Advanced Image Processing and Image Segmentation Techniques – Clustering

Joanna Czajkowska, PhD
Media Systems Group
Institute for Vision and Graphics, University of Siegen

Seminar Schedule

1. Kernelized Clustering Methods
2. Gaussian Mixture Model

Seminar Schedule

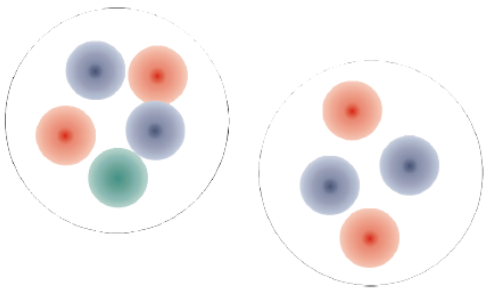
1. Kernelized Clustering Methods
2. Gaussian Mixture Model

Kernelized Clustering Methods

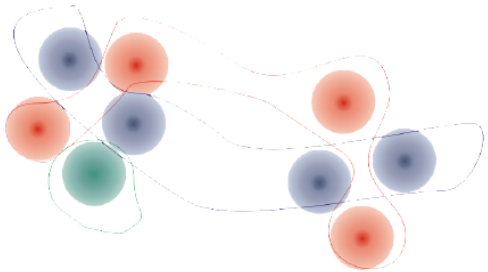
Kernelized Clustering Methods

Clustering

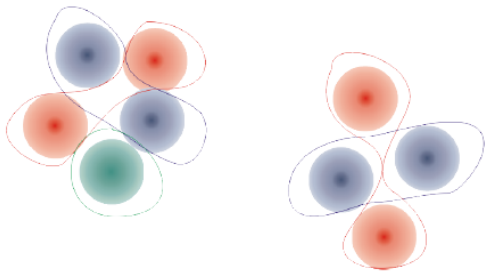
Grouping of the input data set into the subsets containing the elements similar according some criteria.



with respect to the position x and y



with respect to the color

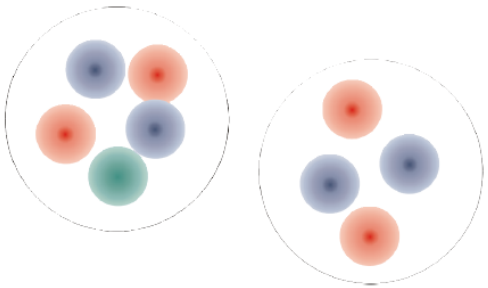


with respect to all the features

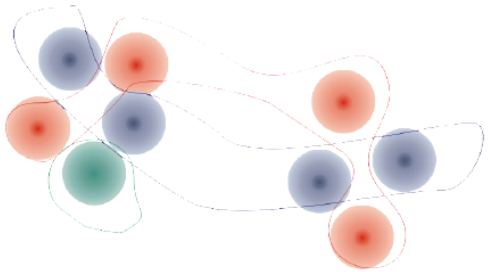
Kernelized Clustering Methods

Clustering

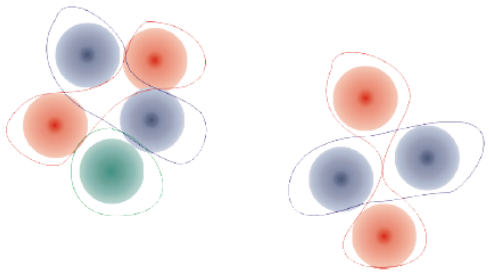
Grouping of the input data set into the subsets containing the elements similar according some criteria.



with respect to the position x and y



with respect to the color



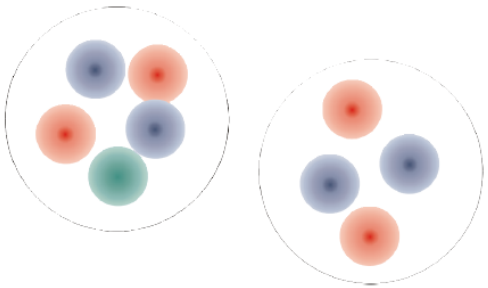
with respect to all the features

In the crisp clustering the result of partitioning are classical sets.

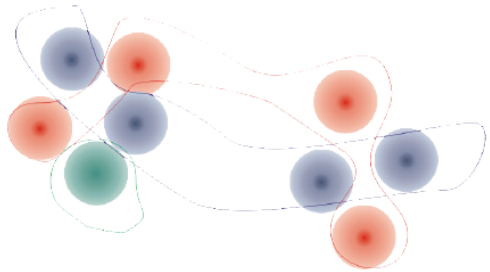
Kernelized Clustering Methods

Clustering

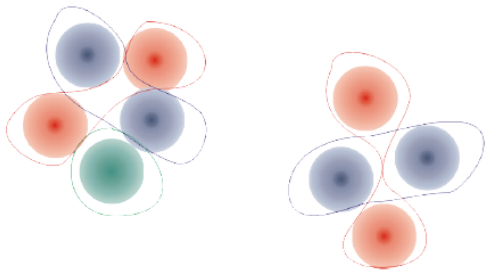
Grouping of the input data set into the subsets containing the elements similar according some criteria.



with respect to the position x and y



with respect to the color



with respect to all the features

In the crisp clustering the result of partitioning are fuzzy sets.

K-Means Clustering - crisp clustering

Distance in the features space

$$d(\underline{x}_1, \underline{x}_2) = \|\underline{x}_1 - \underline{x}_2\| = \\ = \sqrt{(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 + \ldots + (x_{1n} - x_{2n})^2}$$

The goal of the algorithm is minimizing of distances within one group while maximizing distances between groups (group centers/prototypes).

K-Means Clustering - crisp clustering

Objective function:

$$J(\underline{U}, \underline{W}) = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik} ||\underline{x}_k - \underline{w}_i||^2; \qquad \underline{x}_k, \underline{w}_i \in \mathcal{F}$$

- minimizing with respect to the partition matrix elements u_{ik} and the group prototypes $\underline{w}_i$

Kernelized Clustering Methods

K-Means Clustering - crisp clustering

Assumptions:

- $u_{ik} \in \{0, 1\}$
- $u_{ik}||\underline{x}_k - \underline{w}_i||^2$ – similar elements should constitute one group

K-Means Clustering - crisp clustering

========== Algorithm 1. K-Means ==========

1: Initialize groups prototypes V, group number c, number of iterations, accuracy

