COMPSCI 689
Lecture 19: Latent Linear Models

Brendan O'Connor

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Probabilistic Unsupervised Learning

- In probabilistic unsupervised learning, our goal is to model multivariate data $\mathbf{x} = [x_1, ..., x_D]$ generated by an unknown probabilistic process using a probabilistic model learned from a data set $\mathcal{D} = \{\mathbf{x}_n | 1 \leq n \leq N\}$.
- Since the data are vectors, we use vector-valued random variables to model them $\mathbf{X} = [X_1, ..., X_D]$.
- Each data dimension d takes values from a potentially different set $\mathcal{X}_d$. We have $\mathbf{x} \in \mathcal{X}$. $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_D$.

Multivariate Normal

- The multivariate normal (or Gaussian) distribution is a fundamental building block for unsupervised learning with vector-valued random variables $\mathbf{X} \in \mathbb{R}^D$.
- The distribution has two parameters $\theta = [\mu, \Sigma]$. μ is the mean vector and Σ is the covariance matrix.
- The probability density is given below (assuming $\mathbf{x}$ and μ are column vectors):

$$\mathcal{N}(\mathbf{x}; \mu, \Sigma) = \frac{1}{|2\pi\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\Big)$$

- We have $\mu \in \mathbb{R}^D$ and $\Sigma \in \mathbb{S}_+^D$, the space of symmetric, positive definite $D \times D$ matrices.

Example: Bivariate Normal

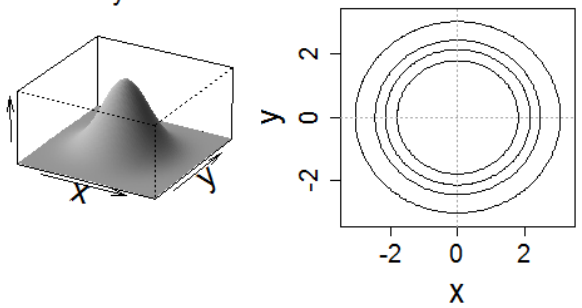
- The bivariate normal distribution is a special case of the multivariate normal where $D = 2$ so that $\mathbf{X} = [X_1, X_2]$.
- In this case the mean vector $\mu = [\mu_1, \mu_2]$ specifies a location in 2D real space.
- The covariance matrix can be represented either directly or via the marginal standard deviations and the correlation between X_1 and X_2:

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12} & \Sigma_{22} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

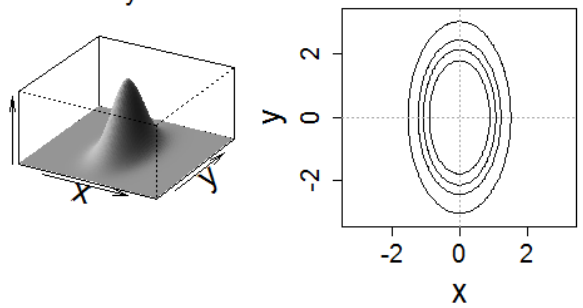
- The level sets of the bivariate normal density are ellipses whose axes are determined by the eigenvalues and eigenvectors of the covariance matrix.

