EE613
Machine Learning for Engineers

SUBSPACE CLUSTERING

Sylvain Calinon
Robot Learning & Interaction Group
Idiap Research Institute
Oct. 21, 2015
SUBSPACE CLUSTERING (Oct. 21)
HIDDEN MARKOV MODELS (Oct. 28)
LINEAR REGRESSION (Nov. 4)
NONLINEAR REGRESSION (Nov. 11)

Time series analysis and synthesis
Multivariate data
• High-dimensional data clustering (HDDC)

• Mixture of factor analyzers (MFA)

• Mixture of probabilistic principal component analyzers (MPPCA)

• GMM with semi-tied covariance matrices
Introduction

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

Considering the two problems separately (one after the other) can be inefficient and produce poor local optima, especially when data of high dimensionality are considered.

- $K$ clusters
- $N$ datapoints
- $D$ dimensions (original space)
- $d$ dimensions (latent space)
Example of application: Whole body motion

→ 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

Walking

Walking

Running

Walking

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

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

For a dataset \( \{ \xi_t \}_{t=1}^N \) with \( \xi_t \in \mathbb{R}^D \), 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 \( \Sigma_i \in \mathbb{R}^{D \times D} \) because the number of parameters to be estimated grows quadratically with \( D \).
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?
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 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).
Gaussian Mixture Model (GMM)

\[ \mathcal{P}(\xi_t) = \sum_{i=1}^{K} \pi_i \mathcal{N}(\xi_t | \mu_i, \Sigma_i) \]

\[ \mathcal{N}(\xi_t | \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left( -\frac{1}{2} (\xi_t - \mu_i)^\top \Sigma_i^{-1} (\xi_t - \mu_i) \right) \]

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

\[ \pi_i \in \mathbb{R} \quad \text{Mixing coefficient} \]

\[ \mu_i \in \mathbb{R}^D \quad \text{Center (or mean)} \]

\[ \Sigma_i \in \mathbb{R}^{D \times D} \quad \text{Covariance matrix} \]

Equidensity contour of one standard deviation

Parameters \( \Theta^{\text{GMM}} = \{ \pi_i, \mu_i, \Sigma_i \}_{i=1}^{K} \)
Covariance structures in GMM

Isotropic

$\Sigma_i = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{bmatrix}$

Diagonal

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

Full

$\Sigma_i = \begin{bmatrix} \Sigma_{i,1,1} & \Sigma_{i,1,2} \\ \Sigma_{i,1,2} & \Sigma_{i,2,2} \end{bmatrix}$

Tied

$\Sigma_i = \begin{bmatrix} \Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{1,2} & \Sigma_{2,2} \end{bmatrix}$
Multivariate normal distribution - Stochastic sampling

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

$$
\Sigma = V D V^\top = \sum_{j=1}^{D} \lambda_j v_j v_j^\top
$$

with $V = [v_1, v_2, \ldots, v_D]$ and $D = \begin{bmatrix}
\lambda_1^2 & 0 & \cdots & 0 \\
0 & \lambda_2^2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_D^2
\end{bmatrix}$

By using this notation, datapoints can be stochastically generated with

$$
\xi \sim \mathcal{N}(\mu, \Sigma) \iff \xi \sim \mu + V D^{\frac{1}{2}} \mathcal{N}(0, I)
$$
Multivariate normal distribution - Equidensity contours

If $\xi \in \mathbb{R}^D \sim \mathcal{N}(\mu, \Sigma)$, then $(\xi - \mu)^\top \Sigma^{-1}(\xi - \mu) \sim \chi^2_D$

(Chi-Square distribution of dimension $D$)

Contour of 2.41 $\Sigma$

- 702 datapoints (70% of total number)

Contour of 3.22 $\Sigma$

- 797 datapoints (80% of total number)

Contour of 4.60 $\Sigma$

- 895 datapoints (90% of total number)

<table>
<thead>
<tr>
<th>Degrees of freedom (df)</th>
<th>$\chi^2$ value$^{[18]}$</th>
<th>Table: Wikipedia</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.004 0.02 0.06 0.15 0.46 1.07 1.64 2.71 3.84 6.64 10.83</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.10 0.21 0.45 0.71 1.39 <strong>2.41</strong> 3.22 <strong>4.60</strong> 5.99 9.21 13.82</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.35 0.58 1.01 1.42 2.37 3.66 4.64 6.25 7.82 11.34 16.27</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.71 1.06 1.65 2.20 3.36 4.88 5.99 7.78 9.49 13.28 18.47</td>
<td></td>
</tr>
</tbody>
</table>