2: Repeat

In j-th iteration:

$$u_{ik} = \begin{cases} 1 & \text{if the distance of } \underline{x}_k \text{ prototype of group } i\text{-th} \\ & \text{is minimal} \\ 0 & \text{otherwise} \end{cases}$$

$$\underline{w}_i = \frac{\sum_{k=1}^{N} u_{ik} \underline{x}_k}{\sum_{k=1}^{N} u_{ik}} \qquad \forall 1 \leq i \leq c$$

until $max|\underline{w}^j - \underline{w}^{j-1}| < \epsilon$ or $j > max_{iter}$

until $max|\underline{U}^j - \underline{U}^{j-1}| < \epsilon$ or $j > max_{iter}$

==========

Fuzzy c-means (FCM) clustering

Objective function:

$$J(\underline{U}, \underline{W}) = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{\eta} ||\underline{x}_k - \underline{w}_i||^2; \qquad \underline{x}_k, \underline{w}_i \in \mathcal{F}$$

$$0 \leq u_{ik} \leq 1 \text{ where } 1 \leq k \leq N, \ 1 \leq i \leq c$$

$$\sum_{i=1}^{c} u_{ik} = 1, \ \forall 1 \leq k \leq N$$

$$0 < \sum_{k=1}^{N} u_{ik} < N, \ \forall 1 \leq i \leq c$$

where $\eta > 1$ – fuzzyfier

In the case of optimal partition of the data the sum of squared distances is minimal.

Fuzzy c-means (FCM) clustering

Under the assumption that the minimizing is performed alternately:

- for the partition matrix $\underline{U}$, the group prototypes are calculated
- for the set prototypes $\underline{W}$, the partition matrix $\underline{U}$ is created

The minimization conditions:

$$u_{ik} = \frac{||\underline{x}_k - \underline{w}_i||^{\frac{-2}{\eta-1}}}{\sum_{z=1}^{c}(||\underline{x}_k - \underline{w}_z||^{\frac{-2}{\eta-1}})}$$

$$\underline{w}_i = \frac{\sum_{k=1}^{N} u_{ik}^{\eta}\underline{x}_k}{\sum_{k=1}^{N} u_{ik}^{\eta}}$$

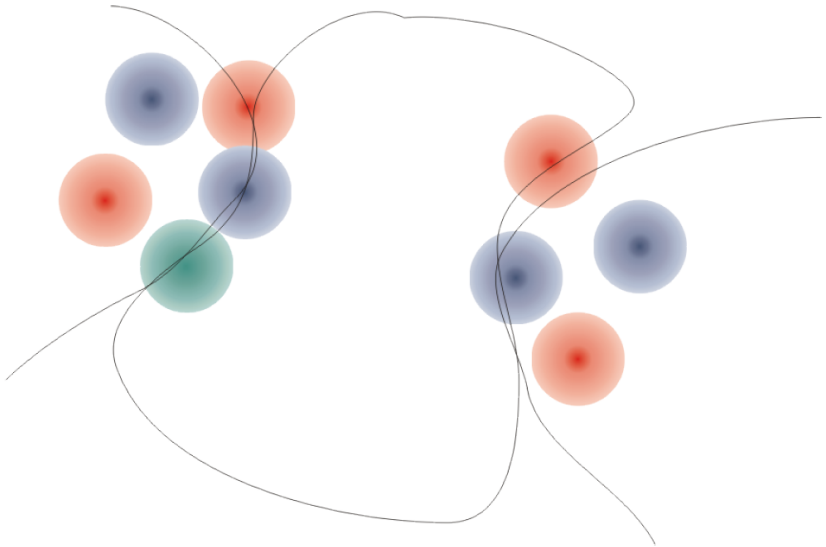
norm $\|\cdot\|$ independently

for the Euclidean norm in $\mathbb{R}^n$

Fuzzy c-means (FCM) clustering

Limitations/drawbacks:

- In most cases FCM clustering results are sufficient, as far as the number c of clusters was chosen properly:

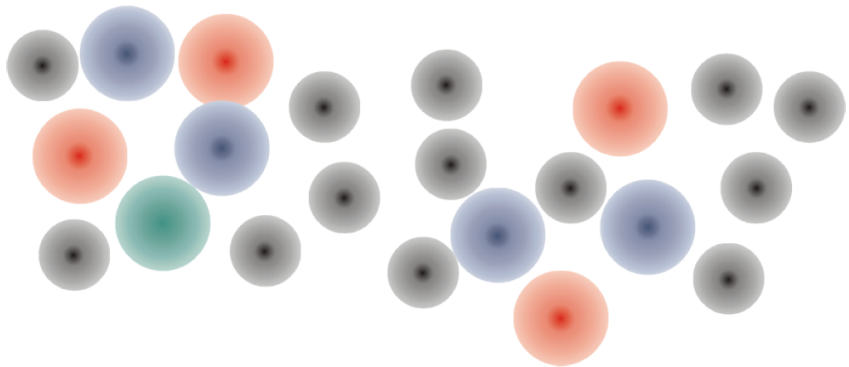


Fuzzy - ISODATA, Cluster Validity Measure

Fuzzy c-means (FCM) clustering

Limitations/drawbacks:

- In most cases FCM clustering results are sufficient, as far as the number c of clusters was chosen properly
- The number of noise data is low enough:

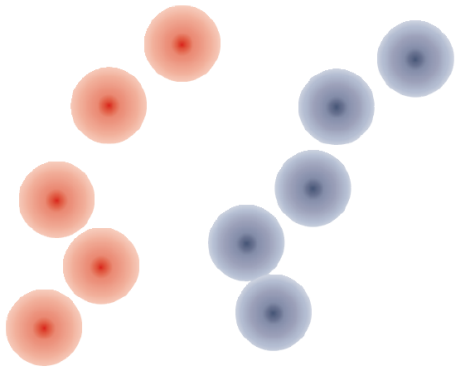


Different group model, taking context information

Fuzzy c-means (FCM) clustering

Limitations/drawbacks:

- In most cases FCM clustering results are sufficient, as far as the number c of clusters was chosen properly
- The number of noise data is low enough
- The groups have a circular shape:

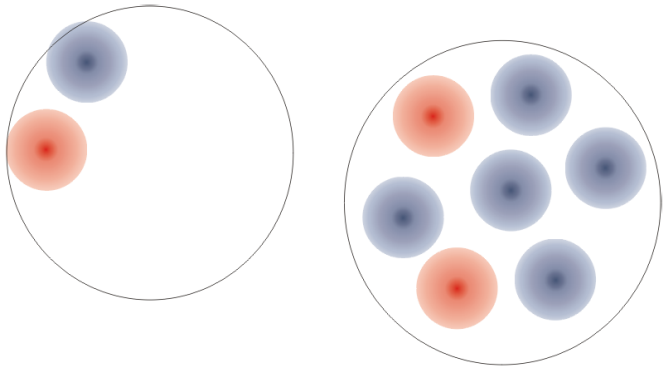


Different norms in $\mathbb{R}^n$, data mapping, C-shells methods

Fuzzy c-means (FCM) clustering

Limitations/drawbacks:

