Bayesian Methods for Machine Learning

Lecture 4 - Factor analysis

Simon Leglaive

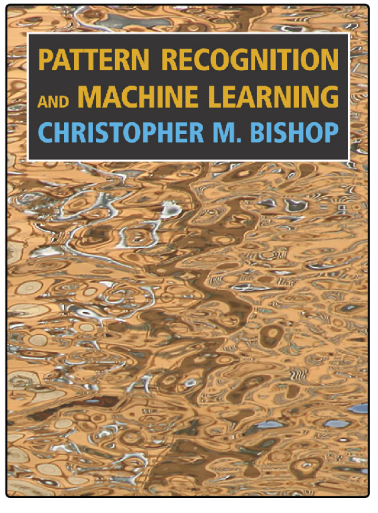
CentraleSupélec

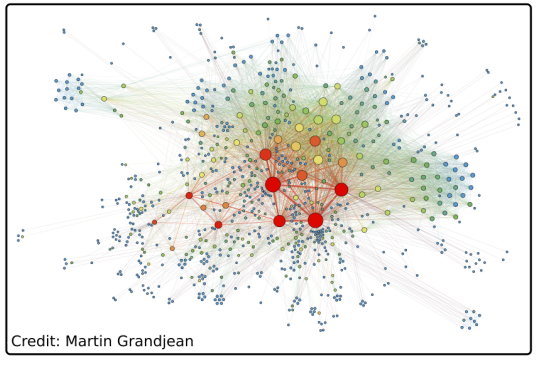
Introduction

Factor Analysis - an introduction

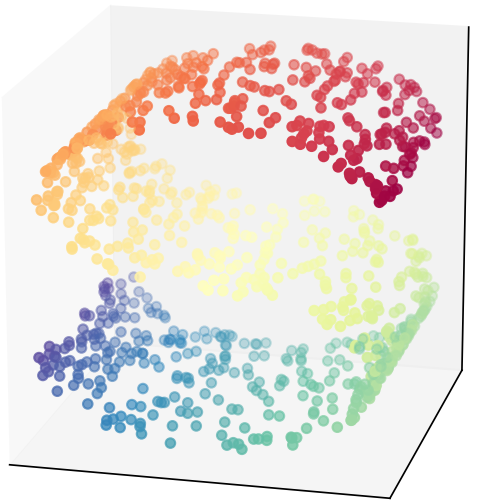
In many problems we have to manipulate high-dimensional data $\mathbf{x} \in \mathbb{R}^D$

- Image: $D \sim 10^6$ pixels.
- Audio: $D \sim 10^4$ samples per second.
- Text: $D \sim 10^6$ characters in a book.
- Social network: $D \sim 10^9$ nodes.

