CS 5135/6035 Learning Probabilistic Models

Lecture 12: Factor Analysis

Gowtham Atluri

October 12, 2018

Reading Material

- Andrew Ng, Factor Analysis
  http://cs229.stanford.edu/notes/cs229-notes9.pdf
- Cosma Shalizi, Factor Analysis
  https://www.stat.cmu.edu/~cshalizi/350/lectures/12/lecture-12.pdf

Factor Analysis

- Problem definition
- Benefits
- Issues

Latent Variables

Mixture Models

$$z_i \sim Multinomial(\lambda_1, \lambda_2, \dots, \lambda_k)$$
$$x_i | z_i \sim \mathcal{N}(\mu_{z_i}, \sigma^2_{z_i})$$

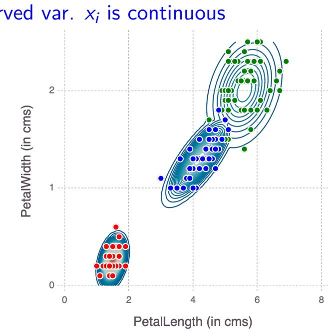
Note: Latent var. z_i is discrete; Observed var. x_i is continuous

Parameter estimation using MLE

EM Algorithm

E-step: Estimate $p(z_i|x_i)$, assuming μ_i and σ_i^2 $(i \in \{1, \dots, k\})$ are avaialble

M-step: Estimate μ_i and σ_i^2 $(i \in \{1, \dots, k\})$, assuming $p(z_i|x_i)$ is available



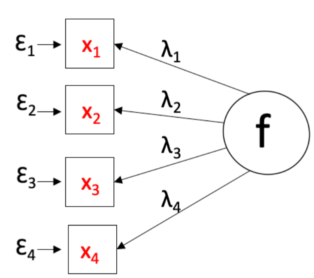
Factor Analysis

- Hidden variable(s) is (are) continuous
- First studied by Charles Spearman in 1904
  - to explain hidden structure of human intelligence
- Spearman observed that schoolchildren's grades in different subjects were correlated with each other
- He explained:
  - the reason grades in math, English, history, etc. are correlated with each other, is they are all correlated with something else
  - a **common factor**, which he named "general intelligence"

ID	math	English	history	Intelligence
1	77.4	82.6	81.6	?
2	58.5	61.3	52.2	?
.	.	.	.	?
100	76.7	72.4	72.2	?

Spearman's factor analysis model

$$x_i = \lambda_i f + \epsilon_i$$



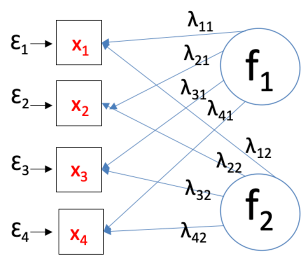
- x_i are the observed variables
  - e.g., x_1, x_2, and x_3 are exam scores obtained by a student in math, English and history.
- f is the underlying commmon factor
  - e.g., student's intelligence
- λ_i are the factor loadings
  - e.g., how much is the contribution of intelligence to exam score
- ϵ_i are unique factors or residuals or random noise terms
  - e.g., how much result differs from student's general ability

When there are multiple factors...

$$x_i = \lambda_{i1} f_1 + \lambda_{i2} f_2 + \ldots + \lambda_{ik} f_k + \epsilon_i$$

Example: Source-apportionment of fine particulate matter (PM $\leq 2.5\mu m$)
Factors include:

- Vehicle emissions
- Coal combustion
- Biomass combustion
- residential heaters
- industrial processes



For multiple samples

$$x_{i,1} = \lambda_{11} f_{i,1} + \lambda_{12} f_{i,2} + \epsilon_{i,1}$$
$$x_{i,2} = \lambda_{21} f_{i,1} + \lambda_{22} f_{i,2} + \epsilon_{i,2}$$
$$x_{i,3} = \lambda_{31} f_{i,1} + \lambda_{32} f_{i,2} + \epsilon_{i,3}$$
$$x_{i,4} = \lambda_{41} f_{i,1} + \lambda_{42} f_{i,2} + \epsilon_{i,4}$$