- In most cases FCM clustering results are sufficient, as far as the number c of clusters was chosen properly
- The number of noise data is low enough
- The groups have a circular shape
- The size/cardinality of groups is similar



Data analysis

ISODATA/Fuzzy-ISODATA

========== Algorithm 2. ISODATA ==========

1: Set parameters: number of groups,
   conditions of merging and splitting of groups;
2: Group for the current number of groups;
3: Estimate mean distance within each group
   and mean distance between groups;
4: If the distance within groups is to big
  (take into consideration the set conditions),
   divide the group and go to 2;
5: If the distance between prototypes is to small,
   merge the groups and go to 2;

==========

Number of cluster selection algorithm - CS cluster validity measure

- Input data set $\{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\}$ consists of vectors $\underline{x}_k = [x_{k1}, x_{k2}, \ldots, x_{kD}]$
- $A_i,\ i \in \{1, \ldots, c\}$ set of all the elements of i-th group, and $|A_i|$ cardinality of i-th group
- Group prototypes:

$$\underline{w}_i = \frac{1}{|A_i|} \sum_{\underline{x}_j \in A_i} \underline{x}_j.$$

- CS cluster validity measure for the number of clusters c:

$$CS(c) = \frac{\sum_{i=1}^{c} \left\{ \frac{1}{|A_i|} \sum_{\underline{x}_j \in A_i} \max_{\underline{x}_k \in A_i} \{||\underline{x}_j - \underline{x}_k||\} \right\}}{\sum_{i=1}^{c} \left\{ \min_{j \in C, j \neq i} \{||\underline{w}_i - \underline{w}_j||\} \right\}},$$

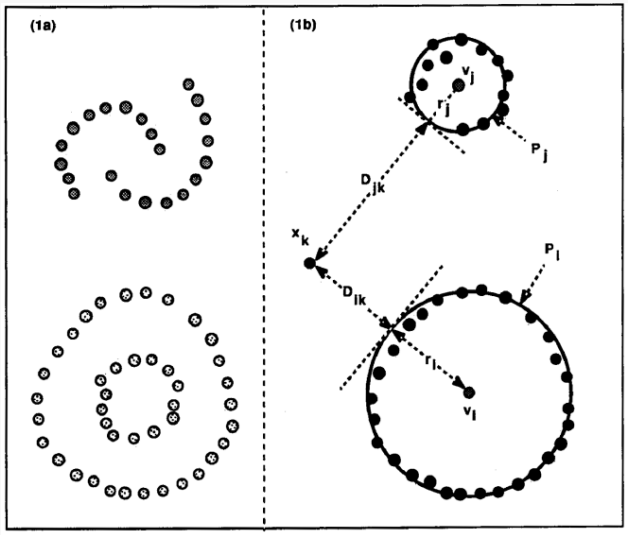
Number of cluster selection algorithm - CS cluster validity measure

The best number of clusters fulfill the formula

$$arg \min_{c \in \{c_{min}, \ldots, c_{max}\}} CS(c).$$

Fuzzy C-Shells

```
===== Algorithm 3. Fuzzy C-Shells
1: Define the group prototypes,
   as the circle with set center
   and radius;
2: Optimize the objective
   function the same as in FCM;
==================
```

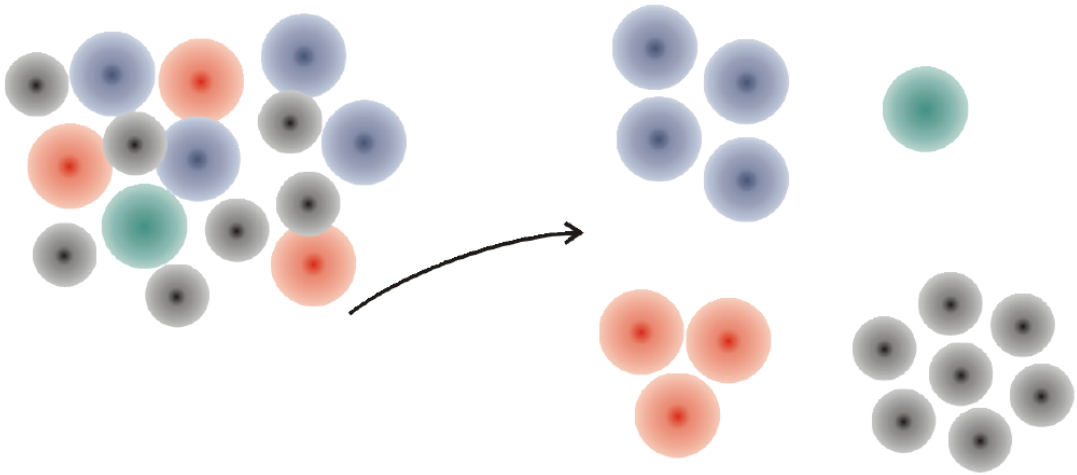


„Robust C-Shells Based Deterministic Annealing Clustering Algorithm", X.L. Yang, FUZZ-IEEE2004

Kernelized Clustering Methods

Mapping the data

Mapping the data into different features space $\mathcal{H}$ some of the FCM clustering limitations do not influence the results any more:



- Separation of data
- Change of data shape
- Separation of noisy data

Kernelized Fuzzy C-Means Clustering

Definition:

Non-linear transformation (mapping) of the data during the clustering procedure – implicit transformation of the data into a multi-dimensional features space:

- Prototypes are calculated in the original space, and the remaining part of clustering procedure is shifted into the kernel space
- The whole clustering procedure is shifted into a kernel space

Kernelized Clustering Methods

Scalar product vs norm

- Scalar product is a measure of data similarity
- Norm is a measure of similarity
- There exist a Hilbert space $\mathcal{H}$, where the scalar product induces a metric

Kernelized Clustering Methods

Scalar product vs norm

- Scalar product is a measure of data similarity
- Norm is a measure of similarity
- There exist a Hilbert space $\mathcal{H}$, where the scalar product induces a metric

The norm (distance) can be replaced by the scalar product e.g. in Euclidean space

Kernelized Fuzzy C-Means Clustering

In the Euclidean space the norm $||\cdot||^2$ is given as

$$\forall \underline{a} \in \mathbb{R}^D \quad ||\underline{a}||^2 = ||(a_1, a_2, \ldots, a_D)||^2 = a_1^2 + a_2^2 + \ldots + a_D^2,$$

$$\forall \underline{x}_k, \underline{w}_i \in \mathcal{X} \quad ||\underline{x}_k - \underline{w}_i||^2 = \sum_{l=1}^{D} (\underline{x}_{kl} - \underline{w}_{il})^2.$$

Kernelized Fuzzy C-Means Clustering

Lemma:

A continuous function $q : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a positive definite kernel over $\mathbb{R}^n$ if is:

1. symmetric

$$q(\underline{x}, \underline{x}') = q(\underline{x}', \underline{x}) \qquad \forall \underline{x}, \underline{x}' \in \mathbb{R}^n$$

2. nonzero-definite

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j (\underline{x}_i, \underline{x}_j) \geq 0$$

for any $n \in \mathbb{N}^+$, *for any objects* $\underline{x}_1, \ldots, \underline{x}_n \in \mathcal{X}$ *and for any coefficients* $c_1, \ldots, c_n \in \mathbb{R}$.