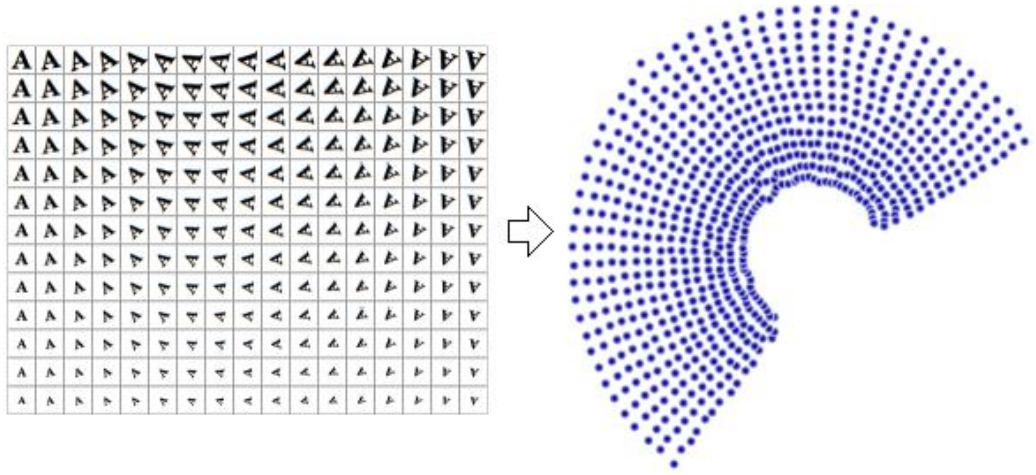




- A "natural" signal $\mathbf{x} \in \mathbb{R}^D$ such as an image or a sound exhibits some form of regularity, which prevents its D dimensions from varying freely.
- In other words, the number of degrees of freedom in $\mathbf{x} \in \mathbb{R}^D$ is much lesser than D.
- The data live in a low-dimensional manifold, embedded in the original high-dimensional space.



There exist latent factors from which the observed data were generated.



- Consider a dataset that contains images of a letter 'A', which has been scaled and rotated by varying amounts.
- Each image has 32x32 pixels, it can be represented as a vector of $D = 1024$ pixel values.
- The intrinsic dimensionality is two, because two variables (rotation and scale) were varied in order to produce the data.
- With a nonlinear dimensionality reduction technique, we can discard the correlated information (the letter 'A') and recover only the varying information (rotation and scale).

Credit: https://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction

Factor analysis

- The Factor Analysis (FA) model is a simple latent variable model where both the observed and latent variables are assumed to be continuous.
- FA is also a generative model of the observed variables.
- The observed data are assumed to lie on a lower-dimensional linear subspace of the original higher-dimensional space.
- FA is an unsupervised method that can be viewed as a generalized dimensionality reduction technique. As we will see, it is somehow related to principal component analysis (PCA).
- FA is commonly used in biology, social and behavioral sciences, marketing, recommendation systems, operational research, and finance.

Generative model and inference for factor analysis

Let $\mathbf{x} = [x_1, ..., x_D]^\top \in \mathbb{R}^D$ denote the observed variables and $\mathbf{z} = [z_1, ..., z_K]^\top \in \mathbb{R}^K$ the latent factors.

FA assumes that each observed variable x_i, $i \in \{1, ..., D\}$, correspond to:

- a linear combination of the $K \ll D$ latent factors z_k,
- with some unknown coefficients $w_{i,k}$,
- plus an arbitrary offset μ_i,
- plus a noise term ϵ_i,

that is:

$$x_i = \sum_{k=1}^{K} w_{i,k} z_k + \mu_i + \epsilon_i,$$

or in vector form:

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{w}_k z_k + \boldsymbol{\mu} + \boldsymbol{\epsilon},$$

where

- $\mathbf{w}_k = [w_{1,k}, ..., w_{D,k}]^\top \in \mathbb{R}^D$
- $\boldsymbol{\mu} = [\mu_1, ..., \mu_D]^\top \in \mathbb{R}^D$
- $\boldsymbol{\epsilon} = [\epsilon_1, ..., \epsilon_D]^\top \in \mathbb{R}^D$

You can also interpret $\mathbf{x}$ as a linear combination of K "basis vectors" $\mathbf{w}_k$.

or in matrix form:

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon},$$

where $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_K] \in \mathbb{R}^{D \times K}$ is sometimes called the "loading matrix".

So far, we do note have a generative model of the observed variables.

We need to define:

- a prior over the latent variables $p(\mathbf{z})$
- the likelihood $p(\mathbf{x}|\mathbf{z})$

- FA assumes a zero-mean unit-covariance Gaussian prior over the latent variables:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

- The likelihood is implicitely defined by modeling the noise term $\boldsymbol{\epsilon}$ as a Gaussian random vector with a zero mean and a diagonal covariance matrix:

$$p(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}),$$

  where $\boldsymbol{\Psi} = \mathrm{diag}(\Psi_1, ..., \Psi_D)$.

How come this noise model implicitely defines the likelihood?

Well, the Gaussian distribution has nice properties...

