EE613
Machine Learning for Engineers

SUBSPACE CLUSTERING

Sylvain Calinon
Robot Learning & Interaction Group
Idiap Research Institute
Oct. 25, 2017

SUBSPACE CLUSTERING (Wed, Oct. 25)

HIDDEN MARKOV MODELS (Wed, Nov. 1)

LINEAR REGRESSION (Thu, Nov. 9)

GAUSSIAN MIXTURE REGRESSION (Wed, Dec. 13)

GAUSSIAN PROCESS REGRESSION (Wed, Dec. 20)

Time series analysis and synthesis,

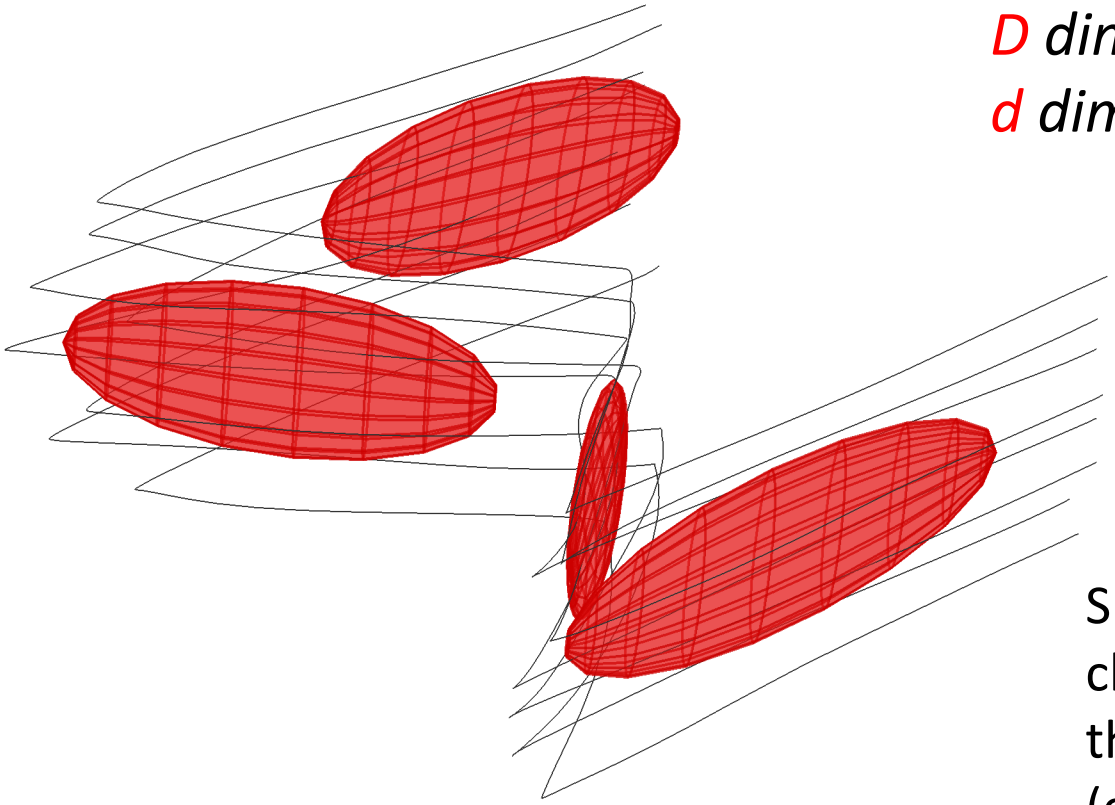
Multivariate data processing

Outline

- High-dimensional data clustering (HDDC)
  Matlab code: *demo_HDDC01.m*
- Mixture of factor analyzers (MFA)
  Matlab code: *demo_MFA01.m*
- Mixture of probabilistic principal component analyzers (MPPCA)
  Matlab code: *demo_MPPCA01.m*
- GMM with semi-tied covariance matrices
  Matlab code: *demo_semitiedGMM01.m*

Introduction

K clusters
N datapoints
D dimensions (original space)
d dimensions (latent space)



Subspace clustering aims at clustering data while reducing the dimension of each cluster (cluster-dependent subspace)

Considering the two problems separately (clustering, then subspace projection) can be inefficient and can produce poor local optima, especially when datapoints of high dimensions are considered.

Example of application: Whole body motion

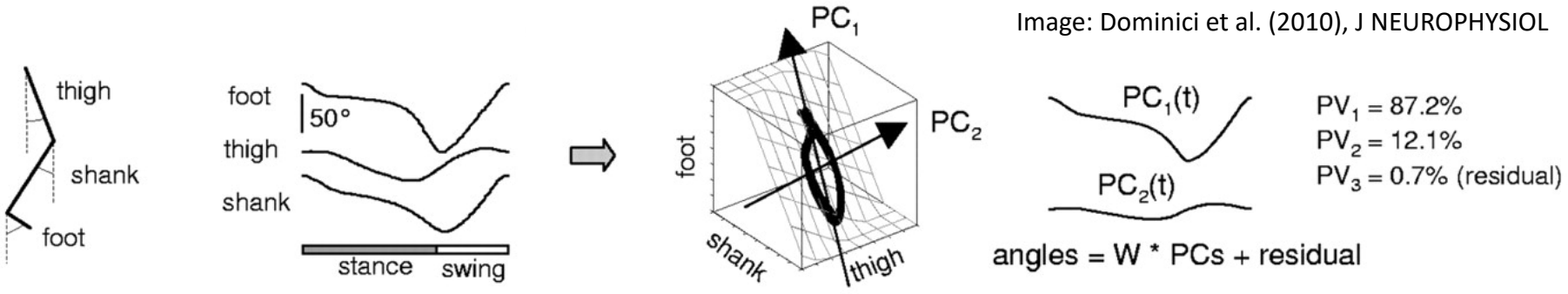
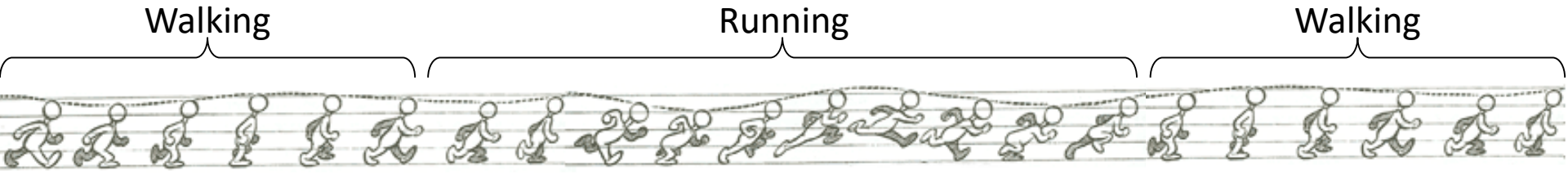


Image: Dominici et al. (2010), J NEUROPHYSIOL

→ **About 90% of variance in walking motion can be explained by 2 principal components**
→ **Each type of periodic motion can be characterized by a different subspace**



→ **Requires clustering of the complete motion into different locomotion phases**
→ **Requires extraction of coordination patterns for each cluster**

Curse of dimensionality in GMM encoding

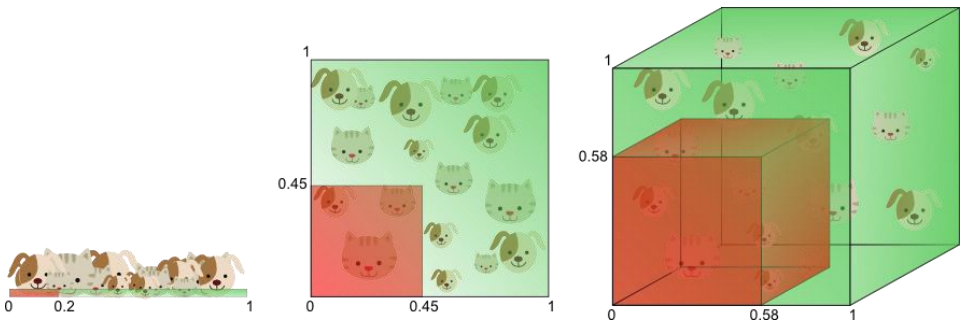


Image: datasciencecentral.com

K clusters
N datapoints
D dimensions (original space)
d dimensions (latent space)

Classical Gaussian mixture models (GMM) tend to perform poorly in high-dimensional spaces if too few datapoints are available.

For a dataset $\{\boldsymbol{\xi}_t\}_{t=1}^{N}$ with $\boldsymbol{\xi}_t \in \mathbb{R}^D$, the *curse of dimensionality* appears if the dimension of the data D is too large compared to the size of the training set N.

In particular, the problem can affect the full covariance matrices $\boldsymbol{\Sigma}_i \in \mathbb{R}^{D\times D}$ because the number of parameters to be estimated grows quadratically with D.

Curse of dimensionality

Some characteristics of high-dimensional spaces can ease the classification of data. Indeed, having different groups living in different subspaces may be a useful property for discriminating the groups.

Subspace clustering exploits the phenomenon that high-dimensional spaces are mostly empty to ease the discrimination between groups of points.

→ Curse of dimensionality or…
blessing of dimensionality?

Curse of dimensionality

N datapoints
D dimensions (original space)
d dimensions (latent space)

Bouveyron and Brunet (2014, COMPUT STAT DATA AN) reviewed various ways of handling the problem of high-dimensional data in clustering problems:

1. Since *D* is too large w.r.t. *N*, a global dimensionality reduction should be applied as a pre-processing step to reduce *D*.
2. Since *D* is too large w.r.t. *N*, the solution space contains many poor local optima. The solution space should be smoothed by introducing ridge or lasso regularization in the estimation of the covariance (avoiding numerical problem and singular solutions when inverting the covariances). A simple form of regularization can be achieved after the maximization step of each EM loop.
3. Since *D* is too large w.r.t. *N*, the model is probably over-parametrized, and a more parsimonious model should be used (thus estimating a fewer number of parameters).