"Kernel Trick"

Lemma:

For any given positive definite kernel over $\mathbb{R}^n$, there exist a Hilbert space $\mathcal{H}$ and a mapping $\phi : \mathcal{X} \to \mathcal{H}$, such that:

$$q(\underline{x}, \underline{x}') = \langle \phi(\underline{x}); \phi(\underline{x}') \rangle \qquad \forall \underline{x}, \underline{x}' \in \mathbb{R}^n$$

where $\langle .; . \rangle$ *denotes scalar product in the Hilbert space* $\mathcal{H}$.

Space $\mathcal{H}$

- Normed vector space of functions

$$f(\cdot) = \sum_{j=1}^{m} \alpha_j q(\underline{x}_j, \cdot)$$

- with scalar product

$$\langle \sum_{i=1}^{m} \alpha_j q(\underline{x}_i, \cdot), \sum_{j=1}^{n} \beta_j q(\underline{x}'_j, \cdot) \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_j \beta_j q(\underline{x}_i, \underline{x}'_j)$$

- and norm

$$\|f\|^2 = \langle f, f \rangle$$

Kernelized Clustering Methods

Applications

- The value of the kernel function is equal to the scalar product of images of data in the space $\mathcal{H}$
- It means, that if there exist a function, which can be represented by scalar product, then even without knowing the image of its elements in the space $\mathcal{H}$ we are able to obtain their scalar product there.
- Knowing the representing a function scalar product, we are able to represent any function in a space $\mathcal{H}$ e.g. distance.

Kernelized Clustering Methods

Applications

- The scalar product makes it possible to look inside the space by showing how the images of two elements in it are similar or dissimilar to each other - thanks to norm.

Kernel functions

- Linear Kernel

$$\forall(\underline{x}_1, \underline{x}_2) \in \mathcal{F} \qquad q(\underline{x}_1, \underline{x}_2) = \langle \underline{x}_1; \underline{x}_2 \rangle ,$$

- RBF - Radial Basis Function

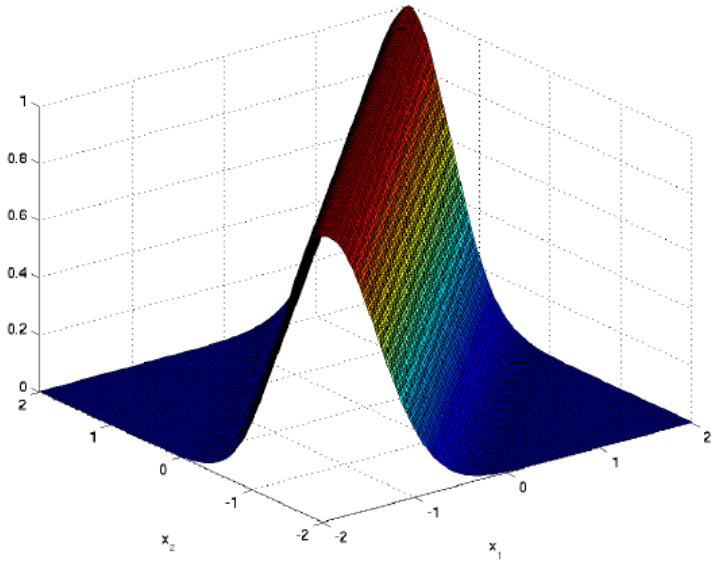
$$\forall(\underline{x}_1, \underline{x}_2) \in \mathcal{F}, \forall \sigma \in \mathbb{R}^+ \qquad q(\underline{x}_1, \underline{x}_2) = e^{-\frac{||\underline{x}_1 - \underline{x}_2||^2}{\sigma_{rbf}}},$$

- Polynomial Kernel

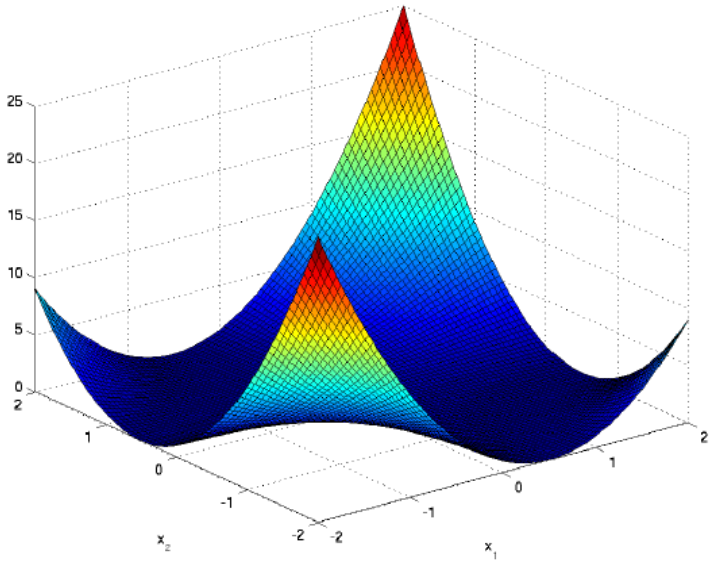
$$\forall(\underline{x}_1, \underline{x}_2) \in \mathcal{F}, \forall a \in \mathbb{R} : a > 0 \qquad q(\underline{x}_1, \underline{x}_2) = (\langle \underline{x}_1; \underline{x}_2 \rangle + a)^b,$$

- linear combination and product of q_1 and q_2
- combination of q_1 with *exp*

Kernel functions



(a) $\mathrm{q} = \exp(\frac{-||\mathbf{x_1}-\mathbf{x_2}||}{2\sigma^2})$



(b) $\mathrm{q} = (\langle \mathbf{x_1}; \mathbf{x_2} \rangle + a)^b$

Kernelized Fuzzy C-Means Clustering

Each kernel function (each kernel) and each input data set of form $\{\underline{x}_1, \ldots, \underline{x}_N\} \subset \mathcal{F}$ can be represented by a symmetric and positive defined hermitrian matrix:

$$\underline{Q}[\underline{z}] = (q(\underline{x}_i, \underline{x}_j))_{i,j}$$

of size $N \times N$, called kernel matrix.