Multivariate Gaussian distribution

For a D-dimensional vector $\mathbf{x}$, the probability density function (pdf) of the multivariate Gaussian distribution is defined by:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}\sqrt{\det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right),$$

where

- $\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}]$ is the mean vector.
- $\boldsymbol{\Sigma} = \mathbb{E}[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^{\top}]$ is the covariance matrix.

For a D-dimensional vector $\mathbf{x}$, the probability density function (pdf) of the multivariate Gaussian distribution is defined by:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}\sqrt{\det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right),$$

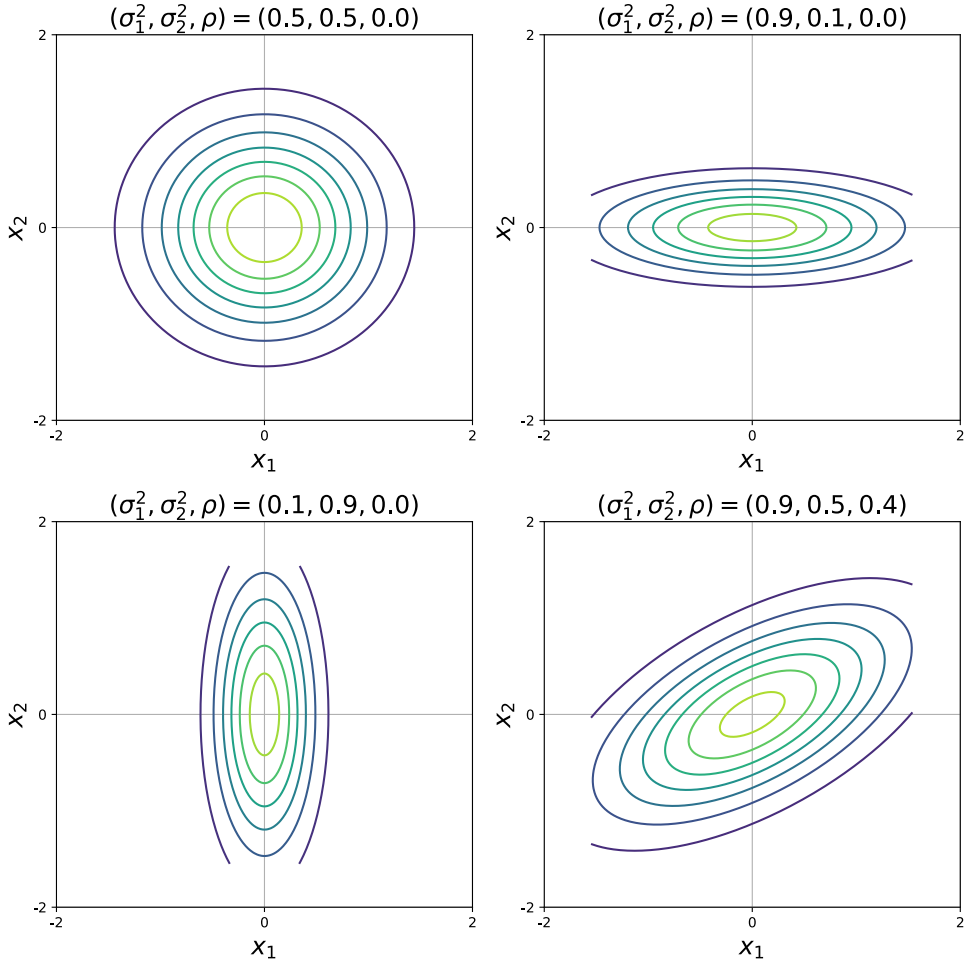
where

- $\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}]$ is the mean vector.
- $\boldsymbol{\Sigma} = \mathbb{E}[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^{\top}]$ is the covariance matrix.

Some properties:

- $\mathbb{E}[\mathbf{x}\mathbf{x}^{\top}] = \boldsymbol{\mu}\boldsymbol{\mu}^{\top} + \boldsymbol{\Sigma}$.
- $\boldsymbol{\Sigma}$ is symmetric, i.e. $\boldsymbol{\Sigma}^{\top} = \boldsymbol{\Sigma}$.
- $\boldsymbol{\Sigma}$ is positive-semidefinite, i.e. $\mathbf{a}^{\top}\boldsymbol{\Sigma}\mathbf{a} \geq 0$ for all $\mathbf{a} \in \mathbb{R}^D$.