Gaussian Mixture Model (GMM)

K Gaussians
N datapoints of dimension *D*

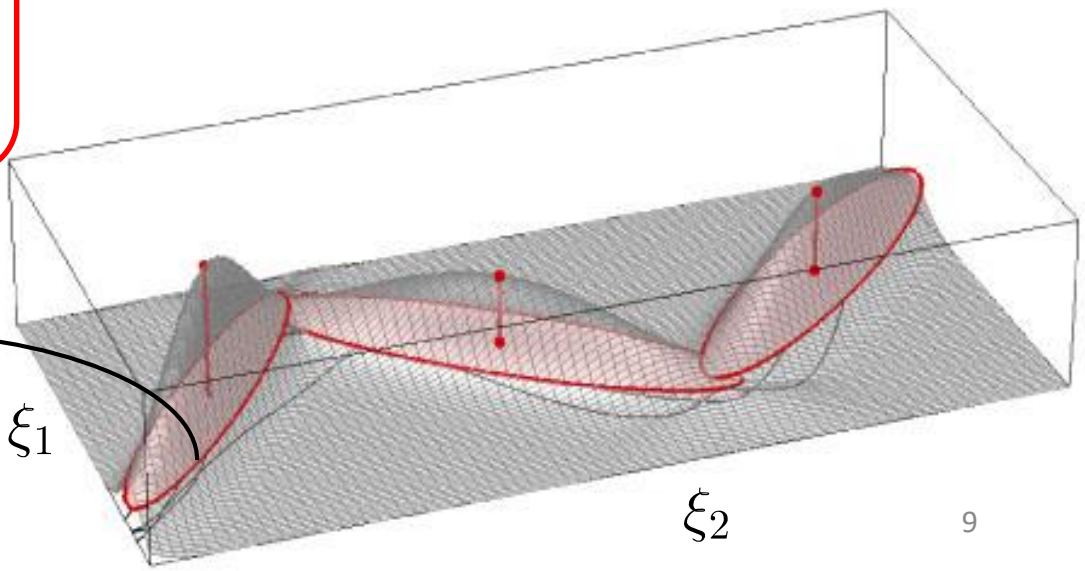
$$\mathcal{P}(\boldsymbol{\xi}_t) = \sum_{i=1}^{K} \pi_i \, \mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

$$\mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{\frac{D}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)\right)$$

$\boldsymbol{\xi} \in \mathbb{R}^{D \times N}$ Observations ($N = \sum_{m=1}^{M} T_m$, the m-th trajectory has T_m datapoints)

$\pi_i \in \mathbb{R}$	Mixing coefficient
$\boldsymbol{\mu}_i \in \mathbb{R}^D$	Center (or mean)
$\boldsymbol{\Sigma}_i \in \mathbb{R}^{D \times D}$	Covariance matrix

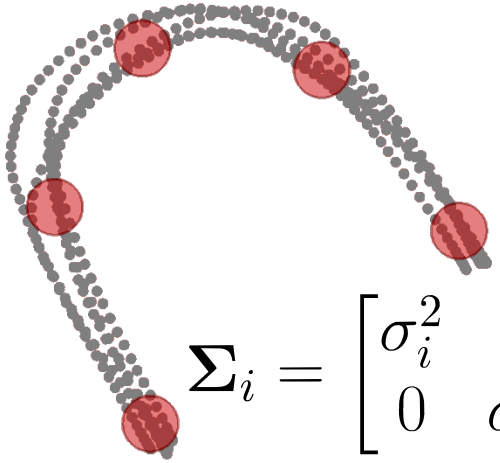
Parameters $\boldsymbol{\Theta}^{\text{GMM}} = \{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{K}$



Equidensity contour of one standard deviation

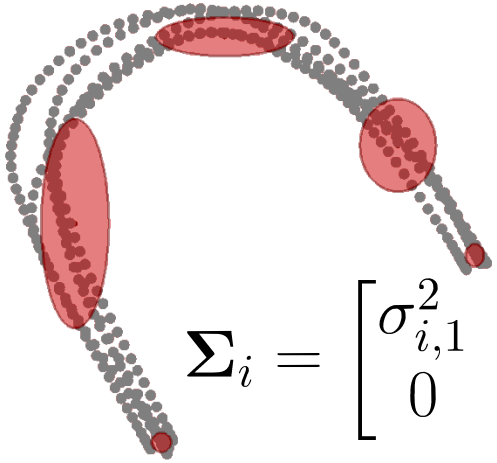
Covariance structures in GMM

Isotropic



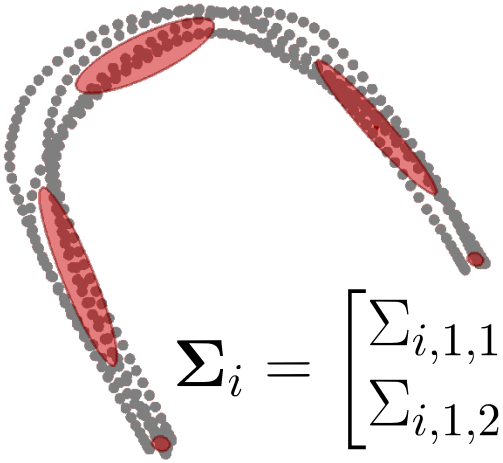
$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{bmatrix}$$

Diagonal



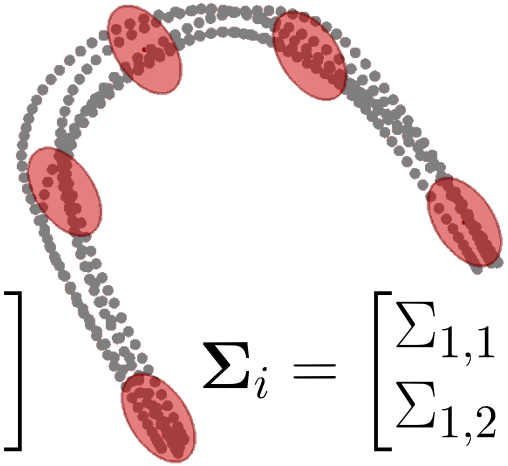
$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_{i,1}^2 & 0 \\ 0 & \sigma_{i,2}^2 \end{bmatrix}$$

Full



$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \Sigma_{i,1,1} & \Sigma_{i,1,2} \\ \Sigma_{i,1,2} & \Sigma_{i,2,2} \end{bmatrix}$$

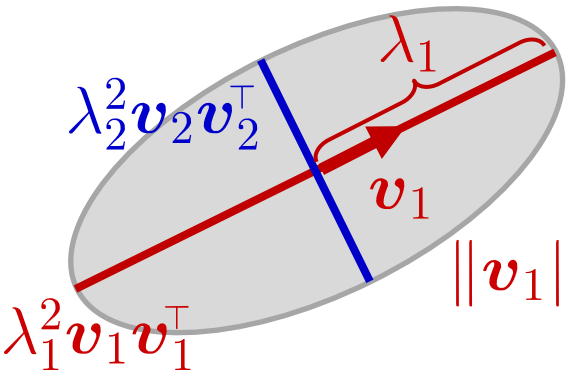
Tied



$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{1,2} & \Sigma_{2,2} \end{bmatrix}$$

Multivariate normal distribution - Stochastic sampling

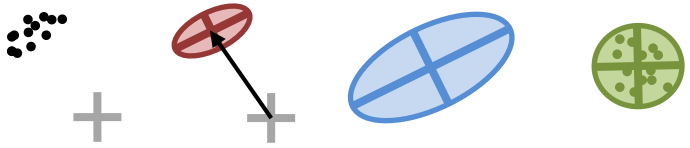
The eigendecomposition of $\boldsymbol{\Sigma}$ is expressed in a matrix form as



$$\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^\top = \sum_{j=1}^{D} \lambda_j^2 \boldsymbol{v}_j \boldsymbol{v}_j^\top$$

with $\boldsymbol{V} = \left[\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D\right]$ and $\boldsymbol{D} = \begin{bmatrix} \lambda_1^2 & 0 & \cdots & 0 \\ 0 & \lambda_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_D^2 \end{bmatrix}$

By using this notation, datapoints can be stochastically generated with



$$\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \Longleftrightarrow \quad \boldsymbol{\xi} \sim \boldsymbol{\mu} + \boldsymbol{V}\boldsymbol{D}^{\frac{1}{2}}\,\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

Expectation-maximization (EM)

$z_{t,i} = 1$ if $\boldsymbol{\xi}_t$ is part of cluster i. It is 0 otherwise.

Each datapoint $\boldsymbol{\xi}_t$ is associated with a hidden/missing variable $\boldsymbol{z}_t$. The goal is to maximize the log-likelihood of the observed data

$$\mathcal{L}(\boldsymbol{\Theta}) = \sum_{t=1}^{N} \log \mathcal{P}(\boldsymbol{\xi}_t|\boldsymbol{\Theta}) = \sum_{t=1}^{N} \log \left(\sum_{\boldsymbol{z}_t} \mathcal{P}(\boldsymbol{\xi}_t, \boldsymbol{z}_t|\boldsymbol{\Theta}) \right)$$

which is hard to optimize ("log cannot be pushed inside the sum").

We can get around this problem by instead employing the expected complete data log-likelihood

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}}) = \mathbb{E}\Big[\sum_{t=1}^{N} \log \mathcal{P}(\boldsymbol{\xi}_t, \boldsymbol{z}_t|\boldsymbol{\Theta}) \;\Big|\; \boldsymbol{\xi}, \boldsymbol{\Theta}^{\text{old}} \Big]$$

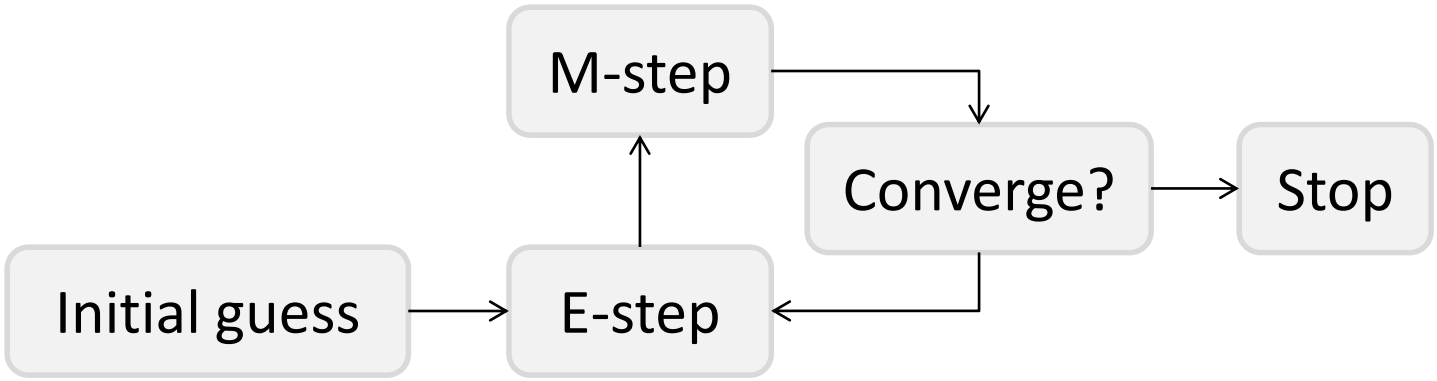
where $\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})$ is called the auxiliary function.

Expectation-maximization (EM)

The expectation is taken with respect to the old model parameters $\boldsymbol{\Theta}^{\text{old}}$ and the observed dataset $\boldsymbol{\xi}$.

The *E-step* computes the terms in $\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})$ of which the likelihood depends on, known as the expected sufficient statistics.

The *M-step* then optimizes $\mathcal{Q}$ with respect to $\boldsymbol{\Theta}$.



EM for GMM

When applied to GMM, the auxiliary function $\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})$ takes the form

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}}) = \mathbb{E}\left[\sum_{t=1}^{N} \log \mathcal{P}(\boldsymbol{\xi}_t, \boldsymbol{z}_t | \boldsymbol{\Theta}) \;\middle|\; \boldsymbol{\xi}, \boldsymbol{\Theta}^{\text{old}} \right]$$

$$= \sum_{t=1}^{N} \mathbb{E}\left[\log \left(\prod_{i=1}^{K} \left(\pi_i \, \mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right)^{z_{t,i}} \right) \;\middle|\; \boldsymbol{\xi}, \boldsymbol{\Theta}^{\text{old}} \right]$$

$$= \sum_{t=1}^{N} \sum_{i=1}^{K} \mathbb{E}\left[\log \left(\left(\pi_i \, \mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right)^{z_{t,i}} \right) \;\middle|\; \boldsymbol{\xi}, \boldsymbol{\Theta}^{\text{old}} \right]$$

$$= \sum_{t=1}^{N} \sum_{i=1}^{K} \mathbb{E}\left[z_{t,i} \,|\, \boldsymbol{\xi}, \boldsymbol{\Theta}^{\text{old}} \right] \log \left(\pi_i \, \mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right)$$

$$= \sum_{t=1}^{N} \sum_{i=1}^{K} h_{t,i} \left(\log(\pi_i) + \log \left(\mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right) \right)$$

$$= \sum_{t=1}^{N} \sum_{i=1}^{K} h_{t,i} \left(\log(\pi_i) - \frac{1}{2} \log |\boldsymbol{\Sigma}_i| - \frac{1}{2} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i) - \frac{D}{2} \log(2\pi) \right)$$

$z_{t,i} = 1$ if $\boldsymbol{\xi}_i$ is part of cluster i. It is 0 otherwise.