Kernelized Fuzzy C-Means Clustering

The scalar product (inner product) is defined as quadratic hermitrian and can be represented as:

$$\left\langle \underline{w}_i - \underline{w}_j; \underline{w}_i - \underline{w}_j \right\rangle = \langle \underline{w}_i; \underline{w}_i \rangle - 2 \left\langle \underline{w}_i; \underline{w}_j \right\rangle + \left\langle \underline{w}_j; \underline{w}_j \right\rangle .$$

Kernelized Clustering Methods

Kernelized Fuzzy C-Means Clustering

There exist two classes of c-means methods with kernel functions:

- Modification in the estimation of values v_i and u_{ik} characteristic for the space $\mathbb{R}^n$ (e.g. FKCM/KFCM). The clustering itself has not been performed fully in the kernel space, though, as the cluster prototypes are computed in the original space.
- Estimation of necessary condition v_i and u_{ik} of existing a minimum for the kernel space (e.g. KWCM) – matrix trace based clustering approach (implicit prototypes).

Approach "from $\mathbb{R}^n$ to $\mathcal{H}$"

- Dependencies in $\mathbb{R}^n$

Prototypes:

$$\underline{w}_i = \frac{\sum_{k=1}^{N} u_{ik}^{\eta} \underline{x}_k}{\sum_{k=1}^{N} u_{ik}^{\eta}}$$

Partition matrix:

$$u_{ik} = \frac{\left\| \underline{x}_k - \underline{w}_i \right\|^{\frac{-1}{\eta-1}}}{\sum_{z=1}^{c} \left(\left\| \underline{x}_k - \underline{w}_z \right\|^{\frac{-1}{\eta-1}} \right)}$$

Approach "from $\mathbb{R}^n$ to $\mathcal{H}$"

- Dependencies in $\mathbb{R}^n$
- Scalar product representation

Prototypes:

$$\underline{w}_i = \frac{\sum_{k=1}^{N} u_{ik}^{\eta} \underline{x}_k}{\sum_{k=1}^{N} u_{ik}^{\eta}}$$

Partition matrix:

$$u_{ik} = \frac{\left\langle \underline{x}_k - \underline{w}_i ; \underline{x}_k - \underline{w}_i \right\rangle^{\frac{-1}{\eta-1}}}{\sum_{z=1}^{c} \left(\left\langle \underline{x}_k - \underline{w}_z ; \underline{x}_k - \underline{w}_z \right\rangle^{\frac{-1}{\eta-1}} \right)}$$

Kernelized Clustering Methods

Approach "from $\mathbb{R}^n$ to $\mathcal{H}$"

- Dependencies in $\mathbb{R}^n$
- Scalar product representation
- Replacing the scalar product by the kernel function

Partition matrix:

$$u_{ik} = \frac{(q(\underline{x}_k;\underline{x}_k) - 2q(\underline{x}_k;\underline{w}_i) + q(\underline{w}_i;\underline{w}_i))^{\frac{-1}{\eta-1}}}{\sum_{z=1}^{c}((q(\underline{x}_k;\underline{x}_k) - 2q(\underline{x}_k;\underline{w}_z) + q(\underline{w}_z;\underline{w}_z))^{\frac{-1}{\eta-1}})}$$

Fuzzy kernel-induced c-means (FKCM) clustering

Kernelized Clustering Methods

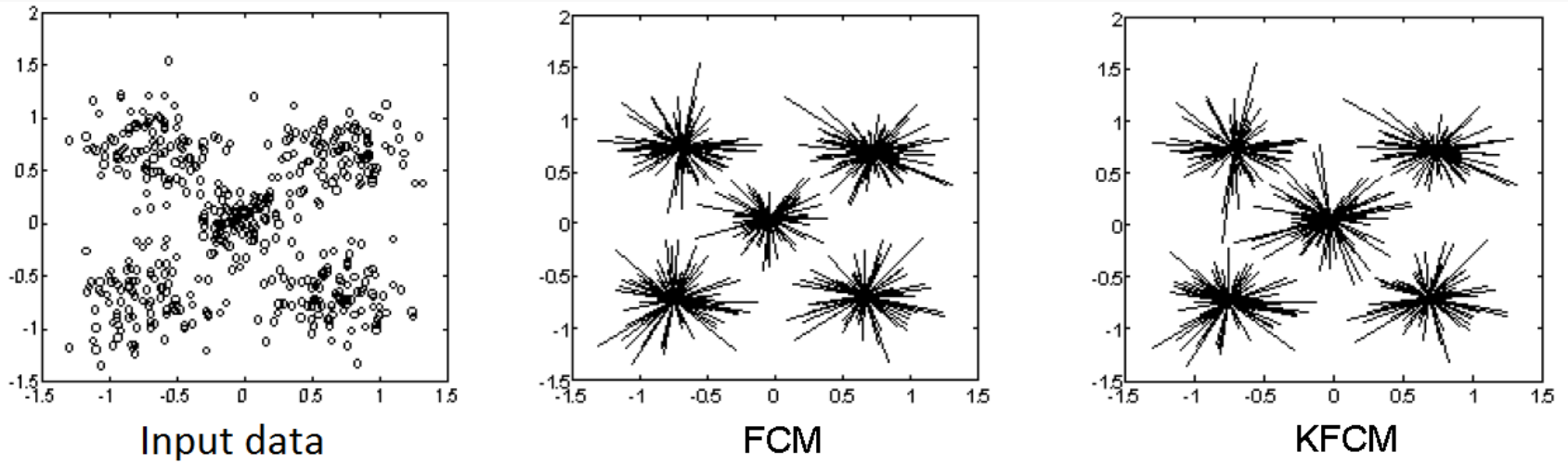
Kernelized Fuzzy C-Means Clustering

Features:

- Decreased sensitivity to artifacts and noise
- No limitation concerning cluster shape (Euclidean space $\longrightarrow$ hyperspherical)
- The cardinality of groups does not influence the results
- Reduction of influence of "unclassified data"