- $x_{i,j}$ is the j^{th} observed variable in the i^{th} sample
  - Total number of variables are denoted by d
- $f_{i,j}$ is the j^{th} factor for sample i
  - Total number of factors are denoted by k
- $\lambda_{i,j}$ is the loading of the j^{th} factor on the i^{th} observed variable
- $\epsilon_{i,j}$ is the unique factor contributing to the j^{th} observed variable in the i^{th} sample

For multiple samples

$$x_{i,1} = \lambda_{11} f_{i,1} + \lambda_{12} f_{i,2} + \epsilon_{i,1}$$
$$x_{i,2} = \lambda_{21} f_{i,1} + \lambda_{22} f_{i,2} + \epsilon_{i,2}$$
$$x_{i,3} = \lambda_{31} f_{i,1} + \lambda_{32} f_{i,2} + \epsilon_{i,3}$$
$$x_{i,4} = \lambda_{41} f_{i,1} + \lambda_{42} f_{i,2} + \epsilon_{i,4}$$

Matrix notation

$$\begin{bmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \\ x_{i,4} \end{bmatrix} = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \\ \lambda_{31} & \lambda_{32} \\ \lambda_{41} & \lambda_{42} \end{bmatrix} \begin{bmatrix} f_{i,1} \\ f_{i,2} \end{bmatrix} + \begin{bmatrix} \epsilon_{i,1} \\ \epsilon_{i,2} \\ \epsilon_{i,3} \\ \epsilon_{i,4} \end{bmatrix}$$

$$\boldsymbol{x} = \boldsymbol{\Lambda} \boldsymbol{f} + \epsilon$$

Advantages: Applications

- Matrix Factorization
  - [Genes × Conditions] = [Genes × Pathways] * [Pathways × Conditions]
  - [Users × Movies] = [Users × Genres] * [Genres × Movies]
- Dimensionality reduction
  - Work with $\boldsymbol{f}$ (k-dim) instead of $\boldsymbol{x}$ (d-dim) ($k < d$)

For all the samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & x_{n4} \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \\ f_{31} & f_{32} \\ \vdots & \vdots \\ f_{n1} & f_{n2} \end{bmatrix} \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \\ \lambda_{31} & \lambda_{32} \\ \lambda_{41} & \lambda_{42} \end{bmatrix}^T + \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} & \epsilon_{14} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} & \epsilon_{24} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} & \epsilon_{34} \\ \vdots & \vdots & \vdots & \vdots \\ \epsilon_{n1} & \epsilon_{n2} & \epsilon_{n3} & \epsilon_{n4} \end{bmatrix}$$

Problem of indeterminacy, Imposing Constraints

$$\boldsymbol{x} = \boldsymbol{\Lambda} \boldsymbol{f} + \epsilon$$

- Problem of indeterminacy
  - No values on the R.H.S. are known
  - Many solutions are possible
- We imppose constraints to reduce indeterminacy
  - Unique factors all have mean zero ($E(\epsilon) = 0$)
  - Common factors all have mean zero ($E(\boldsymbol{f}) = 0$)
  - Let $E(\boldsymbol{x}) = \mu$

$$\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Lambda} \boldsymbol{f} + \epsilon$$

More Constraints

$$\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Lambda} \boldsymbol{f} + \boldsymbol{\epsilon}$$

- More constraints
  - Common factors are standardized uncorrelated random variables $E(\boldsymbol{f}\boldsymbol{f}^T) = \boldsymbol{I}$
    - their variance 1, covariance 0.
  - Unique factors are uncorrelated and heteroscedastic $E(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) = \Psi$

$$\Psi = \begin{bmatrix} \psi_{11} & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & \psi_{rr} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \psi_{dd} \end{bmatrix}$$

  - Unique factors are uncorrelated with the common factors

$$E(\boldsymbol{f}\boldsymbol{\epsilon}^T) = 0 \text{ and } E(\boldsymbol{\epsilon}\boldsymbol{f}^T) = 0)$$

Problem of Identifiability

Identifiability

A statistical model is said to be identifiable if a unique solution exists to the estimation problem.