Example: Bivariate Normal

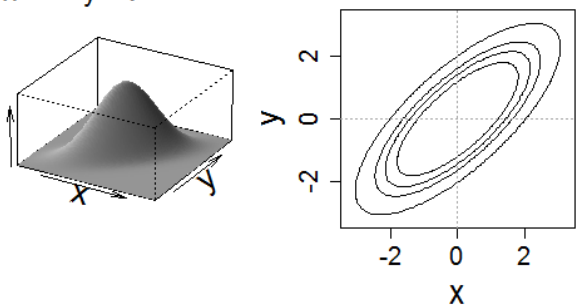
$\sigma_x = \sigma_y,\ \rho = 0$



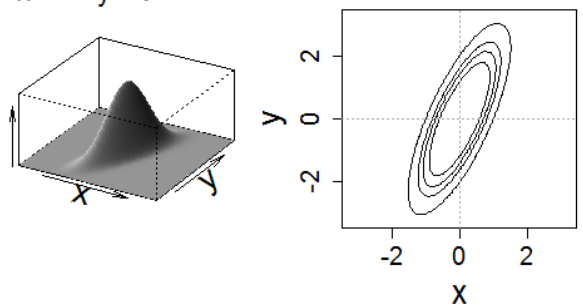
$2\sigma_x = \sigma_y,\ \rho = 0$



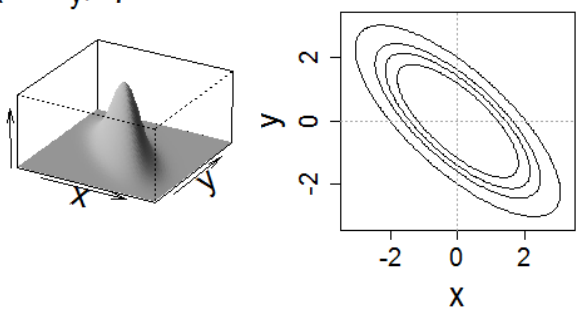
$\sigma_x = \sigma_y,\ \rho = 0.75$



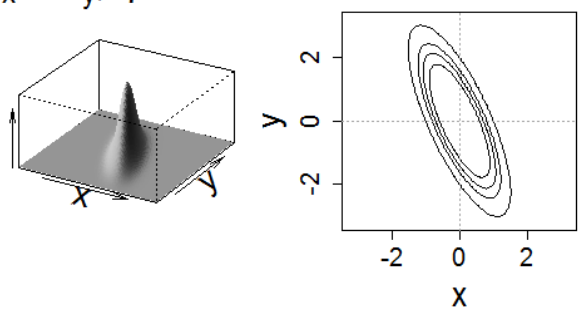
$2\sigma_x = \sigma_y,\ \rho = 0.75$



$\sigma_x = \sigma_y,\ \rho = -0.75$



$2\sigma_x = \sigma_y,\ \rho = -0.75$



MLE for the Multivariate Normal

- Given a data set $\mathcal{D} = \{\mathbf{x}_n\}_{1:N}$, the MLE for the multivariate normal is found by solving the optimization problem:

$$\mu^*, \Sigma^* = \arg\min_{\mu,\Sigma} -\sum_{n=1}^{N} \log \mathcal{N}(\mathbf{x}_n; \mu, \Sigma)$$

- The solutions are:

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n, \qquad \hat{\Sigma} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{x}_n - \hat{\mu})(\mathbf{x}_n - \hat{\mu})^T$$

Other Special Cases of MVNs

- Consider the general case for arbitrary D.
- If $\mu = [0, ..., 0]$ and $\Sigma = I$ is the identity matrix, $\mathcal{N}(\mathbf{x}; \mu, \Sigma)$ is called a “a standard multivariate normal".
- If $\Sigma = \sigma I$, $\mathcal{N}(\mathbf{x}; \mu, \Sigma)$ is called an “isotropic Gaussian."
- If Σ is a diagonal matrix, $\mathcal{N}(\mathbf{x}; \mu, \Sigma)$ is called a “diagonal Gaussian" or “axis-aligned Gaussian"

Marginalization

- Suppose we have a joint distribution on a vector-valued random variable $\mathbf{X} \in \mathbb{R}^D$. Let $A \subseteq \{1, ..., D\}$, $M = |A|$, and $\mathbf{X}_A = [X_{A_1}, ..., X_{A_M}]$.
- The probability distribution $P(\mathbf{X}_A = \mathbf{x}_A)$ is called the *marginal distribution* of $\mathbf{X}_A$.
- Let $B = \{1, ..., D\}/A$. The marginal distribution of $\mathbf{X}_A$ is then given by:

$$P(\mathbf{X}_A = \mathbf{x}_A) = \int_{\mathcal{X}_B} P(\mathbf{X}_A = \mathbf{x}_A, \mathbf{X}_B = \mathbf{x}_B) d\mathbf{x}_B$$

Marginalization for MVNs

- The multivariate normal distribution has the remarkable (and convenient) property of being closed under marginalization.
- Suppose we have an MVN $P(\mathbf{X}|\theta) = \mathcal{N}(\mathbf{X}; \mu, \Sigma)$ for $\mathbf{X} \in \mathbb{R}^D$. Let $A \subseteq \{1, ..., D\}$, $B = \{1, ..., D\}/A$, and $M = |A|$. We have:

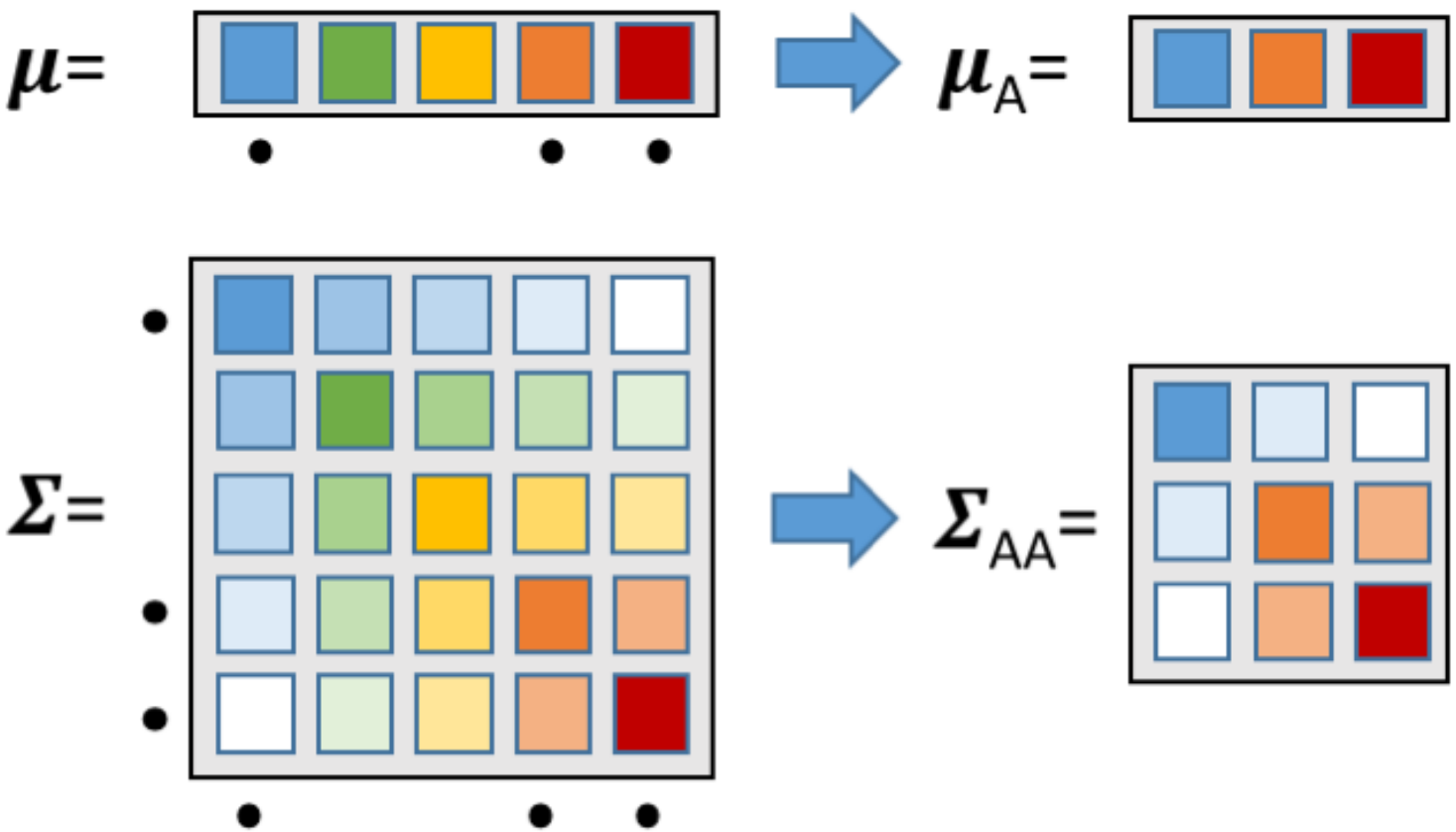
$$P(\mathbf{X}_A = \mathbf{x}_A) = \mathcal{N}(\mu_A, \Sigma_{AA})$$

  where $\mu_A = [\mu_{A_1}, ..., \mu_{A_M}]$ and $(\Sigma_{AA})_{ij} = \Sigma_{A_i, A_j}$.

- In other words, we get the marginal distribution on a subset of $\mathbf{X}$ just by discarding the elements of μ that correspond to B, and the rows and columns of Σ that correspond to B.