2D Gaussian distribution



Consider a 2D Gaussian distribution with zero mean and covariance:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho \\ \rho & \sigma_2^2 \end{pmatrix}.$$

We plot the contours of constant probability density.

- Covariance $\propto$ identity: contours are concentric circles.
- Diagonal covariance: elliptical contours are aligned with the coordinate axes.

Joint Gaussianity

Let $\mathbf{x}_a \sim \mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})$ and $\mathbf{x}_b \sim \mathcal{N}(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{bb})$ be two Gaussian random vectors.

If $\mathbf{x}_a$ and $\mathbf{x}_b$ are independent, then they are jointly Gaussian, i.e. their joint distribution is also Gaussian:

$$\begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right),$$

where

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix},$$

and $\boldsymbol{\Sigma}_{ab} = \mathbb{E}\left[(\mathbf{x}_a - \boldsymbol{\mu}_a)(\mathbf{x}_b - \boldsymbol{\mu}_b)^\top\right] = \boldsymbol{\Sigma}_{ba}^\top = \mathbf{0}$.

The opposite is not true: a pair of jointly Gaussian random vectors may not be independent. They are is they are uncorrelated.

Affine transformation

Let $\mathbf{x}_a \sim \mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})$ and $\mathbf{x}_b$ be an affine transformation of $\mathbf{x}_a$:

$$\mathbf{x}_b = \mathbf{u} + \mathbf{V}\mathbf{x}_a.$$

The random vector $\mathbf{x}_b$ is also Gaussian:

$$\mathbf{x}_b \sim \mathcal{N}(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{bb}),$$

where

$$\boldsymbol{\mu}_b = \mathbf{u} + \mathbf{V}\boldsymbol{\mu}_a, \qquad \boldsymbol{\Sigma}_{bb} = \mathbf{V}\boldsymbol{\Sigma}_{aa}\mathbf{V}^\top.$$

Marginalization

If two random vectors are jointly Gaussian, then the marginal distribution of one vector is also Gaussian.

Let partition the Gaussian random vector $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ according to

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}, \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}.$$

The marginal distribution of $\mathbf{x}_a$ is given by:

$$\mathbf{x}_a \sim \mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa}).$$

The marginal distribution of $\mathbf{x}_b$ is given by:

$$\mathbf{x}_b \sim \mathcal{N}(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{bb}).$$

Conditioning

If two random vectors are jointly Gaussian, then the conditional distribution of one vector conditioned on the other is also Gaussian.

Let partition the Gaussian random vector $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ according to

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}, \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}.$$

The distribution of $\mathbf{x}_a$ conditioned on $\mathbf{x}_b$ is given by:

$$\mathbf{x}_a | \mathbf{x}_b \sim \mathcal{N}\left(\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b}\right),$$

where

$$\begin{aligned} \boldsymbol{\mu}_{a|b} &= \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b), \\ \boldsymbol{\Sigma}_{a|b} &= \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}. \end{aligned}$$

of course, the distribution of $\mathbf{x}_b$ conditioned on $\mathbf{x}_a$ is also given by:

$$\mathbf{x}_b|\mathbf{x}_a \sim \mathcal{N}\left(\boldsymbol{\mu}_{b|a}, \boldsymbol{\Sigma}_{b|a}\right),$$

where

$$\begin{aligned}\boldsymbol{\mu}_{b|a} &= \boldsymbol{\mu}_b + \boldsymbol{\Sigma}_{ba}\boldsymbol{\Sigma}_{aa}^{-1}(\mathbf{x}_a - \boldsymbol{\mu}_a),\\ \boldsymbol{\Sigma}_{b|a} &= \boldsymbol{\Sigma}_{bb} - \boldsymbol{\Sigma}_{ba}\boldsymbol{\Sigma}_{aa}^{-1}\boldsymbol{\Sigma}_{ab}.\end{aligned}$$

This property is very useful for computing posteriors when both observed and latent variables are jointly Gaussian!