- The factor model $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Lambda f} + \epsilon$ is not identifiable
  - there is an infinite number of different matrices $\boldsymbol{\Lambda}$ that can generate the same $\boldsymbol{x}$ values
- The model can be equivalently rewritten as

$$\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Lambda f} + \epsilon = \boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Lambda I f} + \epsilon = \boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Lambda G G}^T + \epsilon$$

- If we set, $\boldsymbol{\Lambda}* = \boldsymbol{\Lambda G}$ and $\boldsymbol{f}^* = \boldsymbol{G}^T \boldsymbol{f}$, the factor model becomes $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Lambda}^* \boldsymbol{f}^* + \epsilon$
- The two factor models are completely equivalent and indistinguishable
  - They have the same properties
  - $E[\boldsymbol{f}^*] = 0$
  - $E[\boldsymbol{f}^* \boldsymbol{f}^{*T}] = \boldsymbol{I}$
  - $E[\boldsymbol{f}^* \epsilon^T] = 0$

Probabilistic Modelling

- Formulation
- Geometric interpretation
- Joint distribution
- Max. Likelihood Estimation
  - EM approach

Probabilistic Modelling

Generative model for factor analysis

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$
$$\boldsymbol{x} | \boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu} + \boldsymbol{\Lambda f}, \boldsymbol{\Psi})$$

Alternatively,

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$
$$\epsilon \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Psi})$$
$$\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Lambda f} + \epsilon$$

Parameters of this model are:

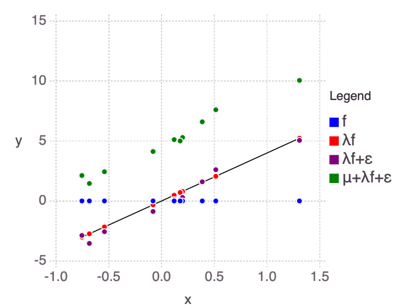
- Vector $\boldsymbol{\mu} \in \mathbb{R}^d$
- Matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{d \times k}$
  - usually $k < d$
- Diagonal matrix $\boldsymbol{\Psi} \in \mathbb{R}^{d \times d}$

This assumption that data follows a Gaussian distr. need not suit all cases. - Can be a disadvantage.

Geometric Interpretation in 2D

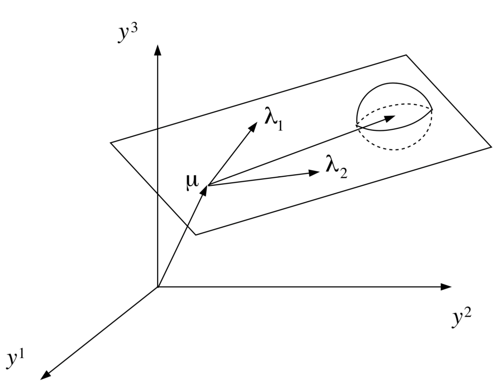
Data generating process:

$$f \sim \mathcal{N}(0, 1)$$
$$x | f \sim \mathcal{N}(\mu + \lambda f + \epsilon, \sigma)$$



- Data points lie in a close to linear subspace.
  - Observed variables lie in 2D
  - Factors in 1D

Geometric Interpretation in 3D



- Data points lie in a close to linear subspace.

Joint Distribution

Our Model

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$
$$\epsilon \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Psi})$$
$$\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Lambda f} + \epsilon$$

RVs $\boldsymbol{x}$ and $\boldsymbol{f}$ have a joint Gaussian distribution

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{x} \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}_{f,x}, \boldsymbol{\Sigma})$$

We will now determine $\boldsymbol{\mu}_{f,x}, \boldsymbol{\Sigma}$

$$\begin{aligned} E[\boldsymbol{x}] &= E[\boldsymbol{\mu} + \boldsymbol{\Lambda f} + \epsilon] \\ &= \boldsymbol{\mu} + \boldsymbol{\Lambda} E[\boldsymbol{f}] + E[\epsilon] \\ &= \boldsymbol{\mu} \text{ (as } E[\boldsymbol{f}] = 0 \text{ and } E[\epsilon] = 0 \text{ from constraints)} \end{aligned}$$