Marginalization for MVNs: Example

$A = \{1,4,5\}$



Conditioning for MVNs

- The multivariate normal distribution has the remarkable (and convenient) property of also being closed under conditioning.
- Suppose we have an MVN $p(\mathbf{X}|\theta) = \mathcal{N}(\mathbf{X}; \mu, \Sigma)$ for $\mathbf{X} \in \mathbb{R}^D$. Let $A \subseteq \{1, ..., D\}$, $B = \{1, ..., D\} \backslash A$. We have:

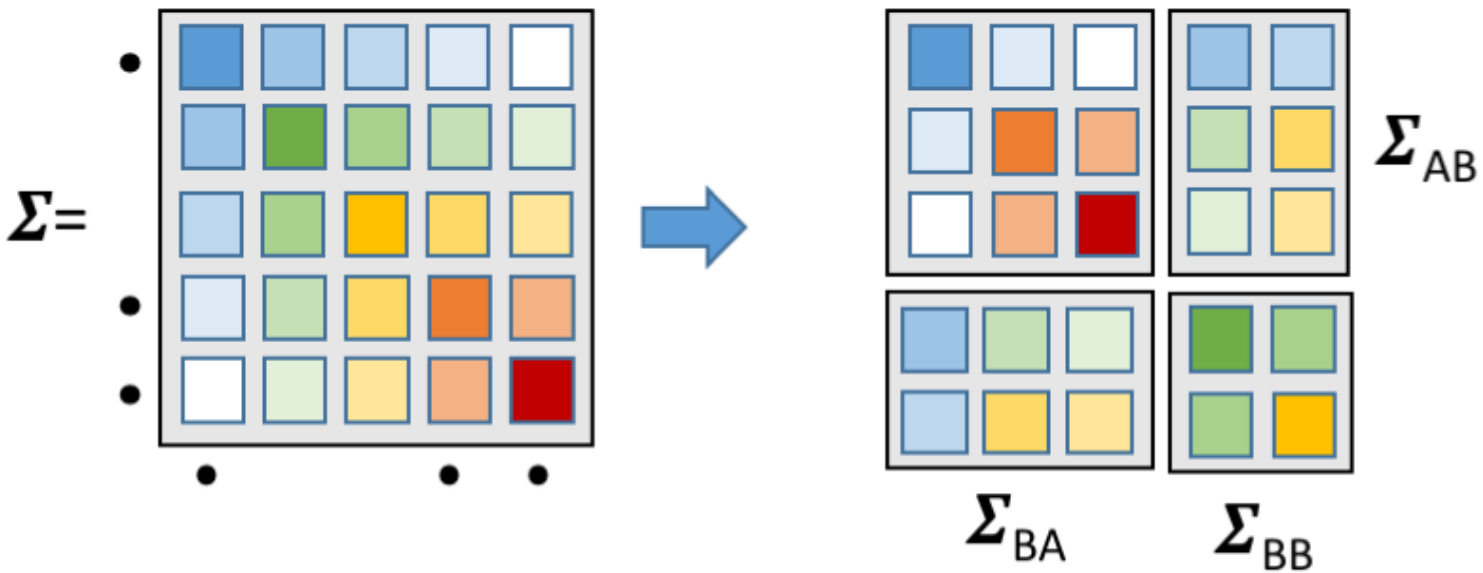
$$p(\mathbf{X}_A = \mathbf{x}_A | \mathbf{X}_B = \mathbf{x}_B) = \mathcal{N}(\mathbf{x}_A; \mu_{A|B}, \Sigma_{AA|B})$$

$$\mu_{A|B} = \mu_A + \Sigma_{AB}(\Sigma_{BB})^{-1}(\mathbf{x}_B - \mu_B)$$

$$\Sigma_{AA|B} = \Sigma_{AA} - \Sigma_{AB}(\Sigma_{BB})^{-1}\Sigma_{BA}$$

Conditioning for MVNs: Example

A= {1,4,5}, B={2,3}



Factor Analysis

- Factor analysis is a classical statistical model for linear manifolds based on the multivariate normal distribution.
- The model asserts that real-valued data $\mathbf{x} \in \mathbb{R}^D$ are generated in a two stage process that starts by first generating a low-dimensional latent factor vector $\mathbf{z} \in \mathbb{R}^K$ from a multivariate normal distribution.
- The observed $\mathbf{x}$'s are then generated by a linear combination of basis vectors weighted by the latent factor values: $\mathbf{Wz}$ with independent Gaussian noise added.
- The matrix $\mathbf{W}$ has size $D \times K$. Each column of $\mathbf{W}$ corresponds to a basis vector.