Pause on notations

- $x \sim \mathcal{N}(\mu, \sigma^2)$ denotes that the random variable x follows a Gaussian distribution with mean μ and variance σ^2.
- $p(x) = \mathcal{N}(x; \mu, \sigma^2)$ dentotes the probability density function of the Gaussian distribution, i.e. a function of the random variable x parametrized by the mean μ and the variance σ^2.
- I abusively use the same symbol $\mathcal{N}$ to denote two different objects.
- Moreover, when it is not confusing (hopefully), I may simpy write $p(x) = \mathcal{N}(\mu, \sigma^2)$.

(back to) Generative model and inference

We recall the model:

$$\begin{aligned}
&\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \\
&\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\
&\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}), \qquad\qquad \boldsymbol{\Psi} = \operatorname{diag}(\Psi_1, \ldots, \Psi_D).
\end{aligned}$$

We further assume that the two Gaussian random vectors $\mathbf{z}$ and $\boldsymbol{\epsilon}$ are independent.

It follows that $\mathbf{z}$ and $\boldsymbol{\epsilon}$ are jointly Gaussian:

$$\begin{pmatrix} \boldsymbol{\epsilon} \\ \mathbf{z} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Psi} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \right).$$

The random vector $[\mathbf{x}^\top, \mathbf{z}^\top]^\top$, as an affine transformation of the Gaussian vector $[\boldsymbol{\epsilon}^\top, \mathbf{z}^\top]^\top$, is also a Gaussian random vector:

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mu} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{I} & \mathbf{W} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{\epsilon} \\ \mathbf{z} \end{pmatrix}.$$

We have:

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where using the previous formulas

$$\begin{aligned} \boldsymbol{\mu} &= \begin{pmatrix} \boldsymbol{\mu} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{I} & \mathbf{W} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{\mu} \\ \mathbf{0} \end{pmatrix} \end{aligned} \qquad \begin{aligned} \boldsymbol{\Sigma} &= \begin{pmatrix} \mathbf{I} & \mathbf{W} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Psi} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{W} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}^\top \\ &= \begin{pmatrix} \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi} & \mathbf{W} \\ \mathbf{W}^\top & \mathbf{I} \end{pmatrix} \end{aligned}$$

Once we know that $\begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ by linearity of the Gaussian distribution, instead of directly applying the affine transform formulas we could have partitioned the parameters as

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_z \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xz} \\ \boldsymbol{\Sigma}_{zx} & \boldsymbol{\Sigma}_{zz} \end{pmatrix},$$

and compute the individual factors as follows:

- $\boldsymbol{\mu}_z = \mathbb{E}[\mathbf{z}] = \mathbf{0}$
- $\boldsymbol{\mu}_x = \mathbb{E}[\mathbf{x}] = \mathbb{E}[\mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}] = \boldsymbol{\mu}$
- $\boldsymbol{\Sigma}_{zz} = \mathbb{E}[(\mathbf{z} - \boldsymbol{\mu}_z)(\mathbf{z} - \boldsymbol{\mu}_z)^\top] = \mathbf{I}$
- $\boldsymbol{\Sigma}_{xz} = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu}_x)(\mathbf{z} - \boldsymbol{\mu}_z)^\top] = \mathbb{E}[(\mathbf{W}\mathbf{z} + \boldsymbol{\epsilon})\mathbf{z}^\top] = \mathbf{W}$
- $\boldsymbol{\Sigma}_{zx} = \boldsymbol{\Sigma}_{xz}^\top = \mathbf{W}^\top$
- $\boldsymbol{\Sigma}_{xx} = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu}_x)(\mathbf{x} - \boldsymbol{\mu}_x)^\top] = \mathbb{E}[(\mathbf{W}\mathbf{z} + \boldsymbol{\epsilon})(\mathbf{W}\mathbf{z} + \boldsymbol{\epsilon})^\top] = \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi}$