$e.g.\ \prod_{i=1}^{3} \pi_i^{z_i} = \pi_1^{z_1} \cdot \pi_2^{z_2} \cdot \pi_3^{z_3} = \pi_1^0 \cdot \pi_2^0 \cdot \pi_3^1 = 1 \cdot 1 \cdot \pi_3$

$\log(ab) = \log(a) + \log(b)$

$\log(a^b) = b \log(a)$

$\log(\exp(a)) = a$

$\mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = (2\pi)^{-\frac{D}{2}} |\boldsymbol{\Sigma}_i|^{-\frac{1}{2}} \cdot \exp\left(-\frac{1}{2} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i) \right)$

where $h_{t,i}$ is the responsibility that cluster i takes for datapoint $\boldsymbol{\xi}_t$.

EM for GMM

Setting

$$\frac{\partial \mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})}{\partial \pi_i} = 0 \qquad \frac{\partial \mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})}{\partial \boldsymbol{\mu}_i} = 0 \qquad \frac{\partial \mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})}{\partial \boldsymbol{\Sigma}_i} = 0$$

and solving for π_i, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ results in an EM procedure to compute the maximum likelihood estimate of the parameters.

EM for GMM

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}}) = \sum_{t=1}^{N} \sum_{i=1}^{K} h_{t,i} \Big(\log(\pi_i) - \frac{1}{2} \log |\boldsymbol{\Sigma}_i| - \frac{1}{2} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i) - \frac{D}{2} \log(2\pi) \Big)$$

By using the linear algebra relations $\quad \big(= 2\boldsymbol{A}\boldsymbol{x} \text{ if } \boldsymbol{A} \text{ symmetric}\big)$

$$\frac{\partial}{\partial \boldsymbol{A}} \log |\boldsymbol{A}| = (\boldsymbol{A}^\top)^{-1} \qquad \frac{\partial}{\partial \boldsymbol{A}} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x} = \boldsymbol{x}\boldsymbol{x}^\top \qquad \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x} = (\boldsymbol{A} + \boldsymbol{A}^\top)\boldsymbol{x}$$

and the derivation chain rule, we obtain

$$\frac{\partial \mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})}{\partial \boldsymbol{\mu}_i} = \frac{1}{2} \sum_{t=1}^{N} h_{t,i}\, 2\boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i) = \boldsymbol{\Sigma}_i^{-1} \sum_{t=1}^{N} h_{t,i} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i) = 0$$

$$\iff \boldsymbol{\mu}_i = \frac{\sum_{t=1}^{N} h_{t,i} \boldsymbol{\xi}_t}{\sum_{t=1}^{N} h_{t,i}}$$

$$\frac{\partial \mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})}{\partial \boldsymbol{\Sigma}_i} = \frac{1}{2} \boldsymbol{\Sigma}_i \sum_{t=1}^{N} h_{t,i} - \frac{1}{2} \sum_{t=1}^{N} h_{t,i} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top = 0$$

$$\iff \boldsymbol{\Sigma}_i = \frac{\sum_{t=1}^{N} h_{t,i} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top}{\sum_{t=1}^{N} h_{t,i}}$$

EM for GMM

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}}) = \sum_{t=1}^{N}\sum_{i=1}^{K} h_{t,i}\Big(\log(\pi_i) - \frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i) - \frac{D}{2}\log(2\pi)\Big)$$

For π_i, we need to ensure the constraint $\sum_{i=1}^{K}\pi_i = 1$, which can be achieved through a Lagrange multiplier λ, yielding

$$\frac{\partial}{\partial \pi_i}\left[\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}}) - \lambda\Big(\sum_{i=1}^{K}\pi_i - 1\Big)\right] = \frac{1}{\pi_i}\sum_{t=1}^{N} h_{t,i} - \lambda = 0$$

The sum over K of the above relation provides

$$\sum_{t=1}^{N}\sum_{i=1}^{K} h_{t,i} = \lambda\sum_{i=1}^{K}\pi_i \quad \Longleftrightarrow \quad \lambda = N$$

$\sum_{i=1}^{K} h_{t,i} = 1,\ \sum_{i=1}^{K}\pi_i = 1$

which can be reintroduced in the equation to find

$$\frac{1}{\pi_i}\sum_{t=1}^{N} h_{t,i} - N = 0 \quad \Longleftrightarrow \quad \pi_i = \frac{\sum_{t=1}^{N} h_{t,i}}{N}$$

EM for GMM: Resulting procedure

K Gaussians
N datapoints

E-step:

$$h_{t,i} = \frac{\pi_i \, \mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

M-step:

$$\pi_i \leftarrow \frac{\sum_{t=1}^{N} h_{t,i}}{N},$$

$$\boldsymbol{\mu}_i \leftarrow \frac{\sum_{t=1}^{N} h_{t,i} \, \boldsymbol{\xi}_t}{\sum_{t=1}^{N} h_{t,i}},$$

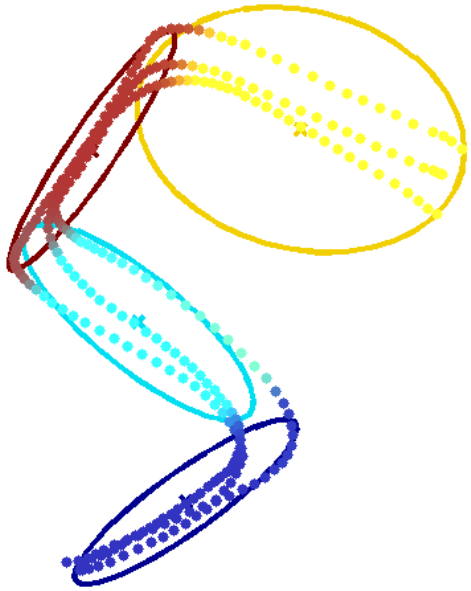
$$\boldsymbol{\Sigma}_i \leftarrow \frac{\sum_{t=1}^{N} h_{t,i} \, (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top}{\sum_{t=1}^{N} h_{t,i}}$$

These results can be intuitively interpreted in terms of normalized counts.
EM provides a systematic approach to derive such procedure.

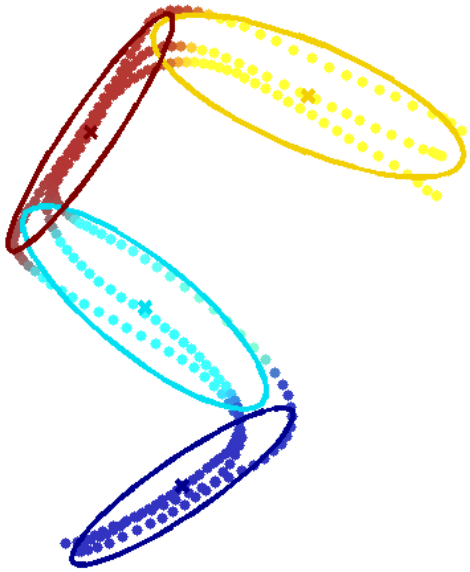
→ **Weighted averages taking into account the responsibility of each datapoint in each cluster.**