From this, we have

$$\boldsymbol{\mu}_{f,x} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\mu} \end{bmatrix}$$

Joint Distribution

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{f,f} & \boldsymbol{\Sigma}_{f,x} \\ \boldsymbol{\Sigma}_{x,f} & \boldsymbol{\Sigma}_{x,x} \end{bmatrix} = \begin{bmatrix} E[(\boldsymbol{f}-E[\boldsymbol{f}])(\boldsymbol{f}-E[\boldsymbol{f}])^T] & E[(\boldsymbol{f}-E[\boldsymbol{f}])(\boldsymbol{x}-E[\boldsymbol{x}])^T] \\ E[(\boldsymbol{x}-E[\boldsymbol{x}])(\boldsymbol{f}-E[\boldsymbol{f}])^T] & E[(\boldsymbol{x}-E[\boldsymbol{x}])(\boldsymbol{x}-E[\boldsymbol{x}])^T] \end{bmatrix}$$

Since $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, $\boldsymbol{\Sigma}_{f,f} = Cov(\boldsymbol{f}) = \boldsymbol{I}$.

$$\begin{aligned} E[(\boldsymbol{f}-E[\boldsymbol{f}])(\boldsymbol{x}-E[\boldsymbol{x}])^T] &= E[\boldsymbol{f}(\boldsymbol{\mu}+\boldsymbol{\Lambda f}+\boldsymbol{\epsilon}-\boldsymbol{\mu})^T] \\ &= E[\boldsymbol{ff}^T]\boldsymbol{\Lambda}^T + E[\boldsymbol{f\epsilon}^T] \\ &= \boldsymbol{\Lambda}^T \end{aligned}$$

$$\begin{aligned} E[(\boldsymbol{x}-E[\boldsymbol{x}])(\boldsymbol{x}-E[\boldsymbol{x}])^T] &= E[(\boldsymbol{\mu}+\boldsymbol{\Lambda f}+\boldsymbol{\epsilon}-\boldsymbol{\mu})^T(\boldsymbol{\mu}+\boldsymbol{\Lambda f}+\boldsymbol{\epsilon}-\boldsymbol{\mu})^T] \\ &= E[\boldsymbol{\Lambda ff}^T\boldsymbol{\Lambda}^T + \boldsymbol{\epsilon f}^T\boldsymbol{\Lambda}^T + \boldsymbol{\Lambda f\epsilon}^T + \boldsymbol{\epsilon\epsilon}^T] \\ &= \boldsymbol{\Lambda}E[\boldsymbol{ff}^T]\boldsymbol{\Lambda}^T + E[\boldsymbol{\epsilon\epsilon}^T] \\ &= \boldsymbol{\Lambda\Lambda}^T + \boldsymbol{\Psi} \end{aligned}$$

Joint Distribution

- Putting it all together...

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{x} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\mu} \end{bmatrix}, \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Lambda}^T \\ \boldsymbol{\Lambda} & \boldsymbol{\Lambda\Lambda}^T + \boldsymbol{\Psi} \end{bmatrix} \right)$$

- From this, marginal distribution of $\boldsymbol{x}$ is $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda\Lambda}^T + \boldsymbol{\Psi})$
- Given a dataset of samples $\boldsymbol{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$, we can write the log likelihood of the parameters as:

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}) = \log \prod_{i=1}^{n} \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Lambda\Lambda}^T + \boldsymbol{\Psi}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu})^T(\boldsymbol{\Lambda\Lambda}^T + \boldsymbol{\Psi})^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})\right)$$

- To perform MLE, we need to maximize this quantity w.r.t. the parameters.
  - No closed form solution exists
  - We will use Expectation-Maximization (EM) algorithm

