z}$$

Some Concluding Thoughts

- Latent variable models for dim-red have many advantages over non-probabilistic counterparts
  - Brings in all the benefits of probabilistic modeling with diverse likelihood/priors
  - Can be easily extended to learning more complex patterns (e.g., mixture of PPCA)
  - Often more efficient too (e.g., using PPCA with EM can be faster than PCA with eigen-decomp.)
  - Handle missing data in a principled way
  - Nonlinear maps from $\boldsymbol{z}$ to $\boldsymbol{x}$ possible (e.g., using deep neural nets or Gaussian Processes)
- These models can also be used for estimating the probability density $p(\boldsymbol{x})$

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z},\theta)p(\boldsymbol{z}|\phi)d\boldsymbol{z}$$

- Can also be extended to model sequential data, e.g., Linear Dynamical Systems (will see later)

Some Concluding Thoughts

- Latent variable models for dim-red have many advantages over non-probabilistic counterparts
  - Brings in all the benefits of probabilistic modeling with diverse likelihood/priors
  - Can be easily extended to learning more complex patterns (e.g., mixture of PPCA)
  - Often more efficient too (e.g., using PPCA with EM can be faster than PCA with eigen-decomp.)
  - Handle missing data in a principled way
  - Nonlinear maps from $\boldsymbol{z}$ to $\boldsymbol{x}$ possible (e.g., using deep neural nets or Gaussian Processes)
- These models can also be used for estimating the probability density $p(\boldsymbol{x})$

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z}, \theta)p(\boldsymbol{z}|\phi)d\boldsymbol{z}$$

- Can also be extended to model sequential data, e.g., Linear Dynamical Systems (will see later)
- EM is only one of the ways to do inference in LVMs. Can also do fully Bayesian inference

Some Concluding Thoughts

- Latent variable models for dim-red have many advantages over non-probabilistic counterparts
  - Brings in all the benefits of probabilistic modeling with diverse likelihood/priors
  - Can be easily extended to learning more complex patterns (e.g., mixture of PPCA)
  - Often more efficient too (e.g., using PPCA with EM can be faster than PCA with eigen-decomp.)
  - Handle missing data in a principled way
  - Nonlinear maps from $\boldsymbol{z}$ to $\boldsymbol{x}$ possible (e.g., using deep neural nets or Gaussian Processes)
- These models can also be used for estimating the probability density $p(\boldsymbol{x})$

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z}, \theta)p(\boldsymbol{z}|\phi)d\boldsymbol{z}$$

- Can also be extended to model sequential data, e.g., Linear Dynamical Systems (will see later)
- EM is only one of the ways to do inference in LVMs. Can also do fully Bayesian inference
  - More on this when we discuss MCMC and Variational Inference