Factor Analysis: Probabilistic Model

- The probabilistic model/generative process for factor analysis is shown below:

$$
\begin{aligned}
p(\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z}) &= p(\mathbf{X} = \mathbf{x} | \mathbf{Z} = \mathbf{z}) p(\mathbf{Z} = \mathbf{z}) \\
p(\mathbf{Z} = \mathbf{z}) &= \mathcal{N}(\mathbf{z}; 0, I) \\
p(\mathbf{X} = \mathbf{x} | \mathbf{Z} = \mathbf{z}) &= \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z} + \mu, \Psi)
\end{aligned}
$$

- Ψ is restricted to be a positive, diagonal matrix. We can learn μ, or simply remove the data set mean and require $\mu = 0$. We will assume the data mean has been removed and thus the optimal value of μ is 0.

Factor Analysis: Marginal Distribution

- The marginal distribution of $\mathbf{X}$ is is given by:

$$\begin{aligned} P(\mathbf{X} = \mathbf{x}) &= \int \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z}, \Psi)\mathcal{N}(\mathbf{z}; 0, I)d\mathbf{z} \\ &= \mathcal{N}(\mathbf{x}; 0, \mathbf{W}\mathbf{W}^T + \Psi) \end{aligned}$$

Factor Analysis: Learning

- To learn the factor analysis model, we need to minimize the negative log marginal likelihood:

$$\begin{aligned}\text{nlml}(\mathcal{D}, \theta) &= -\sum_{n=1}^{N} \log \mathcal{N}(\mathbf{x}_n; 0, \mathbf{W}\mathbf{W}^T + \Psi) \\ &= \frac{N}{2} \log(|2\pi(\mathbf{W}\mathbf{W}^T + \Psi)|) + \frac{1}{2} \sum_{n=1}^{N} \mathbf{x}_n^T (\mathbf{W}\mathbf{W}^T + \Psi)^{-1} \mathbf{x}_n\end{aligned}$$

- Question: To learn the model via direct NLML minimization, what parameter constraints do we need to enforce?

Factor Analysis: Generation/Decoding

- A learned factor analysis model can be used as a *generator*.
- We can choose any vector $\mathbf{z}$, plug it in to the model, and obtain the mean of $p(\mathbf{x}|\mathbf{z})$ as $\mathbf{W}\mathbf{z}$.
- If we want a probabilistic generator, we can sample from $p(\mathbf{x}|\mathbf{z})$.
- This generate a new data case $\mathbf{x}$ based on the latent code $\mathbf{z}$ that we supplied.
- This process is also referred to a *decoding*

Factor Analysis: Dimensionality Reduction/Encoding

- A learned factor analysis model can be used as a probabilistic dimensionality reduction model.
- Given a centered value for $\mathbf{x}$, we need to infer the probability distribution on the low-dimensional code $\mathbf{z}$. We have:

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \bar{\mathbf{z}}, \mathbf{S})$$
$$\mathbf{S} = (I + \mathbf{W}^T \Psi \mathbf{W})^{-1}$$
$$\bar{\mathbf{z}} = \mathbf{S}\mathbf{W}^T \Psi^{-1} \mathbf{x}$$

- $\bar{\mathbf{z}}$ is obtained via a linear projection from D dimensional space to K dimensional space. This process is referred to as *encoding*.

Factor Analysis: Reconstruction

- A learned factor analysis model can be used to “reconstruct" an input $\mathbf{x}$ by first encoding $\mathbf{x}$ into $\bar{\mathbf{z}}$, then decoding it back into $\mathbf{x}'$.
- This process can be useful for solving unsupervised de-noising tasks.
- Learning: As with mixture models, we can simply maximize the sum of the log marginal likelihoods over a data set to learn the model parameters Ψ and $\mathbf{W}$. Alternatively, EM is also possible.