P value (Probability)

- 0.95 0.90 0.80 0.70 0.50 **0.30** **0.20** **0.10** 0.05 0.01 0.001
**Expectation-maximization (EM)**

Each datapoint $\xi_t$ is associated with a hidden/missing variable $z_t$. The goal is to maximize the log-likelihood of the observed data

$$
\mathcal{L}(\Theta) = \sum_{t=1}^{N} \log \mathcal{P}(\xi_t | \Theta) = \sum_{t=1}^{N} \log \left( \sum_{z_t} \mathcal{P}(\xi_t, z_t | \Theta) \right)
$$

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

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

$$
\mathcal{L}'(\Theta) = \sum_{t=1}^{N} \log \mathcal{P}(\xi_t, z_t | \Theta)
$$

This also cannot be computed directly because $z_t$ is unknown, but we can define the expected complete data log-likelihood

$$
Q(\Theta, \Theta^{\text{old}}) = \mathbb{E} \left[ \mathcal{L}'(\Theta) \mid \xi, \Theta^{\text{old}} \right]
$$

where $Q(\Theta, \Theta^{\text{old}})$ is called the auxiliary function.
Expectation-maximization (EM)

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

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

The *M-step* then optimizes $Q$ with respect to $\Theta$. 

---

**Diagram:**

Initial guess $\rightarrow$ E-step $\rightarrow$ Converge? $\rightarrow$ Stop $\rightarrow$ M-step $\rightarrow$ Initial guess
When applied to GMM, the auxiliary function $Q(\Theta, \Theta^{old})$ takes the form

$$Q(\Theta, \Theta^{old}) = \mathbb{E} \left[ \sum_{t=1}^{N} \log \mathcal{P}(\xi_t, z_t | \Theta) \mid \xi, \Theta^{old} \right]$$

$$= \sum_{t=1}^{N} \mathbb{E} \left[ \log \left( \prod_{i=1}^{K} (\pi_i \mathcal{N}(\xi_t | \mu_i, \Sigma_i))^{z_{t,i}} \right) \mid \xi, \Theta^{old} \right]$$

$$= \sum_{t=1}^{N} \sum_{i=1}^{K} \mathbb{E} \left[ \log \left( (\pi_i \mathcal{N}(\xi_t | \mu_i, \Sigma_i))^{z_{t,i}} \right) \mid \xi, \Theta^{old} \right]$$

where $z_{t,i} = 1$ if $\xi_t$ is part of cluster $i$. It is 0 otherwise.

$$= \sum_{t=1}^{N} \sum_{i=1}^{K} \mathbb{E} \left[ z_{t,i} \mid \xi, \Theta^{old} \right] \log \left( \pi_i \mathcal{N}(\xi_t | \mu_i, \Sigma_i) \right)$$

$$= \sum_{t=1}^{N} \sum_{i=1}^{K} h_{t,i} \left( \log(\pi_i) + \log \left( \mathcal{N}(\xi_t | \mu_i, \Sigma_i) \right) \right)$$

where $h_{t,i}$ is the responsibility that cluster $i$ takes for datapoint $\xi_t$. 
EM for GMM

Setting

\[
\frac{\partial Q(\Theta, \Theta^{old})}{\partial \pi_i} = 0 \quad \frac{\partial Q(\Theta, \Theta^{old})}{\partial \mu_i} = 0 \quad \frac{\partial Q(\Theta, \Theta^{old})}{\partial \Sigma_i} = 0
\]

and solving for \( \pi_i, \mu_i \) and \( \Sigma_i \) results in an EM procedure to compute the maximum likelihood estimate of the parameters.
EM for GMM

By using the linear algebra relations
\[
\frac{\partial}{\partial A} \log |A| = (A^T)^{-1} \quad \frac{\partial}{\partial A} x^T A x = x x^T \quad \frac{\partial}{\partial x} x^T A x = (A + A^T)x
\]