EM algorithm

- For any choice of $q(\boldsymbol{Z})$, the following decomposition holds

$$\log p(\boldsymbol{X}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q||p)$$

where we define

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{\boldsymbol{z}} q(\boldsymbol{Z}) \log \left\{ \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})}{q(\boldsymbol{Z})} \right\}$$

$$KL(q||p) = -\sum_{\boldsymbol{z}} q(\boldsymbol{Z}) \log \left\{ \frac{p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta})}{q(\boldsymbol{Z})} \right\}$$

- As KL$(p||q) \geq 0$, $\mathcal{L}(q, \boldsymbol{\theta}) \leq \log p(\boldsymbol{X}|\boldsymbol{\theta})$.
  - $\mathcal{L}(q, \boldsymbol{\theta})$ is the lower bound on $\log p(\boldsymbol{X}|\boldsymbol{\theta})$.
- In E-Step: Lowed bound $\mathcal{L}(q, \boldsymbol{\theta})$ is maximized w.r.t. $q(\boldsymbol{Z})$, fixing $\boldsymbol{\theta}^{old}$
  - Essentially substituting $q(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta})$
- In M-Step: $\mathcal{L}(q, \boldsymbol{\theta})$ is maximized w.r.t. $\boldsymbol{\theta}$ to give some new value $\boldsymbol{\theta}^{new}$

EM Algorithm: E-Step

In the E-step, we need to compute $Q_i(\boldsymbol{f}_i) = p(\boldsymbol{f}_i|\boldsymbol{x}_i; \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi})$

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{x} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\mu} \end{bmatrix}, \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Lambda}^T \\ \boldsymbol{\Lambda} & \boldsymbol{\Lambda\Lambda}^T + \boldsymbol{\Psi} \end{bmatrix} \right)$$

We know for MV Gaussians...

$$\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix} \right)$$

the conditional $p(\boldsymbol{a}|\boldsymbol{b}) = \mathcal{N}(\boldsymbol{a}|\boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\boldsymbol{b} - \boldsymbol{\mu}_b), \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba})$

Using this property, we can write $p(\boldsymbol{f}_i|\boldsymbol{x}_i; \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{f}_i|\boldsymbol{x}_i}, \boldsymbol{\Sigma}_{\boldsymbol{f}_i|\boldsymbol{x}_i})$

$$\begin{aligned} \text{where, } \boldsymbol{\mu}_{\boldsymbol{f}_i|\boldsymbol{x}_i} &= \boldsymbol{\Lambda}^T(\boldsymbol{\Lambda\Lambda}^T + \boldsymbol{\Psi})^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}) \\ \boldsymbol{\Sigma}_{\boldsymbol{f}_i|\boldsymbol{x}_i} &= \boldsymbol{I} - \boldsymbol{\Lambda}^T(\boldsymbol{\Lambda\Lambda}^T + \boldsymbol{\Psi})^{-1}\boldsymbol{\Lambda} \end{aligned}$$

EM Algorithm: M-Step

We need to maximize the following function w.r.t. parameters $\boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}$

$$\sum_{i=1}^{n} \int_{\boldsymbol{f}_i} q(\boldsymbol{f}_i) \log \frac{p(\boldsymbol{x}_i, \boldsymbol{f}_i; \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi})}{q(\boldsymbol{f}_i)} d\boldsymbol{f}_i$$

$$= \sum_{i=1}^{n} \int_{\boldsymbol{f}_i} q(\boldsymbol{f}_i)[\log p(\boldsymbol{x}_i|\boldsymbol{f}_i; \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}) + \log p(\boldsymbol{f}_i) - \log q(\boldsymbol{f}_i)] d\boldsymbol{f}_i$$

$$= \sum_{i=1}^{n} E_{\boldsymbol{f}_i \sim q}[\log p(\boldsymbol{x}_i|\boldsymbol{f}_i; \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}) + \log p(\boldsymbol{f}_i) - \log q(\boldsymbol{f}_i)]$$