Factor Analysis: Comparisons

- FA suffers from rotation invariance. For interpretation, make sure to use a rotation choice method like Varimax. Nearly a century of arguments about this!
- Principal Components Analysis: can be viewed as a limiting case of FA, with additional constraint of orthogonality.
- FA can be viewed as a *linear autoencoder*. Nonlinear AEs via neural networks are an obvious alternative.

Linear Autoencoders

- A linear autoencoder is an autoencoder where the encoder and decoder are linear functions.
- The encoding and decoding functions are:

$$f(\mathbf{x}) = \mathbf{V}\mathbf{x}$$
$$g(\mathbf{h}) = \mathbf{W}\mathbf{h}$$

- Such a linear encoder-decoder model can be learned by minimizing the MSE between the inputs and the reconstructions, often called the *reconstruction error*.

$$\mathbf{V}^*, \mathbf{W}^* = \arg\min_{\mathbf{V},\mathbf{W}} \sum_{n=1}^{N} \|\mathbf{x}_n - g(f(\mathbf{x}_n))\|_2^2$$

PPCA as an Autoencoder

- The Probabilistic Principal Components Analysis (PPCA) model is a special case of Factor Analysis where $\Psi = \sigma^2 I$ and $\mathbf{W}$ is constrained to be an orthogonal matrix (e.g.,$\mathbf{W}^T\mathbf{W} = I$).
- The encoder function is the mean of $P(\mathbf{z}|\mathbf{x})$, but the isotropic assumption assumption results in simplifications:

$$f(\mathbf{x}) = \mathbb{E}_{P(\mathbf{z}|\mathbf{x})}[\mathbf{z}] = \mathbf{V}\mathbf{x}$$
$$\mathbf{V} = (\sigma^2 I + \mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T$$

- The decoder function is simply the mean function of $P(\mathbf{x}|\mathbf{z})$:

$$g(\mathbf{z}) = \mathbb{E}_{P(\mathbf{x}|\mathbf{z})}[\mathbf{x}] = \mathbf{W}\mathbf{z}$$

Classical Principal Components Analysis as an Autoencoder

- In the limit as σ^2 goes to zero, we obtain the classical PCA model. The encoder further simplifies due to orthogonality of $\mathbf{W}$:

$$f(\mathbf{x}) = \mathbb{E}_{P(\mathbf{z}|\mathbf{x})}[\mathbf{z}] = \mathbf{V}\mathbf{x}$$
$$\mathbf{V} = (\sigma^2 I + \mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T \rightarrow (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T = \mathbf{W}^T$$

- The decoder function is again the mean of $P(\mathbf{x}|\mathbf{z})$:

$$g(\mathbf{z}) = \mathbb{E}_{P(\mathbf{x}|\mathbf{z})}[\mathbf{x}] = \mathbf{W}\mathbf{z}$$

Classical Principal Components Analysis

- One of the interesting properties of classical PCA is that you do not need iterative numerical optimization to find the optimal $\mathbf{W}$.
- Let $\mathbf{X}$ be an $N \times D$ data matrix that has been mean centered.
- Define the empirical scatter matrix to be $\mathbf{S} = \mathbf{X}^T\mathbf{X}$.
- Then the optimal $\mathbf{W}$ is given by the leading K eigenvectors of $\mathbf{S}$.
- These eigenvectors are orthogonal by definition and are the rank K maximum variance sub-space of $\mathbf{X}$, also called the *principal sub-space*.
- Interestingly, this estimation approach also identifies the optimal $\mathbf{W}$ for the PPCA model.

Singular Value Decomposition

- The rank K singular value decomposition (SVD) of $\mathbf{X}$ is given by $\mathbf{USV}^T$ where $\mathbf{U}$ is an $N \times K$ orthonormal matrix, $\mathbf{U}$ is a $D \times K$ orthonormal matrix and $\mathbf{S}$ is a positive diagonal matrix.
- The rank K SVD of $\mathbf{X}$ minimizes the objective function $\|\mathbf{X} - \mathbf{USV}^T\|_2^2$ subject to the stated conditions on $\mathbf{U}, \mathbf{S}, \mathbf{V}$.
- The matrix $\mathbf{V}^T$ identified using the SVD is exactly equal to the $\mathbf{W}$ matrix identified using PCA (or PPCA).
- This means that the SVD provides yet another way of identifying the optimal parameters for a linear autoencoder.