\(= 2Ax\) if \(A\) symmetric

and the derivation chain rule, we obtain

\[
\frac{\partial Q(\Theta, \Theta^{\text{old}})}{\partial \mu_i} = \frac{1}{2} \sum_{t=1}^{N} h_{t,i} 2\Sigma_i^{-1}(\xi_t - \mu_i) = \Sigma_i^{-1} \sum_{t=1}^{N} h_{t,i} (\xi_t - \mu_i) = 0
\]

\[\iff \mu_i = \frac{\sum_{t=1}^{N} h_{t,i} \xi_t}{\sum_{t=1}^{N} h_{t,i}}\]

\[
\frac{\partial Q(\Theta, \Theta^{\text{old}})}{\partial \Sigma_i} = \frac{1}{2} \Sigma_i \sum_{t=1}^{N} h_{t,i} - \frac{1}{2} \sum_{t=1}^{N} h_{t,i} (\xi_t - \mu_i)(\xi_t - \mu_i)^T = 0
\]

\[\iff \Sigma_i = \frac{\sum_{t=1}^{N} h_{t,i} (\xi_t - \mu_i)(\xi_t - \mu_i)^T}{\sum_{t=1}^{N} h_{t,i}}\]
EM for GMM

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

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

The sum over $K$ of the above relation provides

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

that can be reintroduced in the equation to find

$$\frac{1}{\pi_i} \sum_{t=1}^{N} h_{t,i} - N = 0 \iff \pi_i = \frac{\sum_{t=1}^{N} h_{t,i}}{N}$$
EM for GMM: Resulting procedure

$E$-step:

$$h_{t,i} = \frac{\pi_i \mathcal{N}(\xi_t | \mu_i, \Sigma_i)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\xi_t | \mu_k, \Sigma_k)}$$

$M$-step:

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

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

$$\Sigma_i \leftarrow \frac{\sum_{t=1}^{N} h_{t,i} (\xi_t - \mu_i)(\xi_t - \mu_i)^\top}{\sum_{t=1}^{N} h_{t,i}}$$

These results can be intuitively re-interpreted in terms of normalized counts, but EM provides a systematic technique to derive the procedure.

→ Weighted averages taking into account the responsibility of each datapoint in each cluster.
EM for GMM

M-step
EM for GMM
EM will improve the likelihood at each iteration, but it can get trapped into poor local optima in complex 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...

... which does not mean that moment-based approaches are deprecated!

They are actually today regaining momentum with new developments related to spectral decomposition.
High-dimensional data clustering (HDDC)

demo_HDDC01.m

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:

$$\Sigma_i \leftarrow V_i \tilde{D}_i V_i^\top$$

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

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

Tikhonov regularization with diagonal isotropic covariance:

$$\Sigma_i \leftarrow \Sigma_i + I \lambda_{\min}$$
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

\[
\bar{\lambda}_i = \frac{1}{D-d_i} \sum_{k=1}^{D} \lambda_{i,k} = \frac{1}{D-d_i} \left( \text{tr}(\Sigma_i) - \sum_{k=1}^{d_i} \lambda_{i,k} \right)
\]

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)

demo_MFA01.m

Mixture of factor analyzers (MFA)

MFA assumes for each covariance $i$ a structure of the form

$$
\Sigma_i = \Lambda_i \Lambda_i^\top + \Psi_i
$$

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

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 $\Psi_i = I \sigma_i^2$. 
Mixture of factor analyzers (MFA)

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

$$\xi = \Lambda_i u + \mu_i + \epsilon_i$$

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

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 $\xi$. 
Mixture of factor analyzers (MFA)

For MFA with covariance structure $\Sigma_i = \Lambda_i \Lambda_i^\top + \Psi$
It follows from this model that the marginal distribution of $\xi$ for the $i$-th component is
\[
\xi \sim \mathcal{N}(\mu_i, \Lambda_i \Lambda_i^T + \Psi_i)
\]
and the joint distribution of $\xi$ and $u$ is
\[
\begin{bmatrix} \xi \\ u \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_i \\ 0 \end{bmatrix}, \begin{bmatrix} \Lambda_i \Lambda_i^T + \Psi_i & \Lambda_i \\ \Lambda_i^T & I \end{bmatrix}\right)
\]
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
\[
u | \xi \sim \mathcal{N}\left(B_i(\mu_i - \xi), I - B_i \Lambda_i \right) \quad \text{with} \quad B_i = \Lambda_i^T(\Lambda_i \Lambda_i^T + \Psi_i)^{-1}
\]
Mixture of factor analyzers (MFA)

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

\[ \mathbb{E}(uu^\top | \xi) = \text{cov}(u | \xi) + \mathbb{E}(u | \xi) \mathbb{E}(u | \xi)^\top = I - B_i \Lambda_i + B_i (\mu_i - \xi)(\mu_i - \xi)^\top B_i^\top \]

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

\[ \Theta^{\text{MFA}} = \{\pi_i, \mu_i, \Lambda_i, \Psi_i\}_{i=1}^K \]

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

\[ \Theta^{\text{FA}} = \{\mu, \Lambda, \Psi\} \]
Estimation of parameters in MFA

In the case of MFA, it is considered that each datapoint $\xi_t$ is associated with hidden variables $z_t$ and $u_t$, and the goal is to maximize

$$L(\Theta) = \sum_{t=1}^{N} \log P(\xi_t | \Theta) = \sum_{t=1}^{N} \log \left( \sum_{z_t} P(\xi_t, z_t, u_t | \Theta) \right)$$

which is as seen before hard to optimize.