Using the conditioning formulas, we finally obtain the likelihood:

$$\mathbf{x}|\mathbf{z} \sim \mathcal{N}\left(\boldsymbol{\mu}_{x|z}, \boldsymbol{\Sigma}_{x|z}\right),$$

- $\boldsymbol{\mu}_{x|z} = \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xz}\boldsymbol{\Sigma}_{zz}^{-1}(\mathbf{z} - \boldsymbol{\mu}_z) = \boldsymbol{\mu} + \mathbf{W}\mathbf{z}$
- $\boldsymbol{\Sigma}_{x|z} = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xz}\boldsymbol{\Sigma}_{zz}^{-1}\boldsymbol{\Sigma}_{zx} = \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi} - \mathbf{W}\mathbf{W}^\top = \boldsymbol{\Psi}$

and for free we also have the posterior (the Bayesian holy grail):

$$\mathbf{z}|\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{\mu}_{z|x}, \boldsymbol{\Sigma}_{z|x}\right),$$

- $\boldsymbol{\mu}_{z|x} = \boldsymbol{\mu}_z + \boldsymbol{\Sigma}_{zx}\boldsymbol{\Sigma}_{xx}^{-1}(\mathbf{x} - \boldsymbol{\mu}_x) = \mathbf{W}^\top(\mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi})^{-1}(\mathbf{x} - \boldsymbol{\mu})$
- $\boldsymbol{\Sigma}_{z|x} = \boldsymbol{\Sigma}_{zz} - \boldsymbol{\Sigma}_{zx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{zx} = \mathbf{I} - \mathbf{W}^\top(\mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi})^{-1}\mathbf{W}$

Let us summarize what we know about FA so far

FA generative model

- prior over the latent variables $\mathbf{z} \in \mathbb{R}^K$:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$$

- likelihood for the observed variables $\mathbf{x} \in \mathbb{R}^D$:

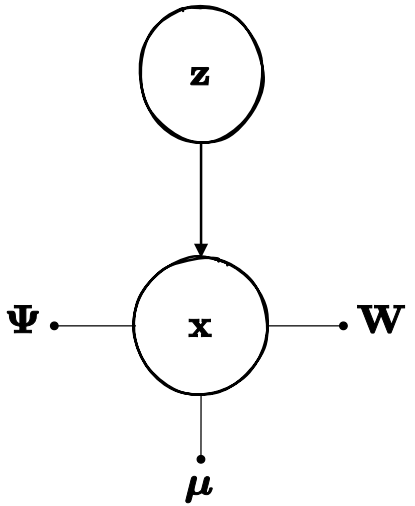
$$p(\mathbf{x}|\mathbf{z}; \theta) = \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu} + \mathbf{W}\mathbf{z}, \boldsymbol{\Psi}\right)$$

- marginal likelihood:

$$p(\mathbf{x}; \theta) = \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi}\right)$$

- model parameters:

$$\theta = \left\{\boldsymbol{\mu} \in \mathbb{R}^D, \mathbf{W} \in \mathbb{R}^{D \times K}, \boldsymbol{\Psi} = \mathrm{diag}(\Psi_1, ..., \Psi_D) \in \mathbb{R}^{D \times D}\right\}$$



The observation model is equivalent to

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$$

where the noise term $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$.

FA inference

When performing factor analysis, our objective is to infer the latent variables $\mathbf{z}$ given the observations $\mathbf{x}$.

The posterior distribution of the latent variables is given by:

$$p(\mathbf{z}|\mathbf{x};\theta) = \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{z|x}, \boldsymbol{\Sigma}_{z|x}\right)$$

- $\boldsymbol{\mu}_{z|x} = \mathbf{W}^\top(\mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi})^{-1}(\mathbf{x} - \boldsymbol{\mu})$
- $\boldsymbol{\Sigma}_{z|x} = \mathbf{I} - \mathbf{W}^\top(\mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi})^{-1}\mathbf{W}$

This posterior depends on the unknown model parameters.