Kernelized Fuzzy C-Means Clustering

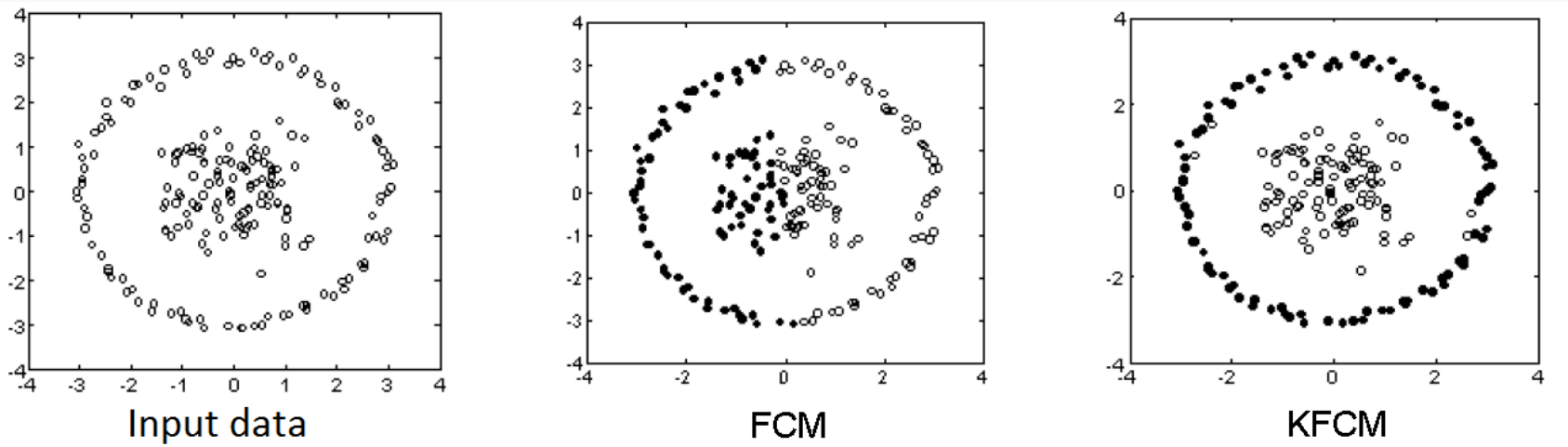
Example 1:



Zhong-dong Wu, Wei-xin Xie, Jian-ping Yu, Fuzzy C-Means Clustering Algorithm based on Kernel Method

Kernelized Fuzzy C-Means Clustering

Example 2:



Zhong-dong Wu, Wei-xin Xie, Jian-ping Yu, Fuzzy C-Means Clustering Algorithm based on Kernel Method

Kernelized Clustering Methods

Kernelized Fuzzy C-Means Clustering

Advantages:

- Easy/intuitive to modify
- Possibility to transfer to other methods
- High computation speed

Kernelized Fuzzy C-Means Clustering

Median modification – Fuzzy C-Means with Median Spatial Constraint:

$$J(\underline{U}_H, \underline{W}) = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{\eta} \|\underline{x}_k - \underline{w}_i\|^2 + \alpha u_{ik}^{\eta} \|\widehat{\underline{x}}_k - \underline{w}_i\|^2$$

$$u_{ik} = \frac{(\|\underline{x}_k - \underline{w}_i\|_H^2 + \alpha \|\widehat{\underline{x}}_k - \underline{w}_i\|_H^2)^{\frac{-1}{\eta-1}}}{\sum_{z=1}^{c} ((\|\underline{x}_k - \underline{w}_z\|_H^2 + \alpha \|\widehat{\underline{x}}_k - \underline{w}_z\|_H^2)^{\frac{-1}{\eta-1}})} \qquad \underline{w}_i = \frac{\sum_{k=1}^{N} u_{ik}^{\eta} (\underline{x}_k + \alpha \widehat{\underline{x}}_k)}{(1+\alpha) \sum_{k=1}^{N} u_{ik}^{\eta}}$$

Kernelized Fuzzy C-Means Clustering

Approach "clustering in $\mathcal{H}$":

$$J(\underline{U}, \underline{W}) = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{\eta} \|\phi_k - \underline{w}_i\|_H^2 \qquad \phi_k, \underline{w}_i \in \mathcal{H},$$

$$u_{ik} = \frac{\|\phi_k - \underline{w}_i\|_H^{2\frac{-1}{\eta-1}}}{\sum_{z=1}^{c} (\|\phi_k - \underline{w}_i\|_H^{2\frac{-1}{\eta-1}})}$$

$$\underline{W} = \{\underline{w}_i : \frac{\partial J}{\partial \underline{w}_i} = 0,\ 1 \leq i \leq c\}$$

It requires the knowledge of $\frac{\partial \|\cdot\|}{\partial \underline{w}_i}$

Kernelized Fuzzy C-Means Clustering

Clustering with hidden prototypes:

If the prototypes required to obtain minimum are set to $\underline{w}_i = \frac{\sum_{k=1}^{N} u_{ik}^{\eta} \phi_k}{\sum_{k=1}^{N} u_{ik}^{\eta}}$, then

$$u'_{ik} = \frac{\langle \phi_k - C_i \sum_{p=1}^{N} u_{ip}^{\eta} \phi_p ; \phi_k - C_i \sum_{p=1}^{N} u_{ip}^{\eta} \phi_p \rangle^{\frac{-1}{\eta-1}}}{\sum_{z=1}^{c} \left(\langle \phi_k - C_z \sum_{p=1}^{N} u_{zp} ; \phi_k - C_z \sum_{p=1}^{N} u_{zp} \rangle^{\frac{-1}{\eta-1}} \right)},$$

where

$$C_j = (\sum_{p=1}^{N} u_{jp}^{\eta})^{-}1$$

Kernelized Fuzzy C-Means Clustering with Hidden Prototypes

Based on the bilinearity of scalar product:

$$u'_{ik} = \frac{\left(\langle \phi_k ; \phi_k \rangle - 2\langle \phi_k ; C_i \sum_{p=1}^{N} u_{ip}^{\eta} \phi_p \rangle + \langle C_i \sum_{p=1}^{N} u_{ip}^{\eta} \phi_p ; C_i \sum_{p=1}^{N} u_{ip}^{\eta} \phi_p \rangle \right)^{\frac{-1}{\eta-1}}}{\sum_{z=1}^{c} \left(\langle \phi_k ; \phi_k \rangle - 2\langle \phi_k ; C_z \sum_{p=1}^{N} u_{zp}^{\eta} \phi_p \rangle + \langle C_z \sum_{p=1}^{N} u_{zp}^{\eta} \phi_p ; C_z \sum_{p=1}^{N} u_{zp}^{\eta} \phi_p \rangle \right)^{\frac{-1}{\eta-1}}}$$

Kernelized Fuzzy C-Means Clustering with Hidden Prototypes

Scalar product in the space $\mathcal{H}$ can be replaced by the proper elements of kernel matrix $\underline{Q}$:

$$\begin{bmatrix} \langle \phi_1; \phi_1 \rangle & \cdots & \langle \phi_1; \phi_N \rangle \\ \vdots & \ddots & \vdots \\ \langle \phi_N; \phi_1 \rangle & \cdots & \langle \phi_N; \phi_N \rangle \end{bmatrix} = \begin{bmatrix} q_{11} & \cdots & q_{1N} \\ \vdots & \ddots & \vdots \\ q_{N1} & \cdots & q_{NN} \end{bmatrix} = \underline{Q},$$

where $q_{ij} = q(\underline{x}_i, \underline{x}_j) = \langle \phi_i; \phi_j \rangle$.