First, computing gradient w.r.t. $\boldsymbol{\Lambda}$ (we only need the first log term)

$$\sum_{i=1}^{n} E[\log p(\boldsymbol{x}_i|\boldsymbol{f}_i; \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi})]$$

$$= \sum_{i=1}^{n} E\Big[\log \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Psi}|^{1/2}} \exp\Big(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{\Lambda f}_i)^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{\Lambda f}_i)\Big)\Big]$$

$$= \sum_{i=1}^{n} E\Big[-\frac{1}{2}\log|\boldsymbol{\Psi}| - \frac{n}{2}\log(2\pi) - \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{\Lambda f}_i)^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{\Lambda f}_i)\Big]$$

EM Algorithm: M-Step

- Some results from multivariate calculus
  - $tra = a$ for $a \in \mathbb{R}$; $trAB = trBA$; $\nabla_A trABA^TC = CAB + C^TAB$

$$\sum_{i=1}^{n} E[\log p(\boldsymbol{x}_i|\boldsymbol{f}_i; \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi})]$$

$$= \sum_{i=1}^{n} E\Big[-\frac{1}{2}\log|\boldsymbol{\Psi}| - \frac{n}{2}\log(2\pi) - \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{\Lambda f}_i)^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{\Lambda f}_i)\Big]$$

$$\begin{aligned} \nabla_{\boldsymbol{\Lambda}} \sum_{i=1}^{n} -E\Big[\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{\Lambda f}_i)^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu} - \boldsymbol{\Lambda f}_i)\Big] \\ = \sum_{i=1}^{n} \nabla_{\boldsymbol{\Lambda}} E\Big[tr\frac{1}{2}\boldsymbol{f}_i^T\boldsymbol{\Lambda}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda f}_i + tr\boldsymbol{f}_i^T\boldsymbol{\Lambda}^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})\Big] \\ = \sum_{i=1}^{n} \nabla_{\boldsymbol{\Lambda}} E\Big[tr\frac{1}{2}\boldsymbol{\Lambda}^T\boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda f}_i\boldsymbol{f}_i^T + tr\boldsymbol{\Lambda}^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})\boldsymbol{f}_i^T\Big] \\ = \sum_{i=1}^{n} E\Big[-\boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda f}_i\boldsymbol{f}_i^T + \boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})\boldsymbol{f}_i^T\Big] \end{aligned}$$

EM Algorithm: M-Step

Gradient w.r.t. $\mathbf{\Lambda}$ is

$$\sum_{i=1}^{n} E\Big[-\mathbf{\Psi}^{-1}\mathbf{\Lambda}\boldsymbol{f}_i\boldsymbol{f}_i^T + \mathbf{\Psi}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})\boldsymbol{f}_i^T\Big]$$

Setting this to 0 and simplifying,

$$\sum_{i=1}^{n} \mathbf{\Lambda} E_{f_i\sim q}\Big[\boldsymbol{f}_i\boldsymbol{f}_i^T\Big] = \sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu})E_{f_i\sim q}\Big[\boldsymbol{f}_i^T\Big]$$

Solving for $\mathbf{\Lambda}$, we get

$$\mathbf{\Lambda} = \Big(\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu})E_{f_i\sim q}\Big[\boldsymbol{f}_i^T\Big]\Big)\Big(\sum_{i=1}^{n}E_{f_i\sim q}\Big[\boldsymbol{f}_i\boldsymbol{f}_i^T\Big]\Big)^{-1}$$

EM Algorithm: M-Step

$$\mathbf{\Lambda} = \Big(\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu})E_{f_i\sim q}\Big[\boldsymbol{f}_i^T\Big]\Big)\Big(\sum_{i=1}^{n}E_{f_i\sim q}\Big[\boldsymbol{f}_i\boldsymbol{f}_i^T\Big]\Big)^{-1}$$

We know from our definition of q being Gaussian and $Cov(Y) = E[YY^T] - E[Y]E[Y^T]$,

$$E_{f_i\sim q}\Big[\boldsymbol{f}_i^T\Big] = \boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}^T$$
$$E_{f_i\sim q}\Big[\boldsymbol{f}_i\boldsymbol{f}_i^T\Big] = \boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}^T + \mathbf{\Sigma}_{f_i|\boldsymbol{x}_i}$$

$$\mathbf{\Lambda} = \Big(\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu})\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}^T\Big)\Big(\sum_{i=1}^{n}\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}^T + \mathbf{\Sigma}_{f_i|\boldsymbol{x}_i}\Big)^{-1}$$

EM Algorithm: M-Step

Similarly, we can find M-Step optimizations for $\boldsymbol{\mu}$ and $\mathbf{\Psi}$.