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

$$L'(\Theta) = \sum_{t=1}^{N} \log P(\xi_t, z_t, u_t | \Theta)$$

This also cannot be computed directly because $z_t$ and $u_t$ are unknown, but we can define the expected complete data log-likelihood

$$Q(\Theta, \Theta^{\text{old}}) = \mathbb{E} \left[ L'(\Theta) \mid \xi, \Theta^{\text{old}} \right]$$

with $Q(\Theta, \Theta^{\text{old}})$ the auxiliary function.
Alternating Expectation Conditional Maximization (AECM)

In AECM, each iteration consists of the two cycles:

**Cycle 1**
Estimate $\mu_i$ and $\pi_i$ with missing variables $z_t$ based on auxiliary function $Q_1(\Theta, \Theta^{\text{old}})$.

**Cycle 2**
Estimate $\Lambda_i$ and $\Psi_i$ with missing variables $z_t$ and $u_t$ based on auxiliary function $Q_2(\Theta, \Theta^{\text{old}})$.

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

AECM guarantees convergence of the likelihood to the closest local optimum.
The auxiliary function $Q_2(\Theta, \Theta^{\text{old}})$ to estimate $\Lambda_i$ and $\Psi_i$ becomes (see *McNicholas and Murphy (2008)* for details of computation)

$$Q_2(\Theta, \Theta^{\text{old}}) = \sum_{t=1}^{N} \sum_{i=1}^{K} h_{t,i} \left( \frac{1}{2} \log |\Psi_i^{-1}| - \text{tr}(\Psi_i^{-1} S_i) \right) + \text{tr}(\Psi_i^{-1} \Lambda_i B_i S_i) - \frac{1}{2} \text{tr}(\Lambda_i^T \Psi_i^{-1} \Lambda_i \theta_i) + C$$

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

with $S_i = \frac{\sum_{t=1}^{N} h_{t,i} (\xi_t - \mu_i)(\xi_t - \mu_i)^T}{\sum_{t=1}^{N} h_{t,i}}$, $B_i = \Lambda_i^T (\Lambda_i \Lambda_i^T + \Psi_i)^{-1}$

and $\theta_i = I - B_i \Lambda_i + B_i S_i B_i^T$
AECM for MFA (UUU model in McNicholas and Murphy, 2008)

**E-step:**

\[ h_{t,i} = \frac{\pi_i \mathcal{N}(\xi_t | \mu_i, \Lambda_i \Lambda_i^T + \Psi_i)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\xi_t | \mu_k, \Lambda_k \Lambda_k^T + \Psi_k)} \]

**CM-step:**

Same as standard GMM

- \( \pi_i \leftarrow \frac{\sum_{t=1}^{N} h_{t,i}}{N} \)
- \( \mu_i \leftarrow \frac{\sum_{t=1}^{N} h_{t,i} \xi_t}{\sum_{t=1}^{N} h_{t,i}} \)
- \( \Lambda_i \leftarrow S_i B_i^T (I - B_i \Lambda_i + B_i S_i B_i^T)^{-1} \)
- \( \Psi_i \leftarrow \text{diag} \{S_i - \Lambda_i B_i S_i\} \)

computed with the help of the intermediary variables

\[ S_i = \frac{\sum_{t=1}^{N} h_{t,i} (\xi_t - \mu_i)(\xi_t - \mu_i)^T}{\sum_{t=1}^{N} h_{t,i}} \]

\[ B_i = \Lambda_i^T (\Lambda_i \Lambda_i^T + \Psi_i)^{-1} \]
Mixture of probabilistic PCA (MPPCA)

demo_MPPCA01.m

Mixture of probabilistic PCA (MPPCA)