It requires inverting a $D \times D$ matrix, which is computationally demanding in high dimension.

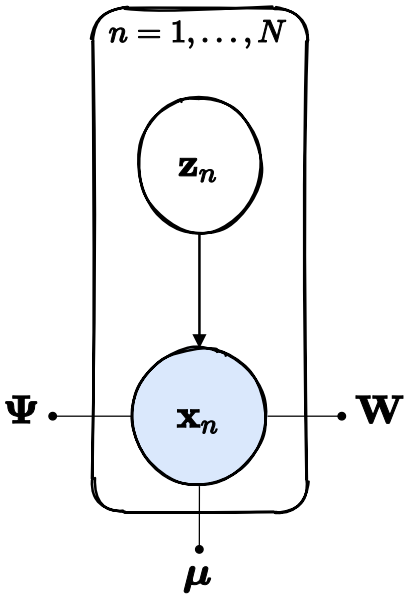
We can use Woodbury matrix inversion identity to rewrite the posterior parameters in terms of the inverse of a $K \times K$ matrix.

Parameters estimation

For parameters estimation, we usually have access to a dataset $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$ of N i.i.d realizations from the previous FA model.

Similarly, we define the set $\mathcal{Z} = \{\mathbf{z}_n\}_{n=1}^N$ of latent vectors associated with the observed data $\mathcal{X}$.

The resulting graphical model along with the associated joint distribution are given by:



$$p(\mathcal{X}, \mathcal{Z}; \theta) = \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n; \theta) p(\mathbf{z}_n),$$

where

- $p(\mathbf{z}_n) = \mathcal{N}(\mathbf{z}_n; \mathbf{0}, \mathbf{I})$,
- $p(\mathbf{x}_n|\mathbf{z}_n; \theta) = \mathcal{N}\left(\mathbf{x}_n; \boldsymbol{\mu} + \mathbf{W}\mathbf{z}_n, \boldsymbol{\Psi}\right)$.

Ideally, we would like to estimate the parameters θ by maximizing the log-marginal likelihood for the dataset $\mathcal{X}$:

$$
\begin{aligned}
\ln p(\mathcal{X};\theta) &= \ln \prod_{n=1}^{N} p(\mathbf{x}_n;\theta) \\
&= \sum_{n=1}^{N} \ln \mathcal{N}\left(\mathbf{x}_n;\boldsymbol{\mu},\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi}\right) \\
&= \sum_{n=1}^{N} \ln \frac{1}{(2\pi)^{D/2}\sqrt{\det(\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi})}} \exp\left(-\frac{1}{2}(\mathbf{x}_n-\boldsymbol{\mu})^\top(\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi})^{-1}(\mathbf{x}_n-\boldsymbol{\mu})\right) \\
&= -\frac{N}{2}\ln\det(\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi}) - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n-\boldsymbol{\mu})^\top(\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi})^{-1}(\mathbf{x}_n-\boldsymbol{\mu}) + cst(\theta)
\end{aligned}
$$

The log-marginal likelihood is a quadratic function in $\boldsymbol{\mu}$.

Canceling its partial derivative with respect to $\boldsymbol{\mu}$ gives the empirical mean $\boldsymbol{\mu} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n$.

This solution represents the unique maximum, as could be confirmed by computing second order derivative.

We could have centered the data by substracting this empirical mean, such that the "new" mean of the preprocessed data would be zero.

In the following, without loss of generality, we will assume that the data are centered, i.e. $\boldsymbol{\mu} = \mathbf{0}$.