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$$

$$\Phi = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T - \boldsymbol{x}_i\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}^T\mathbf{\Lambda}^T - \mathbf{\Lambda}\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}\boldsymbol{x}_i^T + \mathbf{\Lambda}(\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}^T + \mathbf{\Sigma}_{f_i|\boldsymbol{x}_i})\mathbf{\Lambda}^T$$

and setting $\mathbf{\Psi}_{ii} = \mathbf{\Phi}_{ii}$, i.e., letting $\mathbf{\Psi}$ be the diagonal matrix containing only diagonal elements of $\mathbf{\Phi}$.

EM algorithm

E Step

$$\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i} = \mathbf{\Lambda}^T(\mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi})^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})$$
$$\mathbf{\Sigma}_{f_i|\boldsymbol{x}_i} = \boldsymbol{I} - \mathbf{\Lambda}^T(\mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi})^{-1}\mathbf{\Lambda}$$

M Step

$$\mathbf{\Lambda} = \Big(\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu})\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}^T\Big)\Big(\sum_{i=1}^{n}\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}^T + \mathbf{\Sigma}_{f_i|\boldsymbol{x}_i}\Big)^{-1}$$

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$$

$$\Phi = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T - \boldsymbol{x}_i\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}^T\mathbf{\Lambda}^T - \mathbf{\Lambda}\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}\boldsymbol{x}_i^T + \mathbf{\Lambda}(\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i}^T + \mathbf{\Sigma}_{f_i|\boldsymbol{x}_i})\mathbf{\Lambda}^T$$

and setting $\mathbf{\Psi}_{ii} = \mathbf{\Phi}_{ii}$, i.e., letting $\mathbf{\Psi}$ be the diagonal matrix containing only diagonal elements of $\mathbf{\Phi}$.

Julia Implementation

- E-Step and M-Step
- Computing Log-likelihood
- EM algorithm
- Application on a synthetic dataset

E-Step

$$\boldsymbol{\mu}_{f_i|\boldsymbol{x}_i} = \mathbf{\Lambda}^T(\mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi})^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})$$
$$\mathbf{\Sigma}_{f_i|\boldsymbol{x}_i} = \boldsymbol{I} - \mathbf{\Lambda}^T(\mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi})^{-1}\mathbf{\Lambda}$$

```
function E_Step(X,mu,Lambda,Psi,k)
    mu_f_by_x = (X - repmat(mu',size(X,1),1))*
                (Lambda'*inv(Lambda*Lambda' + Psi))';
    Sig_f_by_x = eye(k) - Lambda'*
                inv(Lambda*Lambda' + Psi)*Lambda;
    return mu_f_by_x,Sig_f_by_x;
end
```

```
## E_Step (generic function with 1 method)
```

M-Step

```
function M_Step(X,mu_f_by_x,Sig_f_by_x,k)
    nrows, ncols = size(X);
    #Computing mu
    mu = mean(X,1)';
    #Computing Lambda
    Lambda_term1 = zeros(ncols,k); Lambda_term2 = zeros(k,k);
    for i=1:nrows
        Lambda_term1 += ((X[i,:] - mu)*mu_f_by_x[i,:]');
        Lambda_term2 += (mu_f_by_x[i,:]*mu_f_by_x[i,:]'
                        + Sig_f_by_x;
    end
    Lambda = Lambda_term1*inv(Lambda_term2);
    #Computing Psi
    Phi = zeros(ncols,ncols);
    for i=1:nrows
        Phi += (X[i,:]*X[i,:]' - X[i,:]*mu_f_by_x[i,:]'*Lambda'
            - Lambda*mu_f_by_x[i,:]*X[i,:]'
            + Lambda*(mu_f_by_x[i,:]*mu_f_by_x[i,:]'+Sig_f_by_x)*Lambda')
    end
    Psi = diagm(diag(Phi./nrows));
    return mu, Lambda, Psi
```