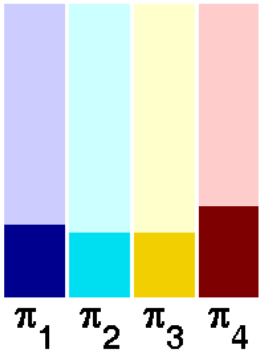
EM for GMM

E-step

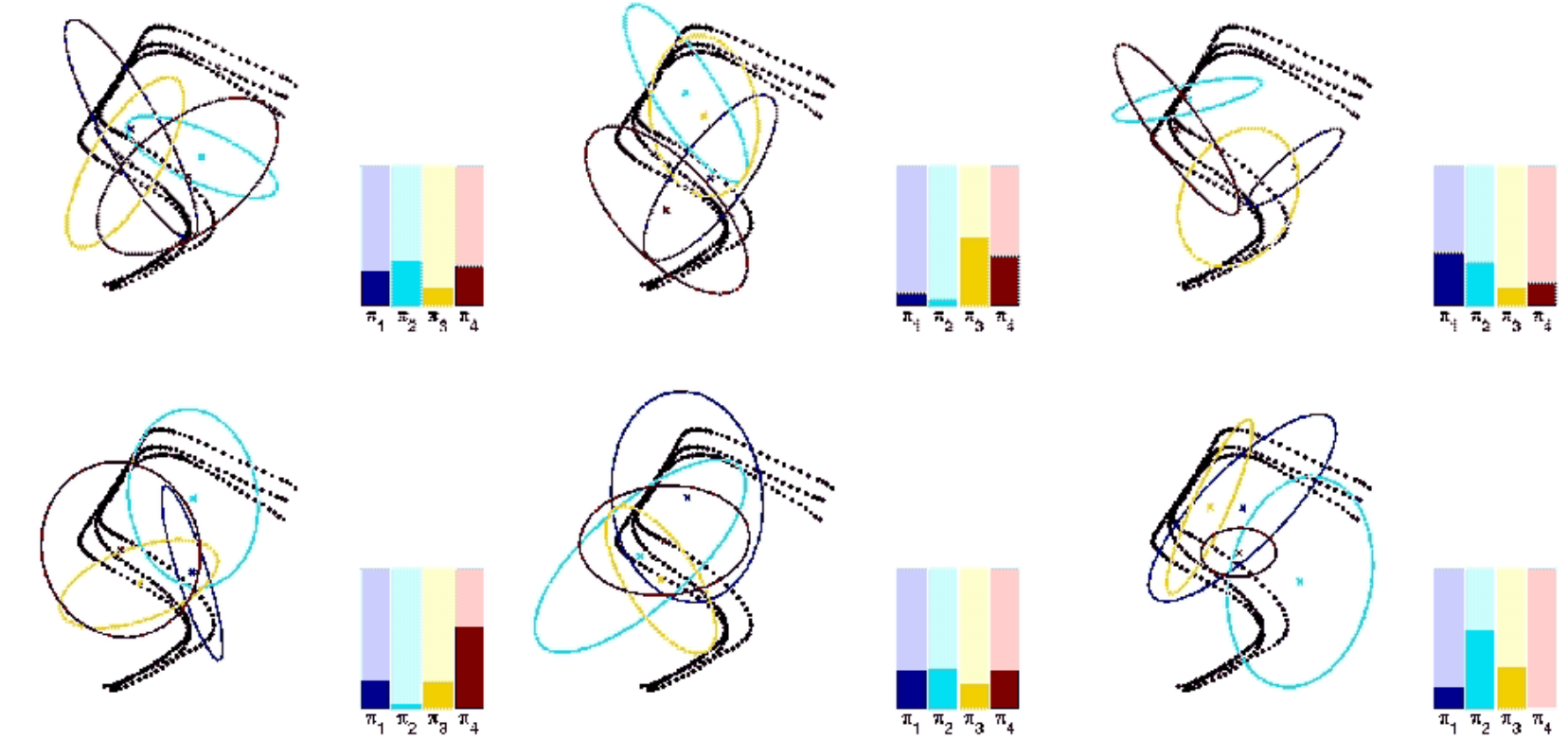


M-step





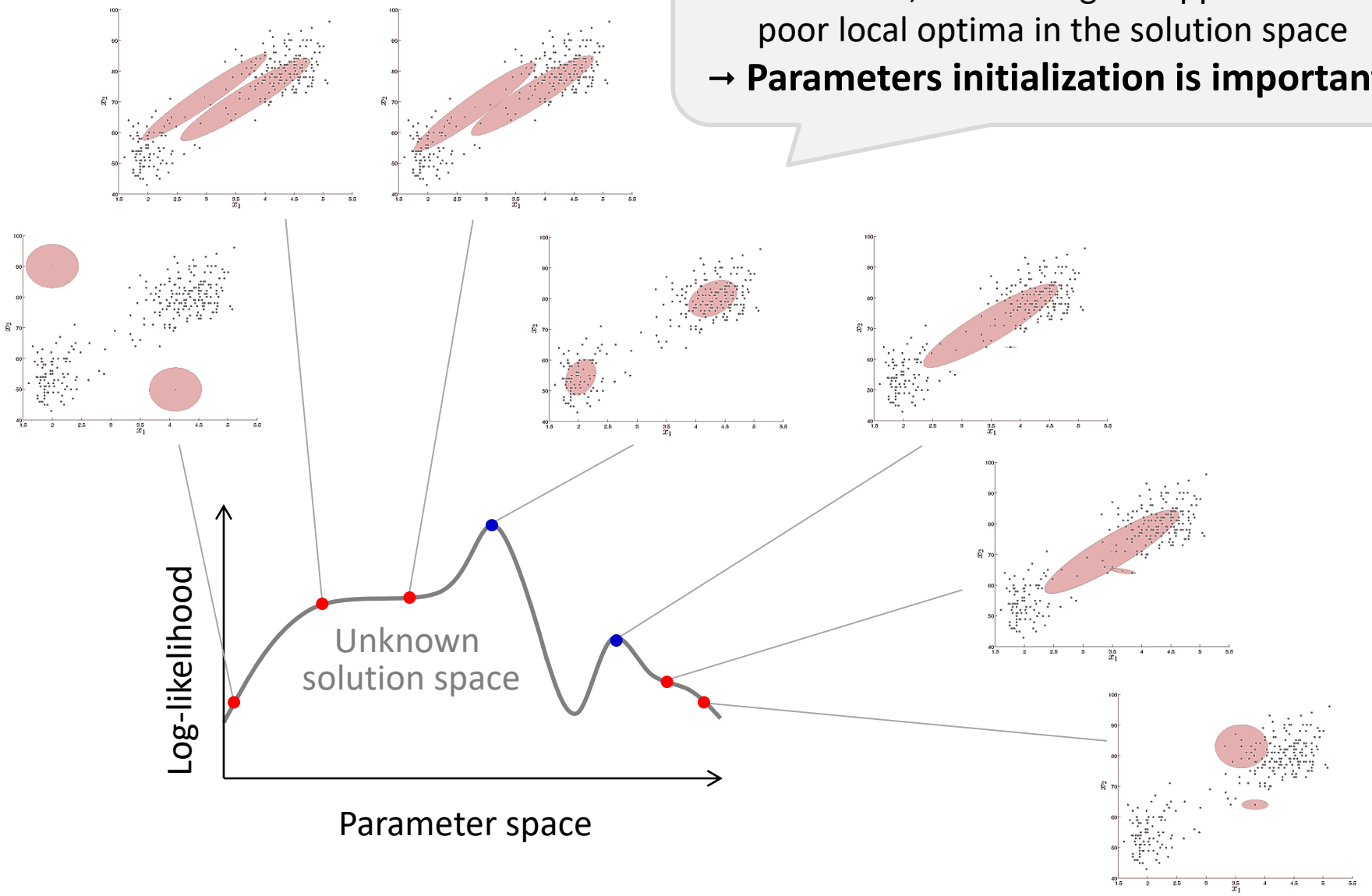
EM for GMM: Local optima issue



Local optima in EM

EM will improve the likelihood at each iteration, but it can get trapped into poor local optima in the solution space

→ **Parameters initialization is important!**



Parameters estimation in GMM… in 1893

54 pages!
Proposed solution:
Moment-based approach requiring to solve a polynomial of degree 9…

III. *Contributions to the Mathematical Theory of Evolution.*

By KARL PEARSON, *University College, London.*

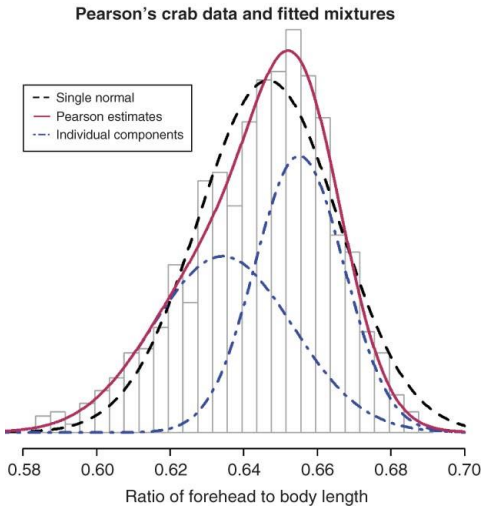
Communicated by Professor HENRICI, *F.R.S.*

Received October 18,—Read November 16, 1893.

[PLATES 1—5.]

CONTENTS.

	Page.
I.—On the Dissection of Asymmetrical Frequency-Curves. General Theory, §§ 1–8.	71–85
Example: Professor WELDON'S measurements of the "Forehead" of Crabs. §§ 9–10	85–90
II.—On the Dissection of Symmetrical Frequency-Curves. General Theory, §§ 11–12	
Application. Crabs "No. 4." §§ 13–15	90–100
	00–106
	106
	107
	07–110



… which does not mean that moment-based approaches are old-fashioned!

They are actually today popular again with new developments related to spectral decomposition.

High-dimensional data clustering (HDDC)

Matlab code: demo_HDDC01.m

[C. Bouveyron and C. Brunet. Model-based clustering of high-dimensional data: A review. Computational Statistics and Data Analysis, 71:52–78, March 2014]

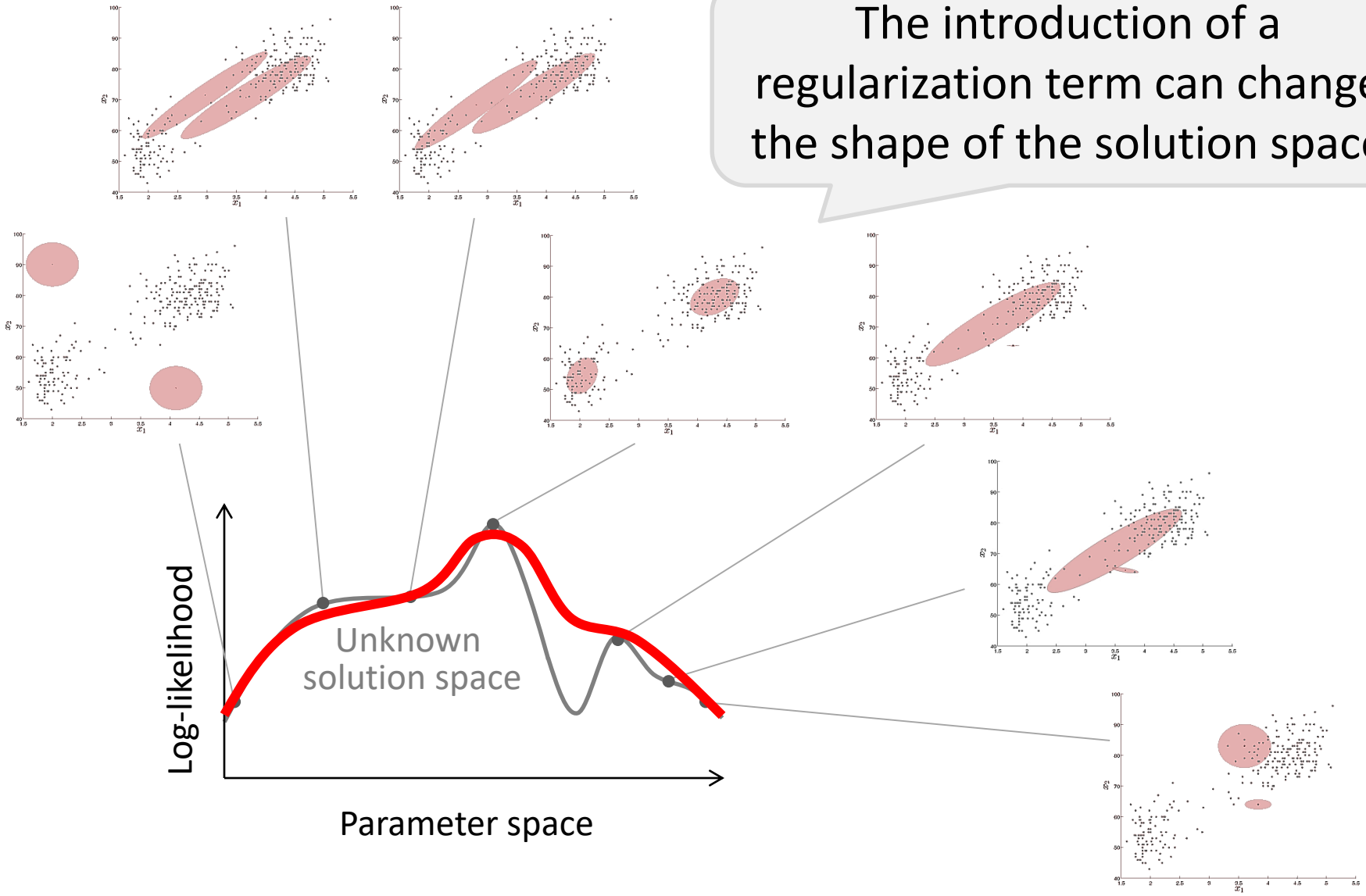
Curse of dimensionality

Bouveyron and Brunet (2014, COMPUT STAT DATA AN) reviewed various ways of viewing the problem and coping with high-dimensional data in clustering problems:

1. Since ***D*** is too large wrt ***N***, a global dimensionality reduction should be applied as a pre-processing step to reduce ***D***.
2. Since ***D*** is too large wrt ***N***, the solution space contains many poor local optima; the solution space should be smoothed by introducing ridge or lasso regularization in the estimation of the covariance (avoiding numerical problem and singular solutions when inverting the covariances). A simple form of regularization can be achieved after the maximization step of each EM loop.
3. Since ***D*** is too large wrt ***N***, the model is probably over-parametrized, and a more parsimonious model should be used (thus estimating a fewer number of parameters).