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

\[
\tilde{\Lambda}_i \leftarrow S_i \Lambda_i \left( I \sigma_i^2 + M_i^{-1} \Lambda_i^T S_i \Lambda_i \right)^{-1}
\]

\[
\Psi_i \leftarrow I \sigma_i^2
\]

computed with the help of the intermediary variables

\[
S_i = \frac{\sum_{t=1}^{N} h_{t,i}(\xi_t - \mu_i)(\xi_t - \mu_i)^T}{\sum_{t=1}^{N} h_{t,i}}
\]

\[
M_i = \Lambda_i^T \Lambda_i + I \sigma_i^2
\]

\[
\sigma_i^2 = \frac{1}{D} \text{tr}(S_i - S_i \Lambda_i M_i^{-1} \tilde{\Lambda}_i^T)
\]

where \( \Lambda_i \) is replaced by \( \tilde{\Lambda}_i \) at each iteration.
### A taxonomy of parsimonious GMM

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

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

Sharing of parameters in mixture models
GMM/HMM with semi-tied covariance matrices

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 $H \in \mathbb{R}^{D \times D}$ and a component-specific diagonal covariance $\Sigma_{i, \text{diag}} \in \mathbb{R}^{D \times D}$ with

$$\Sigma_{i} = H \Sigma_{i, \text{diag}} H^\top$$

The latent feature matrix encodes the most relevant synergistic directions/basis vectors that are shared across 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.
The parameters of a GMM with semi-tied covariances are \( \Theta^{\text{tiedGMM}} = \{H, \{\pi_i, \mu_i, \Sigma_i^{\text{diag}}\}_{i=1}^K\} \). By setting \( B = H^{-1} \), we have

\[
\log |B^{-1} \Sigma_i^{\text{diag}} B^{-\top}| = \log \left( \frac{|\Sigma_i^{\text{diag}}|}{|B|^2} \right) = \log |\Sigma_i^{\text{diag}}| - 2 \log |B|
\]

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

\[
Q(\Theta, \Theta^{\text{old}}) = \sum_{t=1}^N \sum_{i=1}^K h_{t,i} \left( \log(\pi_i) + \log |B| - \frac{1}{2} \log |\Sigma_i^{\text{diag}}| \\
- \frac{1}{2} (\xi_t - \mu_i) \top B \Sigma_i^{(\text{diag})^{-1}} B (\xi_t - \mu_i) - \frac{D}{2} \log(2\pi) \right).
\]
GMM with semi-tied covariance matrices

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

Following this, we get a row-by-row optimisation of \( B \), with \( b_d \) (\( d \)-th row of \( B \)) related to all other rows by the cofactor of \( B \)

\[
B^{-1} = \frac{\text{cof}(B)^\top}{|B|}
\]

\[
\begin{bmatrix} c_1 \\ \vdots \\ c_D \end{bmatrix} = |B| (B^\top)^{-1} \quad b_d = c_d G_d^{-1} \sqrt{\sum_{t=1}^{T} \sum_{i=1}^{K} h_{t,i} \over c_d G_d^{-1} c_d^\top}
\]

where \( c_d \) is the \( d \)-th row of cofactors of \( B \) recomputed after each update of \( b_d \), and

\[
G_d = \sum_{i=1}^{K} \frac{1}{\sum_{i,d}^{\text{diag}} S_i} \sum_{t=1}^{T} h_{t,i}
\]
GMM with semi-tied covariance matrices

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

$$S_i = \frac{\sum_{t=1}^{T} h_{t,i} (\xi_t - \mu_i)(\xi_t - \mu_i)^\top}{\sum_{t=1}^{T} h_{t,i}}$$

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

$$\Sigma_i^{\text{diag}} = \text{diag} \{ BS_iB^\top \}$$

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

Both $B$ and $\Sigma_i^{\text{diag}}$ are iteratively improved in each EM step and the likelihood is guaranteed to increase at each step until convergence.
GMM with semi-tied covariances - Example

\[ D = 94 \]

[A.K. Tanwani and S. Calinon. Learning Manipulation Tasks with Tied Gaussian Mixture Models, under review]
Main references

Parsimonious GMM


MFA


MPPCA


GMM with semi-tied covariances

General textbooks

Machine Learning
A Probabilistic Perspective
Kevin P. Murphy

PATTERN RECOGNITION AND MACHINE LEARNING
CHRISTOPHER M. BISHOP

Finite Mixture Models
Geoffrey McLachlan
David Peel
Coordinated MFA by using common factor loadings


Estimation of $K$ and $d_k$ in MFA with Bayesian nonparametrics


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
Sparse subspace clustering with L1 regularization


Deep MFA


Mixture of tensor analyzers (MTA)