Computing Log-Likelihood

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}) = \log \prod_{i=1}^{n} \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}|^{1/2}} \exp\left(-\frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{\mu})^T (\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi})^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}) \right)$$

```
function compute_llh(X,mu,Lambda,Psi)
    llh = 0;
    for i=1:size(X,1)
        llh = llh + log(pdf(MvNormal(vec(mu),
                            (Lambda*Lambda')+Psi),X[i,:]));
    end
    return llh;
end
```

```
## compute_llh (generic function with 1 method)
```


Overall EM approach for Factor Analysis

```
function fa_em(X,k)
    max_Iter = 100;
    eps = 0.0001;
    llh = -Inf*ones(max_Iter+1);
    mu = mean(X,1)';
    Lambda = rand(size(X,2),k);
    Psi = diagm(rand(size(X,2)));
    llh[1] = compute_llh(X,mu,Lambda,Psi);
    for i=1:max_Iter
        mu_f_by_x,Sig_f_by_x = E_Step(X,mu,Lambda,Psi,k);
        mu_new, Lambda_new, Psi_new = M_Step(X,mu_f_by_x,Sig_f_by_x,k);
        llh[i+1] = compute_llh(X,mu_new,Lambda_new,Psi_new);
        if(sum(abs.(mu_new-mu))<eps &&
            sum(abs.(Lambda_new-Lambda))<eps && sum(abs.(Psi_new-Psi))<eps)
            break;
        end
        mu = mu_new; Lambda = Lambda_new; Psi = Psi_new;
    end
    return mu, Lambda, Psi, llh;
end
```


Generating sample data

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$
$$\boldsymbol{x}|\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu} + \boldsymbol{\Lambda}\boldsymbol{f}, \boldsymbol{\Psi})$$

```
mu = [0 0 0 0]';
Lambda =
[1.0  0
 1.0  0
 0    0.99
 0    1.0];
Psi =  diagm([0.1, 0.1, 0.1, 0.1]);
d1 = MvNormal([0,0],ones(2));
X = zeros(10000,4);
for i=1:10000
    f = rand(d1,1);
    d2 = MvNormal(vec(mu+ Lambda*f),Psi);
    x = rand(d2,1);
    X[i,:] = x';
end
```


Using EM to learn Lambda

```
mu_est, Lambda_est, Psi_est, llh = fa_em(X,2);
```

```
mu_est
```

```
## 4×1 Array{Float64,2}:
##  -0.0092927
##  -0.0126923
##  -0.00481124
##   0.00146455
```

```
Lambda_est
```

```
## 4×2 Array{Float64,2}:
##   2.73949  1.78666
##   2.71648  1.76701
##  -2.61815  3.98344
##  -2.62815  4.01916
```

```
Psi_est
```

```
## 4×4 Array{Float64,2}:
##  0.094361  0.0       0.0       0.0
##  0.0       0.10286   0.0       0.0
##  0.0       0.0       0.100012  0.0
##  0.0       0.0       0.0       0.099
```


Summary

- Factor analysis model: $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Lambda}\boldsymbol{f} + \epsilon$
  - Latent variable $\boldsymbol{f}$ is continuous
- Factor Analysis is useful to model data when it lies in a 'nearly' linear subspace
- Probabilistic forumation
  - $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  - $\boldsymbol{x}|\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu} + \boldsymbol{\Lambda}\boldsymbol{f}, \boldsymbol{\Psi})$
- One approach for parameter estimation is MLE
  - Closed form solution is not possible
  - EM approach for modeling continuous latent variables
- *Advantages:* Identifying hidden factors, Dimensionality reduction
- *Disadvantages:* Indentifiability/Rotation problem, Data must follow Gaussian dist.