The log-marginal likelihood for centered data is:

$$
\begin{aligned}
\ln p(\mathcal{X};\theta) &= \ln \prod_{n=1}^{N} p(\mathbf{x}_n;\theta) \\
&= \sum_{n=1}^{N} \ln \mathcal{N}\left(\mathbf{x}_n;\mathbf{0},\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi}\right) \\
&= \sum_{n=1}^{N} \ln \frac{1}{(2\pi)^{D/2}\sqrt{\det(\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi})}} \exp\left(-\frac{1}{2}{\mathbf{x}_n}^\top(\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi})^{-1}\mathbf{x}_n\right) \\
&= -\frac{N}{2}\ln\det(\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi}) - \frac{1}{2}\sum_{n=1}^{N}{\mathbf{x}_n}^\top(\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi})^{-1}\mathbf{x}_n + cst(\theta) \\
&= -\frac{N}{2}\ln\det(\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi}) - \frac{N}{2}\mathrm{trace}\left[(\mathbf{W}\mathbf{W}^\top+\boldsymbol{\Psi})^{-1}\hat{\mathbf{R}}_{xx}\right] + cst(\theta),
\end{aligned}
$$

where $\hat{\mathbf{R}}_{xx} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top$ is the empirical covariance matrix.

If estimating the mean is quite straightforward, estimating the other model parameters

$$\mathbf{W} \in \mathbb{R}^{D \times K}, \qquad \boldsymbol{\Psi} = \mathrm{diag}(\Psi_1, ..., \Psi_D) \in \mathbb{R}^{D \times D},$$

by maximizing the log-marginal likelihood does not have an exact closed-form solution...

If estimating the mean is quite straightforward, estimating the other model parameters

$$\mathbf{W} \in \mathbb{R}^{D \times K}, \qquad \boldsymbol{\Psi} = \mathrm{diag}(\Psi_1, ..., \Psi_D) \in \mathbb{R}^{D \times D},$$

by maximizing the log-marginal likelihood does not have an exact closed-form solution...

... except with additional assumptions.

Probabilistic PCA (principal component analysis)

Let us assume that $\boldsymbol{\Psi} = \text{diag}(\Psi_1, ..., \Psi_D) = \sigma^2 \mathbf{I}$.

The log-marginal likelihood becomes:

$$\ln p(\mathcal{X}; \theta) = -\frac{N}{2} \left(\ln \det(\mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I}) + \text{trace}\left[(\mathbf{W}\mathbf{W}^\top + \sigma^2 \mathbf{I})^{-1} \hat{\mathbf{R}}_{xx} \right] \right) + cst(\theta).$$

Maximization with respect to $\mathbf{W}$ and σ^2 now has an exact closed-form solution (cf. Bishop, PRML, section 12.2.1, p. 574).

Christopher M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.

Probablistic PCA differs from FA only in the structure of the noise covariance matrix $\boldsymbol{\Psi}$:

- FA assumes a heteroscedastic noise:

  The covariance matrix is diagonal, i.e. the noise variance is different for each dimension.

- Probabilistic PCA assumes a homoscedastic noise:

  The covariance matrix is proportional to the identity, i.e. the noise variance is the same for each dimension.

So, except is we make additional assumptions, we cannot directly maximize the log-marginal likelihood to estimate the model parameters $\mathbf{W} \in \mathbb{R}^{D \times K}$ and $\boldsymbol{\Psi} = \text{diag}(\Psi_1, ..., \Psi_D) \in \mathbb{R}^{D \times D}$.

We have observed variables, we have latent variables, ...

So, except is we make additional assumptions, we cannot directly maximize the log-marginal likelihood to estimate the model parameters $\mathbf{W} \in \mathbb{R}^{D \times K}$ and $\boldsymbol{\Psi} = \text{diag}(\Psi_1, ..., \Psi_D) \in \mathbb{R}^{D \times D}$.

We have observed variables, we have latent variables, ...

let's derive an EM algorithm!

EM recipe reminder

Given an initialization θ_0 of the model parameters, iterate for $t = 0 : T - 1$:

- E-Step: $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathcal{Z}|\mathcal{X};\theta_t)}[\ln p(\mathcal{X}, \mathcal{Z}; \theta)]$;
- M-Step: $\theta_{t+1} = \arg\max_{\theta} Q(\theta, \theta_t)$.

Exercise

Derive the EM algorithm for the FA model.