Regularization of the GMM parameters

The introduction of a regularization term can change the shape of the solution space



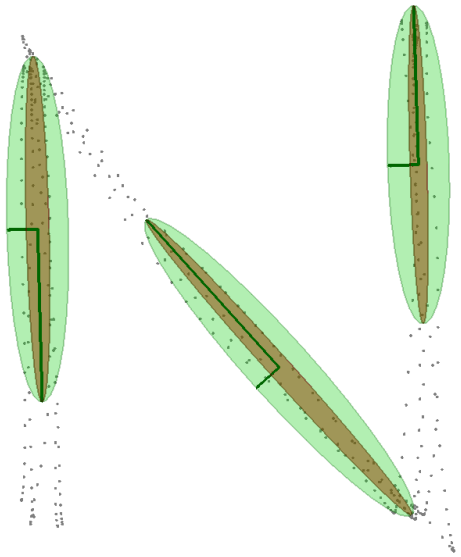
Regularization of the GMM parameters

Regularization with minimal admissible eigenvalue:

$$\boldsymbol{\Sigma}_i \leftarrow \boldsymbol{V}_i \tilde{\boldsymbol{D}}_i \boldsymbol{V}_i^\top$$

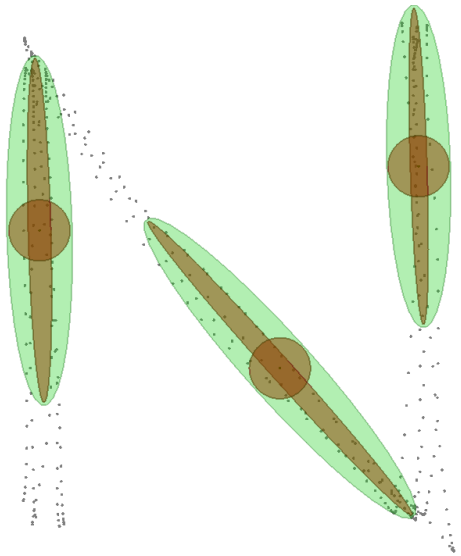
$$\text{with } \tilde{\boldsymbol{D}}_i = \begin{bmatrix} \tilde{\lambda}_{i,1}^2 & 0 & \cdots & 0 \\ 0 & \tilde{\lambda}_{i,2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{\lambda}_{i,D}^2 \end{bmatrix}$$

$$\text{and } \tilde{\lambda}_{i,j}^2 = \max(\lambda_{i,j}^2, \lambda_{\min}^2) \; \forall j \in \{1, \ldots, D\}$$



Tikhonov regularization with diagonal isotropic covariance:

$$\boldsymbol{\Sigma}_i \leftarrow \boldsymbol{\Sigma}_i + \boldsymbol{I}\lambda_{\min}^2$$



High-dimensional data clustering (HDDC)

The HDDC approach (Bouveyron, 2007, COMPUT STAT DATA AN) addresses both subspace clustering and regularization.

One implementation considers that the subspace of each cluster i is generated by the first d_i eigenvectors associated with the first d_i eigenvalues $\lambda_{i,k}$, and that outside of this subspace, the variance is isotropic, modeled by

$$\bar{\lambda}_i = \frac{1}{D-d_i}\sum_{k=d_i+1}^{D}\lambda_{i,k} = \frac{1}{D-d_i}\left(\mathrm{tr}(\mathbf{\Sigma}_i) - \sum_{k=1}^{d_i}\lambda_{i,k}\right)$$

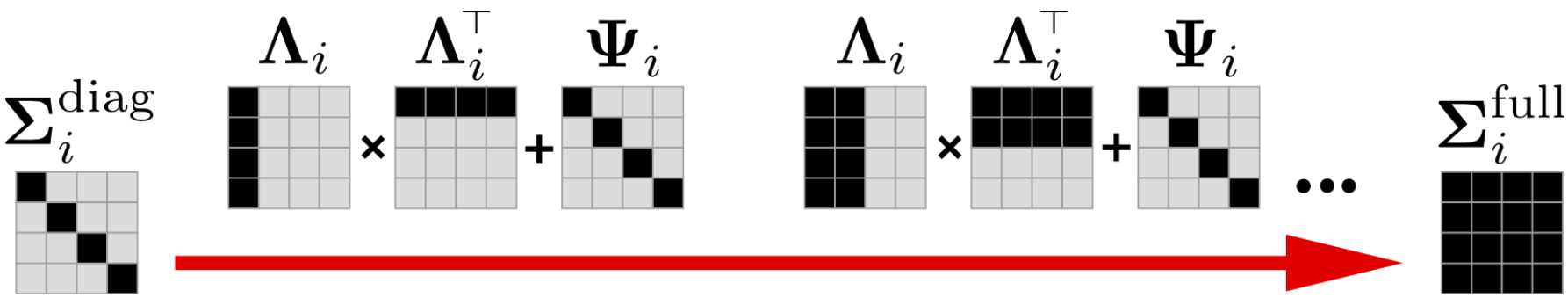
which is used to reconstruct a full covariance matrix by replacing the last $D - d_i$ eigenvalues with $\bar{\lambda}_i$.

Mixture of factor analyzers (MFA)

Matlab code: demo_MFA01.m

[P. D. McNicholas and T. B. Murphy. Parsimonious Gaussian mixture models. Statistics and Computing, 18(3):285–296, September 2008]

Mixture of factor analyzers (MFA)



MFA assumes for each covariance i a structure of the form

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i$$

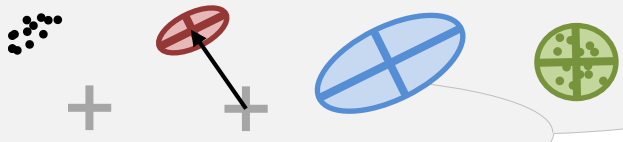
where $\boldsymbol{\Lambda}_i \in \mathbb{R}^{D \times d}$, known as the *factor loading matrix*, typically has $d < D$ (providing a parsimonious representation of the data), and a diagonal noise matrix $\boldsymbol{\Psi}_i$.

The *mixture of probabilistic principal component analyzers* (MPPCA) is a special case of MFA with the distribution of the errors assumed to be isotropic with $\boldsymbol{\Psi}_i = \boldsymbol{I}\sigma_i^2$.

Mixture of factor analyzers (MFA) $\boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i$

$$\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{D}^{\frac{1}{2}}(\boldsymbol{V}\boldsymbol{D}^{\frac{1}{2}})^\top$$

$$\boldsymbol{\xi} \sim \boldsymbol{\mu} + \boldsymbol{V}\boldsymbol{D}^{\frac{1}{2}}\mathcal{N}(\mathbf{0}, \boldsymbol{I})$$



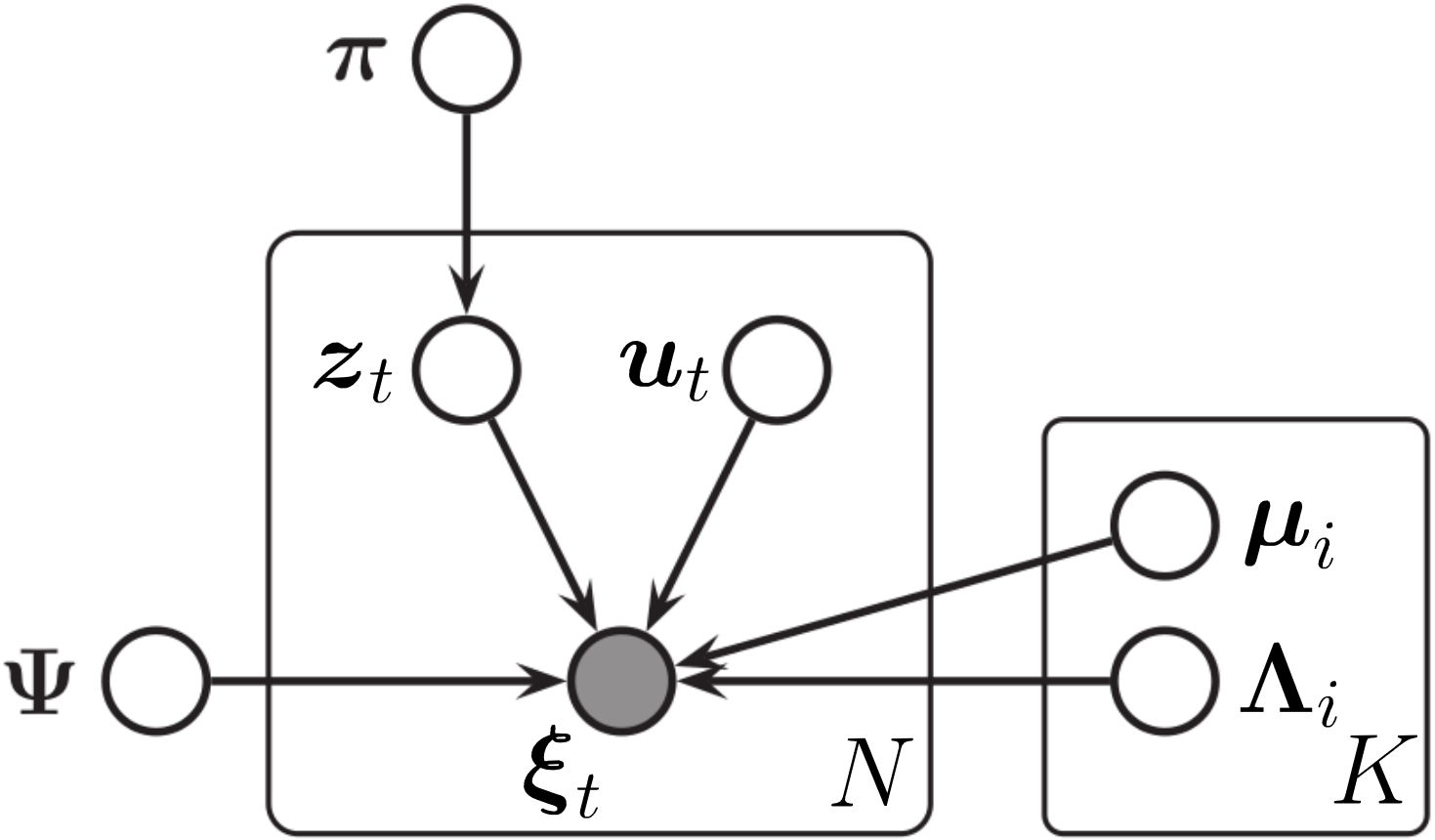
In MFA, the generative model for the i-th mixture component assumes that a D-dimensional random vector $\boldsymbol{\xi}$ is modeled using a d-dimensional vector of latent (unobserved) factors $\boldsymbol{u}$

$$\boldsymbol{\xi} = \boldsymbol{\Lambda}_i \boldsymbol{u} + \boldsymbol{\mu}_i + \boldsymbol{\epsilon}_i$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^D$ is the mean vector of the i-th factor analyzer, $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ (the factors are assumed to be distributed according to a zero-mean normal with unit variance), and $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}_i)$ is a normal noise with diagonal covariance $\boldsymbol{\Psi}_i$.

This diagonality is a key assumption in factor analysis. Namely, the observed variables are independent given the factors, and the goal of MFA is to best model the covariance structure of $\boldsymbol{\xi}$.