Kernelized Fuzzy C-Means Clustering

Clustering with hidden prototypes (KWCM):

$$u'_{ik} = \frac{(C_i^2 \sum_{p=1}^{N}\sum_{r=1}^{N}(u_{ip}u_{ir})^{\eta} q_{pr} - 2C_i \sum_{p=1}^{N} u_{ip}^{\eta} q_{pk} + q_{kk})^{\frac{-1}{\eta-1}}}{\sum_{z=1}^{c}(C_z^2 \sum_{p=1}^{N}\sum_{r=1}^{N}(u_{zp}u_{ir})^{\eta} q_{pr} - 2C_z \sum_{p=1}^{N} u_{zp}^{\eta} q_{pk} + q_{kk})^{\frac{-1}{\eta-1}}}$$

where

$$C_j = (\sum_{p=1}^{N} u_{jp}^{\eta})^{-}1$$

Kernelized Clustering Methods

Kernelized Fuzzy C-Means Clustering

Drawbacks of kernelized clustering methods:

- Computational complexity:
  - "from $\mathbb{R}^n$ to $\mathcal{H}$" – $O(n)$
  - "Clustering in $\mathcal{H}$" – $O(n^2)$

Gaussian Mixture Model

Teaching

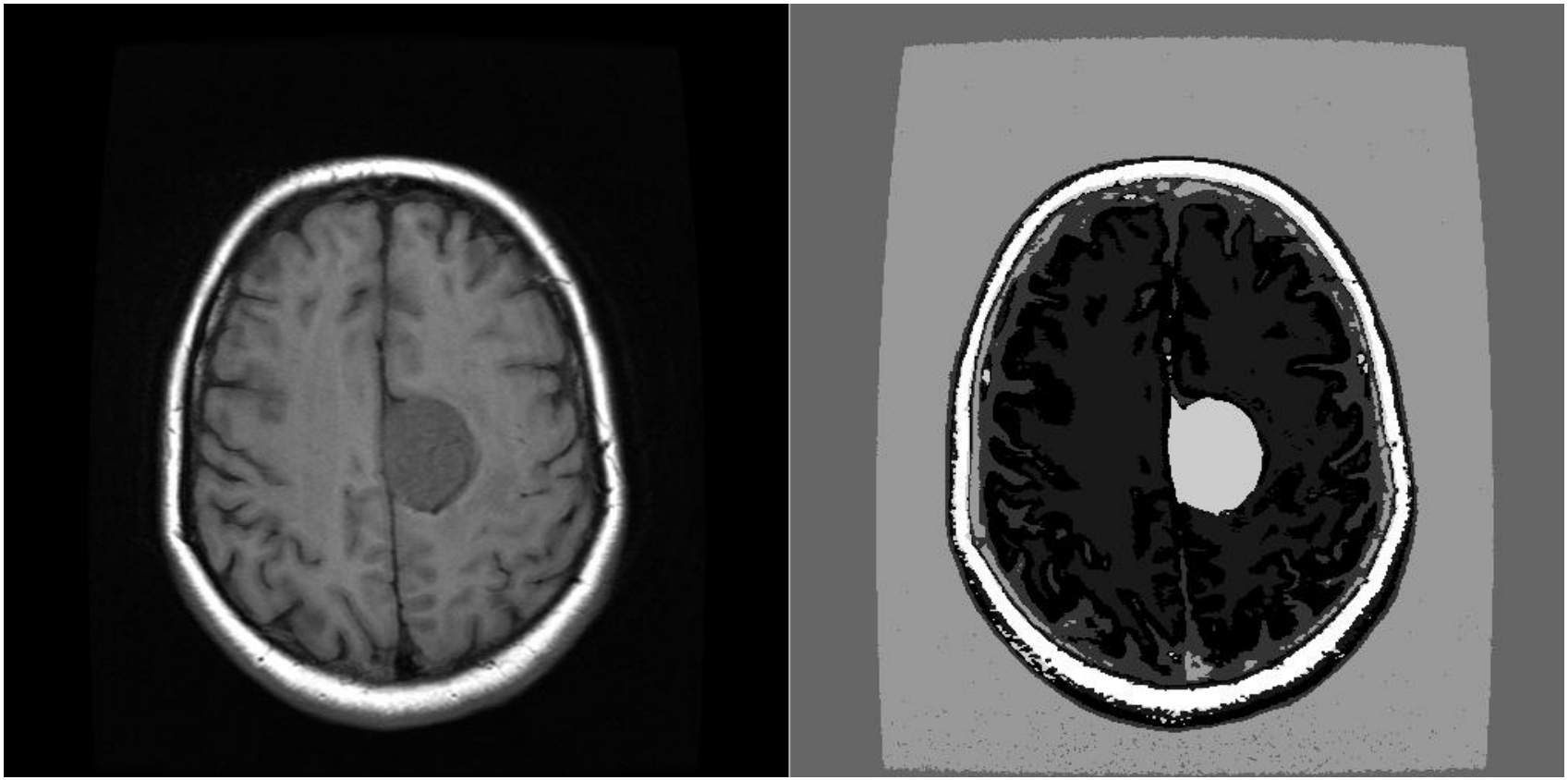
Supervised

- Classification based on available information concerning division of the data set into the groups.

Unsupervised

- Describe the data structure or data regularity in a case, when the information concerning analyzed data set are negligible or we do not know anything.

Example:



http://www.mathworks.com/matlabcentral

Clustering

Clustering

- Division of the data set into the subsets of distinguishable group (clusters) – generalization of information.

Cluster

- Set of objects, which are maximally similar to each other and maximally different from the objects belonging to different sets (groups).

Classification

Classifier

Decision rule, using which the object is assigned to particular classes.

Classification

Supervised

- Bayesian Classifier
- Support Vector Machine
- Artificial Neural Networks
- ...