Mixture of factor analyzers (MFA): graphical model



For MFA with covariance structure $\boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}$

(for $\boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i$, $\boldsymbol{\Psi}_i$ is moved to the right)

Mixture of factor analyzers (MFA)

$$\boldsymbol{\xi} = \boldsymbol{\Lambda}_i \boldsymbol{u} + \boldsymbol{\mu}_i + \boldsymbol{\epsilon}_i$$

It follows from this model that the marginal distribution of $\boldsymbol{\xi}$ for the i-th component is

$$\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\mu}_i,\ \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i)$$

and the joint distribution of $\boldsymbol{\xi}$ and $\boldsymbol{u}$ is

$$\begin{bmatrix} \boldsymbol{\xi} \\ \boldsymbol{u} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_i \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i & \boldsymbol{\Lambda}_i \\ \boldsymbol{\Lambda}_i^\top & \boldsymbol{I} \end{bmatrix} \right)$$

To make some parallels with PCA, the above can be used to show that the d factors are informative projections of the data, which can be computed by Gaussian conditioning, corresponding to the affine projection

$$\boldsymbol{u}|\boldsymbol{\xi} \sim \mathcal{N}\Big(\boldsymbol{B}_i(\boldsymbol{\mu}_i - \boldsymbol{\xi}),\ \boldsymbol{I} - \boldsymbol{B}_i \boldsymbol{\Lambda}_i \Big) \quad \text{with} \quad \boldsymbol{B}_i = \boldsymbol{\Lambda}_i^\top (\boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i)^{-1}$$

Mixture of factor analyzers (MFA) $\boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i$

This can be used to estimate the second moment of the factors

$$\begin{aligned}\mathbb{E}(\boldsymbol{u}\boldsymbol{u}^\top|\boldsymbol{\xi}) &= \mathrm{cov}(\boldsymbol{u}|\boldsymbol{\xi}) + \mathbb{E}(\boldsymbol{u}|\boldsymbol{\xi})\,\mathbb{E}(\boldsymbol{u}|\boldsymbol{\xi})^\top \\ &= \boldsymbol{I} - \boldsymbol{B}_i\boldsymbol{\Lambda}_i + \boldsymbol{B}_i(\boldsymbol{\mu}_i - \boldsymbol{\xi})(\boldsymbol{\mu}_i - \boldsymbol{\xi})^\top \boldsymbol{B}_i^\top\end{aligned}$$

which provides a measure of uncertainty in the factors that has no analogue in PCA.

This relation is exploited to derive an EM algorithm to train an MFA model of K components with parameters

$$\boldsymbol{\Theta}^{\mathrm{MFA}} = \{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i, \boldsymbol{\Psi}_i\}_{i=1}^K$$

In the special case of a single cluster, it is worth noting that, in contrast to PPCA, FA also requires an EM algorithm to estimate

$$\boldsymbol{\Theta}^{\mathrm{FA}} = \{\boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}\}$$

Estimation of parameters in MFA

In the case of MFA, it is considered that each datapoint $\boldsymbol{\xi}_t$ is associated with hidden variables $\boldsymbol{z}_t$ and $\boldsymbol{u}_t$, and the goal is to maximize

$$\mathcal{L}(\boldsymbol{\Theta}) = \sum_{t=1}^{N} \log \mathcal{P}(\boldsymbol{\xi}_t|\boldsymbol{\Theta}) = \sum_{t=1}^{N} \log \left(\sum_{\boldsymbol{z}_t} \mathcal{P}(\boldsymbol{\xi}_t, \boldsymbol{z}_t, \boldsymbol{u}_t|\boldsymbol{\Theta}) \right)$$

which is, as seen before in the case of GMM, hard to optimize.

We can get around this problem by instead employing the expected complete data log-likelihood

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}}) = \mathbb{E}\Big[\sum_{t=1}^{N} \log \mathcal{P}(\boldsymbol{\xi}_t, \boldsymbol{z}_t, \boldsymbol{u}_t|\boldsymbol{\Theta}) \;\Big|\; \boldsymbol{\xi}, \boldsymbol{\Theta}^{\text{old}} \Big]$$

with $\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})$ the auxiliary function.

Alternating Expectation Conditional Maximization (AECM)

In AECM, each iteration consists of the two cycles:

Cycle 1

Estimate $\boldsymbol{\mu}_i$ and π_i with missing variables $\boldsymbol{z}_t$ based on auxiliary function $\mathcal{Q}_1(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})$.

Cycle 2

Estimate $\boldsymbol{\Lambda}_i$ and $\boldsymbol{\Psi}_i$ with missing variables $\boldsymbol{z}_t$ and $\boldsymbol{u}_t$ based on auxiliary function $\mathcal{Q}_2(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})$.

Each cycle has an E-step and a CM-step.

AECM guarantees convergence of the likelihood to the closest local optimum.

AECM for MFA (UUU model in McNicholas and Murphy, 2008)

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i$$

The auxiliary function $\mathcal{Q}_2(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})$ to estimate $\boldsymbol{\Lambda}_i$ and $\boldsymbol{\Psi}_i$ becomes (see *McNicholas and Murphy (2008)* for details of computation)

$$\mathcal{Q}_2(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}}) = \sum_{t=1}^{N}\sum_{i=1}^{K} h_{t,i}\left(\frac{1}{2}\log|\boldsymbol{\Psi}_i^{-1}| - \text{tr}(\boldsymbol{\Psi}_i^{-1}\boldsymbol{S}_i) + \text{tr}(\boldsymbol{\Psi}_i^{-1}\boldsymbol{\Lambda}_i\boldsymbol{B}_i\boldsymbol{S}_i) - \frac{1}{2}\text{tr}(\boldsymbol{\Lambda}_i^\top\boldsymbol{\Psi}_i^{-1}\boldsymbol{\Lambda}_i\boldsymbol{\theta}_i)\right) + C$$

$$\boldsymbol{x}^\top\boldsymbol{S}\boldsymbol{x} = \text{tr}(\boldsymbol{S}\boldsymbol{x}\boldsymbol{x}^\top)$$

$$\text{with } \boldsymbol{S}_i = \frac{\sum_{t=1}^{N} h_{t,i}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top}{\sum_{t=1}^{N} h_{t,i}}, \quad \boldsymbol{B}_i = \boldsymbol{\Lambda}_i^\top(\boldsymbol{\Lambda}_i\boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i)^{-1}$$

covariance as in GMM

$$\text{and } \boldsymbol{\theta}_i = \boldsymbol{I} - \boldsymbol{B}_i\boldsymbol{\Lambda}_i + \boldsymbol{B}_i\boldsymbol{S}_i\boldsymbol{B}_i^\top$$

AECM for MFA (UUU model in McNicholas and Murphy, 2008)

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i$$

E-step:

$$h_{t,i} = \frac{\pi_i \, \mathcal{N}(\boldsymbol{\xi}_t \mid \boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i)}{\sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{\xi}_t \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k \boldsymbol{\Lambda}_k^\top + \boldsymbol{\Psi}_k)}$$

CM-step:

Same as standard GMM

$$\pi_i \leftarrow \frac{\sum_{t=1}^{N} h_{t,i}}{N}$$

$$\boldsymbol{\mu}_i \leftarrow \frac{\sum_{t=1}^{N} h_{t,i} \, \boldsymbol{\xi}_t}{\sum_{t=1}^{N} h_{t,i}}$$

$$\boldsymbol{\Lambda}_i \leftarrow \boldsymbol{S}_i \boldsymbol{B}_i^\top \overbrace{\left(\boldsymbol{I} - \boldsymbol{B}_i \boldsymbol{\Lambda}_i + \boldsymbol{B}_i \boldsymbol{S}_i \boldsymbol{B}_i^\top\right)^{-1}}^{\boldsymbol{\theta}_i^{-1}}$$

$$\boldsymbol{\Psi}_i \leftarrow \operatorname{diag}\left\{\boldsymbol{S}_i - \boldsymbol{\Lambda}_i \boldsymbol{B}_i \boldsymbol{S}_i\right\}$$

computed with the help of the intermediary variables

$$\boldsymbol{S}_i = \frac{\sum_{t=1}^{N} h_{t,i} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top}{\sum_{t=1}^{N} h_{t,i}}$$

$$\boldsymbol{B}_i = \boldsymbol{\Lambda}_i^\top (\boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i)^{-1}$$

covariance as in GMM

Mixture of probabilistic PCA (MPPCA)

Matlab code: demo_MPPCA01.m

[M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. Neural Computation, 11(2):443–482, 1999]

Mixture of probabilistic PCA (MPPCA)

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i$$

For comparison, the CM-step in MPPCA is given by

$$\tilde{\boldsymbol{\Lambda}}_i \leftarrow \boldsymbol{S}_i \boldsymbol{\Lambda}_i \left(\boldsymbol{I}\sigma_i^2 + \boldsymbol{M}_i^{-1} \boldsymbol{\Lambda}_i^\top \boldsymbol{S}_i \boldsymbol{\Lambda}_i \right)^{-1}$$

$$\boldsymbol{\Psi}_i \leftarrow \boldsymbol{I}\sigma_i^2$$

computed with the help of the intermediary variables

covariance as in GMM

$$\boldsymbol{S}_i = \frac{\sum_{t=1}^N h_{t,i} (\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top}{\sum_{t=1}^N h_{t,i}}$$

$$\boldsymbol{M}_i = \boldsymbol{\Lambda}_i^\top \boldsymbol{\Lambda}_i + \boldsymbol{I}\sigma_i^2$$

$$\sigma_i^2 = \frac{1}{D} \mathrm{tr}\left(\boldsymbol{S}_i - \boldsymbol{S}_i \boldsymbol{\Lambda}_i \boldsymbol{M}_i^{-1} \tilde{\boldsymbol{\Lambda}}_i^\top \right)$$

where $\boldsymbol{\Lambda}_i$ is replaced by $\tilde{\boldsymbol{\Lambda}}_i$ at each iteration.

A taxonomy of parsimonious GMMs

D in the slides of this lecture

Model name	Cov. structure	Nb. of parameters	$K=4, d=3$ $p=100$
UUUU - UUU	$S_k = \Lambda_k \Lambda_k^t + \Psi_k$	$(K-1) + Kp + Kd[p-(d-1)/2] + Kp$	1991
UUCU -	$S_k = \Lambda_k \Lambda_k^t + \omega_k \Delta_k$	$(K-1) + Kp + Kd[p-(d-1)/2] + [1 + K(p-1)]$	1988
UCUU -	$S_k = \Lambda_k \Lambda_k^t + \omega_k \Delta$	$(K-1) + Kp + Kd[p-(d-1)/2] + [K + (p-1)]$	1694
UCCU - UCU	$S_k = \Lambda_k \Lambda_k^t + \Psi$	$(K-1) + Kp + Kd[p-(d-1)/2] + p$	1691
UCUC - UUC	$S_k = \Lambda_k \Lambda_k^t + \psi_k \mathbf{I}_p$	$(K-1) + Kp + Kd[p-(d-1)/2] + K$	1595
UCCC - UCC	$S_k = \Lambda_k \Lambda_k^t + \psi \mathbf{I}_p$	$(K-1) + Kp + Kd[p-(d-1)/2] + 1$	1592
CUUU - CUU	$S_k = \Lambda \Lambda^t + \Psi_k$	$(K-1) + Kp + d[p-(d-1)/2] + Kp$	1100
CUCU -	$S_k = \Lambda \Lambda^t + \omega \Delta_k$	$(K-1) + Kp + d[p-(d-1)/2] + [1 + K(p-1)]$	1097
CCUU -	$S_k = \Lambda \Lambda^t + \omega_k \Delta$	$(K-1) + Kp + d[p-(d-1)/2] + [K + (p-1)]$	803
CCCU - CCU	$S_k = \Lambda \Lambda^t + \Psi$	$(K-1) + Kp + d[p-(d-1)/2] + p$	800
CCUC - CUC	$S_k = \Lambda \Lambda^t + \psi_k \mathbf{I}_p$	$(K-1) + Kp + d[p-(d-1)/2] + K$	704
CCCC - CCC	$S_k = \Lambda \Lambda^t + \psi \mathbf{I}_p$	$(K-1) + Kp + d[p-(d-1)/2] + 1$	701

where $\omega_k \in \mathbb{R}^+$ and $|\Delta_k| = 1$.

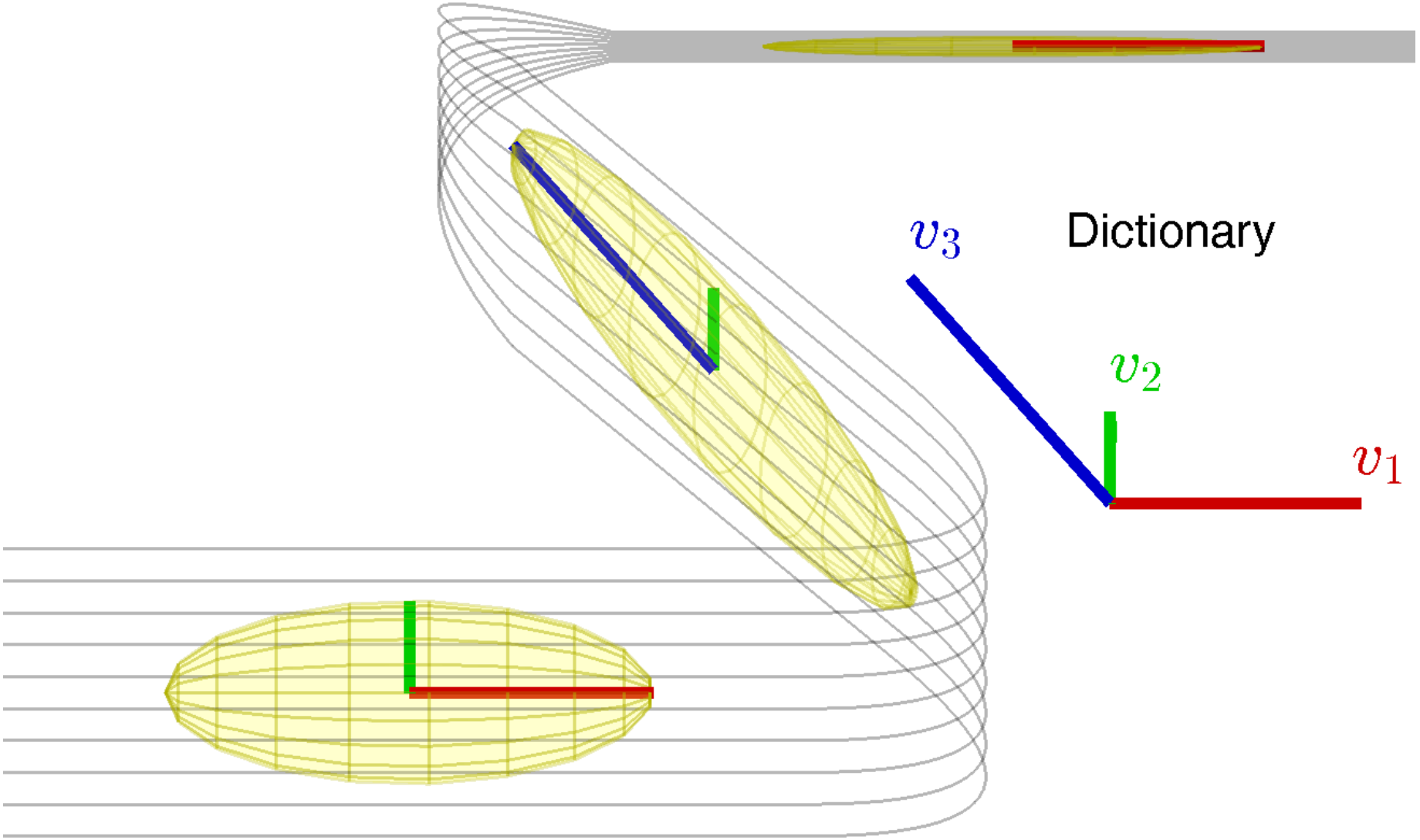
[C. Bouveyron and C. Brunet. Model-based clustering of high-dimensional data: A review. Computational Statistics and Data Analysis, 71:52–78, March 2014]

GMM with semi-tied covariance matrices

Matlab code: demo_semitiedGMM01.m

[M. J. F. Gales. Semi-tied covariance matrices for hidden Markov models. IEEE Trans. on Speech and Audio Processing, 7(3):272–281, 1999]

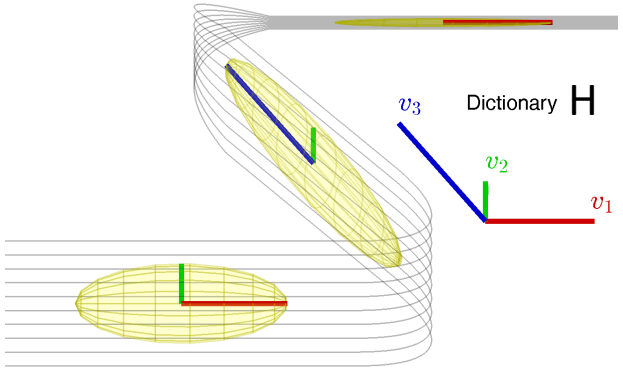
Sharing of parameters in mixture models



GMM with semi-tied covariance matrices

The covariances share the same set of parameters for the latent feature space, where each covariance is composed of a common latent feature matrix $\boldsymbol{H} \in \mathbb{R}^{D \times D}$ and a component-specific diagonal covariance $\boldsymbol{\Sigma}_i^{\text{diag}} \in \mathbb{R}^{D \times D}$ with

$$\boldsymbol{\Sigma}_i = \boldsymbol{H}\, \boldsymbol{\Sigma}_i^{\text{diag}}\, \boldsymbol{H}^\top$$



The latent feature matrix encodes the most relevant synergistic directions/basis vectors that are shared among all components, with the diagonal matrix representing the convex combination of basis vectors.

In other words, the aim is to find a global linear transformation of the data such that the transformed data can be modeled by a mixture of diagonal covariance matrices only.

GMM with semi-tied covariance matrices

$$\boldsymbol{\Sigma}_i = \boldsymbol{H}\boldsymbol{\Sigma}_i^{\text{diag}}\boldsymbol{H}^\top$$

The parameters of a GMM with semi-tied covariances are $\boldsymbol{\Theta}^{\text{tiedGMM}} = \{\boldsymbol{H}, \{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i^{\text{diag}}\}_{i=1}^K\}$. By setting $\boldsymbol{B} = \boldsymbol{H}^{-1}$, we have

$$\log\left|\boldsymbol{B}^{-1}\boldsymbol{\Sigma}_i^{\text{diag}}\boldsymbol{B}^{-\top}\right| = \log\left(\frac{\left|\boldsymbol{\Sigma}_i^{\text{diag}}\right|}{|\boldsymbol{B}|^2}\right) = \log\left|\boldsymbol{\Sigma}_i^{\text{diag}}\right| - 2\log|\boldsymbol{B}|$$

and the auxiliary function $\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})$ of the standard GMM can be rewritten as

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}}) = \sum_{t=1}^{N}\sum_{i=1}^{K} h_{t,i}\left(\log(\pi_i) - \frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i) - \frac{D}{2}\log(2\pi)\right)$$

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}}) = \sum_{t=1}^{N}\sum_{i=1}^{K} h_{t,i}\left(\log(\pi_i) + \log|\boldsymbol{B}| - \frac{1}{2}\log\left|\boldsymbol{\Sigma}_i^{\text{diag}}\right| - \frac{1}{2}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^\top\boldsymbol{B}^\top\boldsymbol{\Sigma}_i^{(\text{diag})-1}\boldsymbol{B}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i) - \frac{D}{2}\log(2\pi)\right).$$

GMM with semi-tied covariance matrices

$$\boldsymbol{\Sigma}_i = \boldsymbol{B}^{-1}\boldsymbol{\Sigma}_i^{\text{diag}}\boldsymbol{B}^{-\top}$$

Setting $\frac{\partial \mathcal{Q}(\boldsymbol{\Theta},\boldsymbol{\Theta}^{\text{old}})}{\partial \boldsymbol{B}}$ and $\frac{\partial \mathcal{Q}(\boldsymbol{\Theta},\boldsymbol{\Theta}^{\text{old}})}{\partial \boldsymbol{\Sigma}_i^{\text{diag}}}$ equal to 0, and solving for $\boldsymbol{B}$ and $\boldsymbol{\Sigma}_i^{\text{diag}}$ results in an expectation-maximization procedure to compute the maximum likelihood estimate of the parameters.

Following this, we get a row-by-row optimisation of $\boldsymbol{B}$, with $\boldsymbol{b}_d$ (d-th row of $\boldsymbol{B}$) related to all other rows by the cofactor of $\boldsymbol{B}$

$$\boldsymbol{B}^{-1} = \frac{\text{cof}(\boldsymbol{B})^\top}{|\boldsymbol{B}|}$$

$$\iff \begin{bmatrix} \boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{c}_D \end{bmatrix} = |\boldsymbol{B}|\,(\boldsymbol{B}^\top)^{-1}$$

$$\boldsymbol{b}_d = \boldsymbol{c}_d \boldsymbol{G}_d^{-1} \sqrt{\frac{\sum_{t=1}^{T}\sum_{i=1}^{K} h_{t,i}}{\boldsymbol{c}_d \boldsymbol{G}_d^{-1} \boldsymbol{c}_d^\top}}$$

where $\boldsymbol{c}_d$ is the d-th row of cofactors of $\boldsymbol{B}$ recomputed after each update of $\boldsymbol{b}_d$, and

$$\boldsymbol{G}_d = \sum_{i=1}^{K} \frac{1}{\sum_{i,d}^{\text{diag}}} \boldsymbol{S}_i \sum_{t=1}^{T} h_{t,i}$$

GMM with semi-tied covariance matrices

$$\boldsymbol{\Sigma}_i = \boldsymbol{B}^{-1}\boldsymbol{\Sigma}_i^{\text{diag}}\boldsymbol{B}^{-\top}$$

$\Sigma_{i,d}^{\text{diag}}$ is the d-th diagonal element of the i-th Gaussian, and $\boldsymbol{S}_i$ is the full sample covariance matrix given by

covariance as in GMM

$$\boldsymbol{S}_i = \frac{\sum_{t=1}^{T} h_{t,i}\,(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)(\boldsymbol{\xi}_t - \boldsymbol{\mu}_i)^{\top}}{\sum_{t=1}^{T} h_{t,i}}$$

The corresponding maximum likelihood estimate of $\boldsymbol{\Sigma}_i^{\text{diag}}$ is computed as

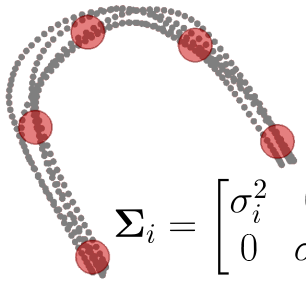
$$\boldsymbol{\Sigma}_i^{\text{diag}} = \text{diag}\left\{\boldsymbol{B}\boldsymbol{S}_i\boldsymbol{B}^{\top}\right\}$$

Note the variational nature of optimisation where the current estimate of $\boldsymbol{\Sigma}_i^{\text{diag}}$ is dependent on $\boldsymbol{B}$ and vice versa.

Both $\boldsymbol{B}$ and $\boldsymbol{\Sigma}_i^{\text{diag}}$ are iteratively improved in each EM step and the likelihood is guaranteed to increase at each step until convergence.

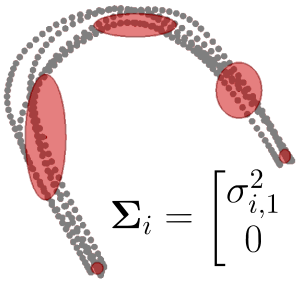
Summary of relevant covariance structures

Isotropic



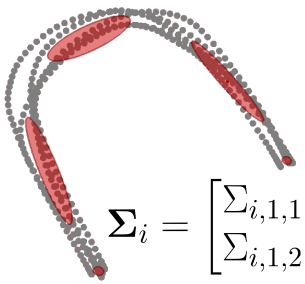
$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{bmatrix}$$

Diagonal



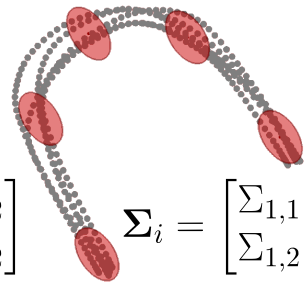
$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_{i,1}^2 & 0 \\ 0 & \sigma_{i,2}^2 \end{bmatrix}$$

Full

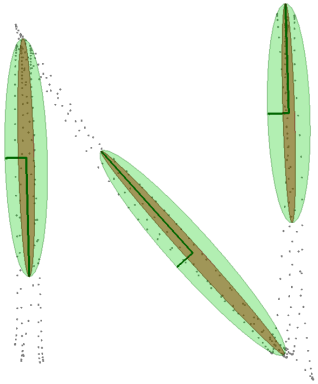


$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \Sigma_{i,1,1} & \Sigma_{i,1,2} \\ \Sigma_{i,1,2} & \Sigma_{i,2,2} \end{bmatrix}$$

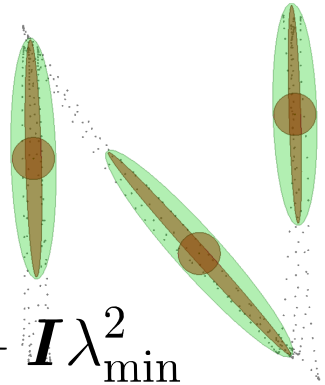
Tied



$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{1,2} & \Sigma_{2,2} \end{bmatrix}$$



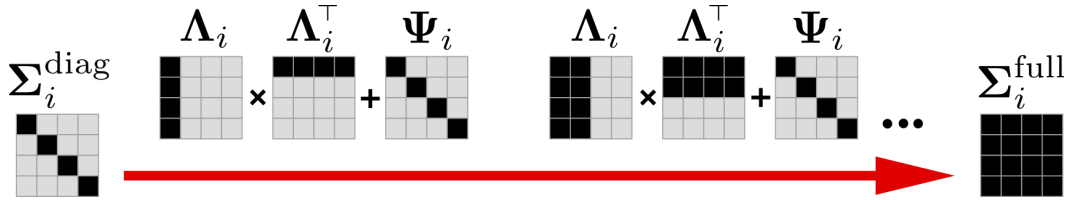
$$\boldsymbol{\Sigma}_i \leftarrow \boldsymbol{V}_i \tilde{\boldsymbol{D}}_i \boldsymbol{V}_i^\top \text{ with}$$
$$\tilde{\boldsymbol{D}}_i = \text{diag}(\tilde{\lambda}_{i,1}^2, \ldots, \tilde{\lambda}_{i,D}^2) \text{ and}$$
$$\tilde{\lambda}_{i,j}^2 = \max(\lambda_{i,j}^2, \lambda_{\min}^2)$$



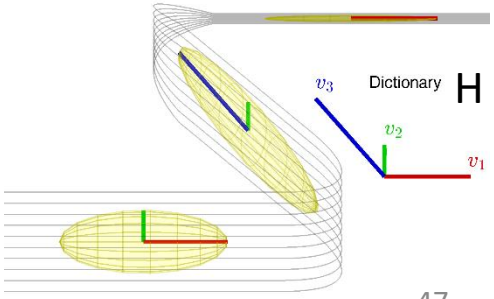
$$\boldsymbol{\Sigma}_i \leftarrow \boldsymbol{\Sigma}_i + \boldsymbol{I}\lambda_{\min}^2$$

$$\text{MFA: } \boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top + \boldsymbol{\Psi}_i$$

$$\text{MPPCA: } \boldsymbol{\Sigma}_i = \boldsymbol{\Lambda}_i^\top \boldsymbol{\Lambda}_i + \boldsymbol{I}\sigma_i^2$$



$$\boldsymbol{\Sigma}_i = \boldsymbol{H}\boldsymbol{\Sigma}_i^{\text{diag}}\boldsymbol{H}^\top$$



Main references

Parsimonious GMM

C. Bouveyron and C. Brunet. Model-based clustering of high-dimensional data: A review. Computational Statistics and Data Analysis, 71:52–78, March 2014

P. D. McNicholas and T. B. Murphy. Parsimonious Gaussian mixture models. Statistics and Computing, 18(3):285–296, September 2008

MFA

G. J. McLachlan, D. Peel, and R. W. Bean. Modelling high-dimensional data by mixtures of factor analyzers. Computational Statistics and Data Analysis, 41(3-4):379–388, 2003

G. E. Hinton, P. Dayan, and M. Revow. Modeling the manifolds of images of handwritten digits. IEEE Trans. on Neural Networks, 8(1):65–74, 1997

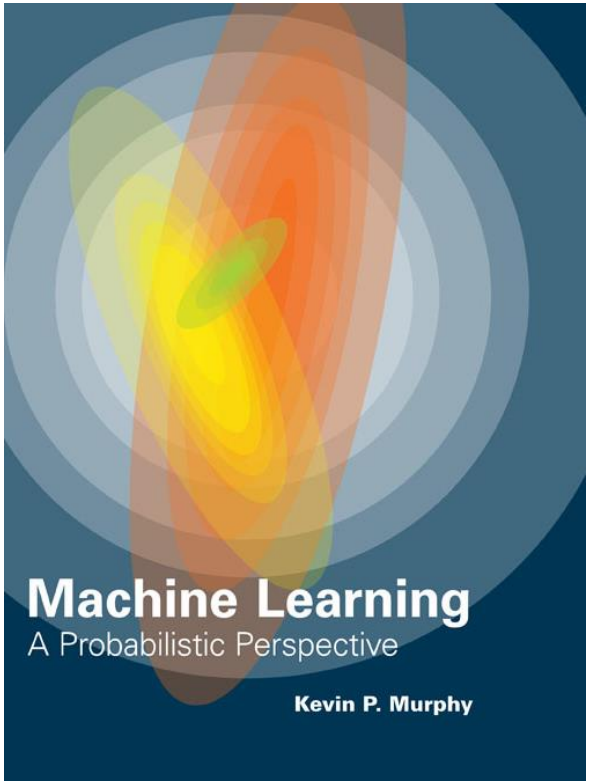
MPPCA

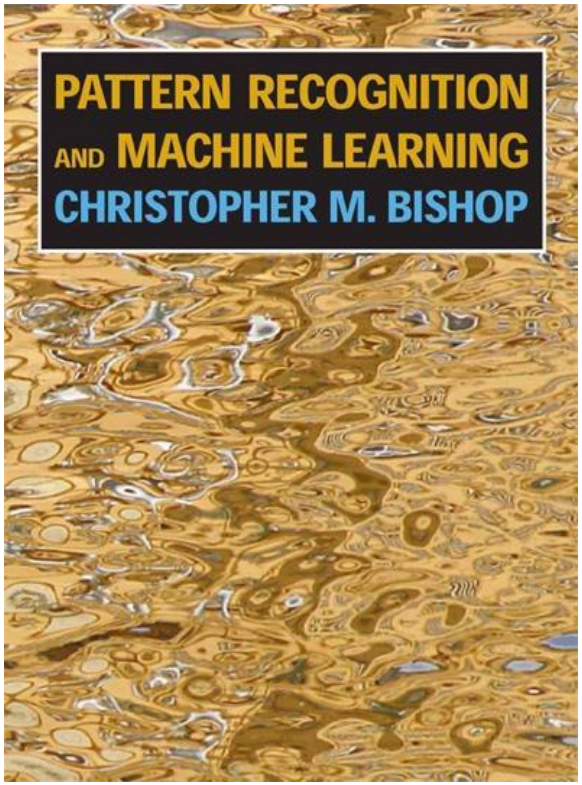
M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. Neural Computation, 11(2):443–482, 1999

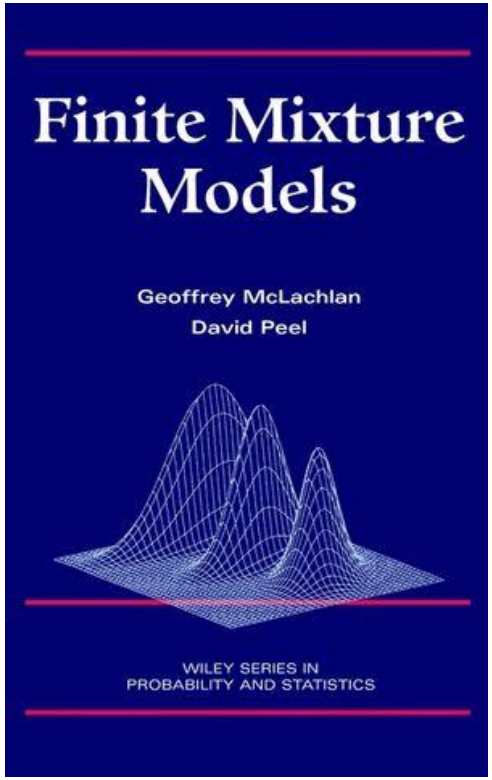
GMM with semi-tied covariances

M. J. F. Gales. Semi-tied covariance matrices for hidden Markov models. IEEE Trans. on Speech and Audio Processing, 7(3):272–281, 1999

General textbooks







Advanced related research topics

(not covered in the course)

Coordinated MFA by using common factor loadings

J. Baek, G. J. McLachlan, and L. K. Flack. Mixtures of factor analyzers with common factor loadings: Applications to the clustering and visualization of high-dimensional data. IEEE Trans. Pattern Anal. Mach. Intell., 32(7):1298–1309, 2010

J. Verbeek. Learning nonlinear image manifolds by global alignment of local linear models. IEEE Trans. on Pattern Analysis & Machine Intelligence, 28(8):1236–1250, August 2006

Estimation of K and d_k in MFA with Bayesian nonparametrics

Y. Wang and J. Zhu. DP-space: Bayesian nonparametric subspace clustering with small-variance asymptotics. In Proc. Intl Conf. on Machine Learning (ICML), pages 1–9, Lille, France, 2015

Online parameters estimation in MPPCA

A. Bellas, C. Bouveyron, M. Cottrell, and J. Lacaille. Model-based clustering of high-dimensional data streams with online mixture of probabilistic PCA. Advances in Data Analysis and Classification, 7(3):281–300, 2013

Advanced related research topics

(not covered in the course)

Sparse subspace clustering with L_1 regularization

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. Journal of Computational and Graphical Statistics, 15(2):265–286, 2006

Y. Guan and J. G. Dy. Sparse probabilistic principal component analysis. In Intl Conf. on Artificial Intelligence and Statistics, pages 185–192, 2009

Deep MFA

Y. Tang, R. Salakhutdinov, and G. Hinton. Deep mixtures of factor analysers. In Proc. Intl Conf. on Machine Learning (ICML), Edinburgh, Scotland, 2012

Mixture of tensor analyzers (MTA)

Y. Tang, R. Salakhutdinov, and G. Hinton. Tensor analyzers. In Proc. Intl Conf. on Machine Learning (ICML), Atlanta, USA, 2013