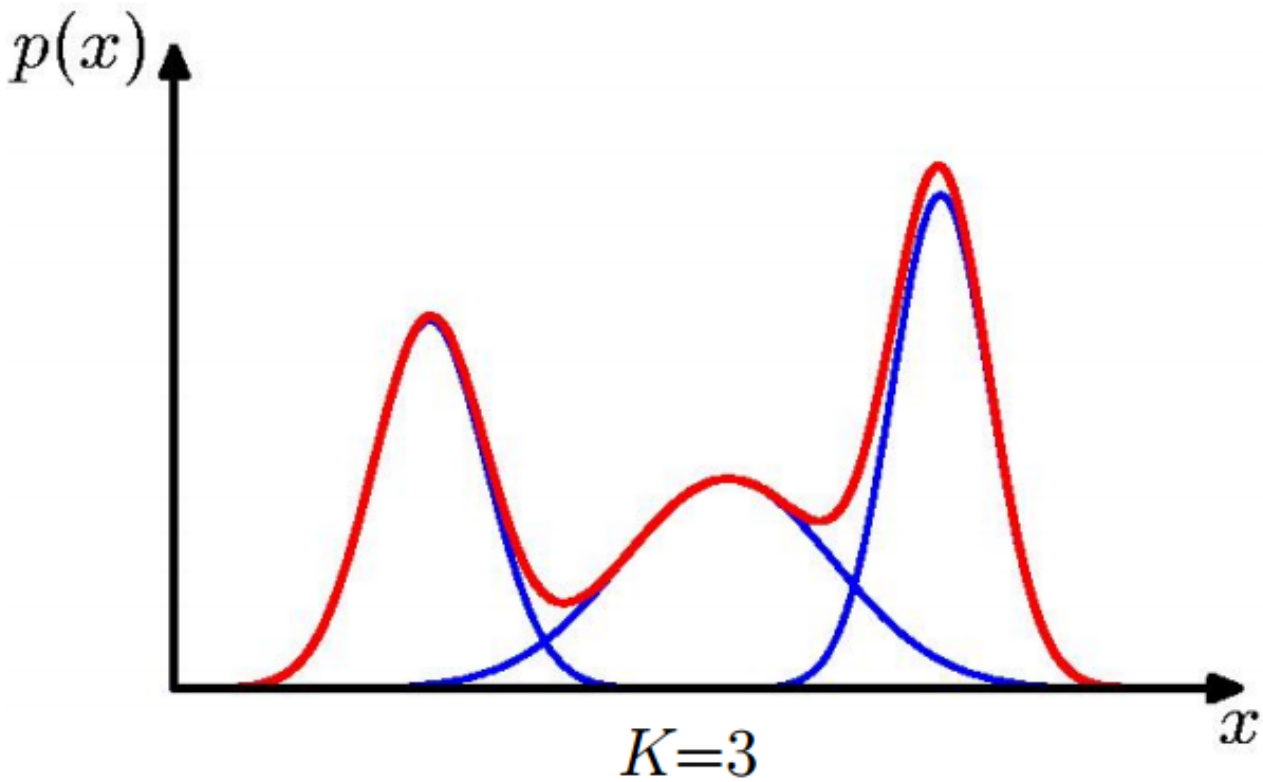
Unsupervised

- Hierarchical methods
- K-mean method and its modification
- Statistical methods (e.g. mixture models)
- ...

Mixture Model

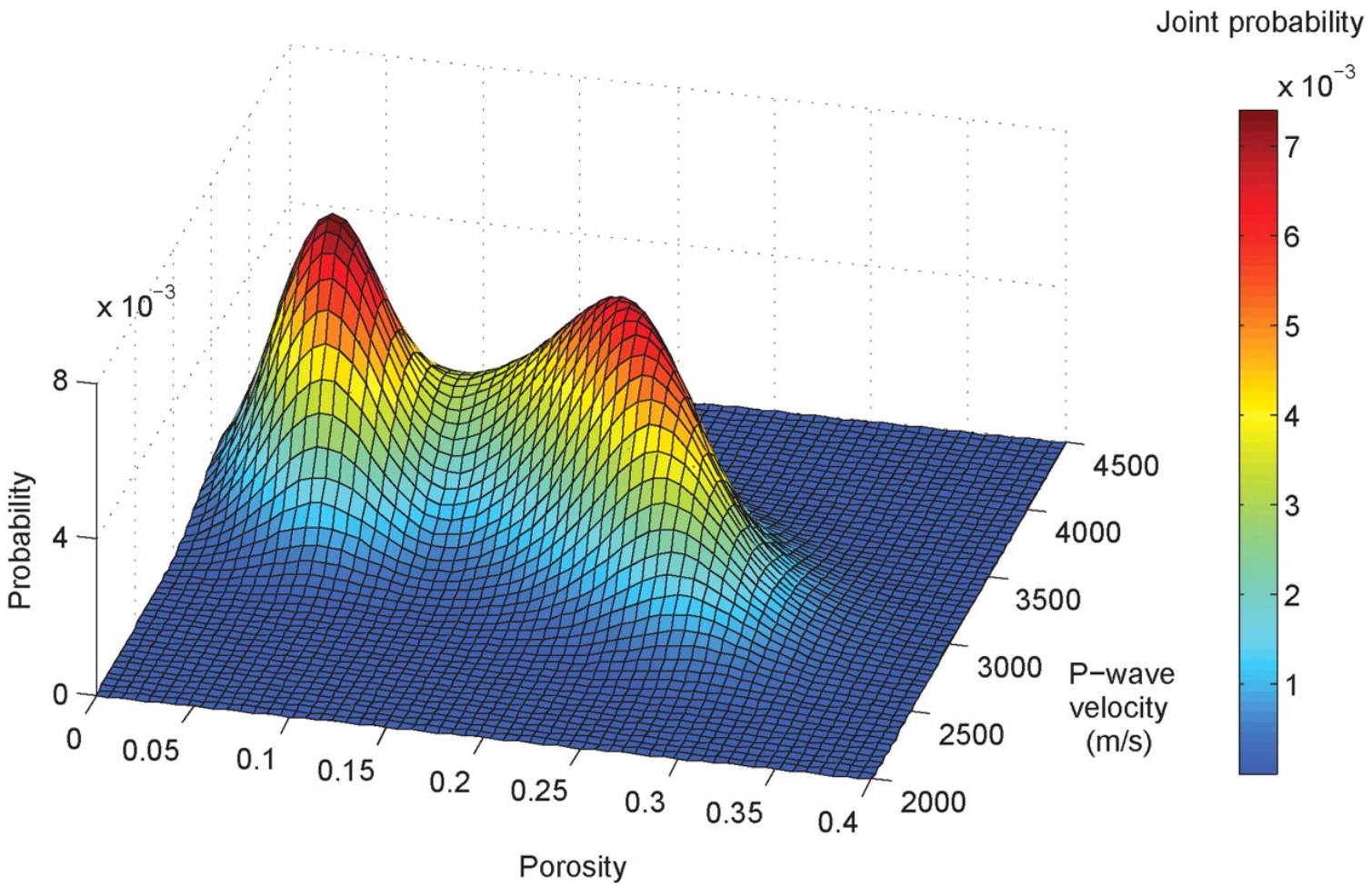
In this approach the data are considered as coming from the population with the probability distribution created by a mixture of probability distributions, in which each component represents separate resulting group.

One-dimensional data:



http://www.robots.ox.ac.uk/

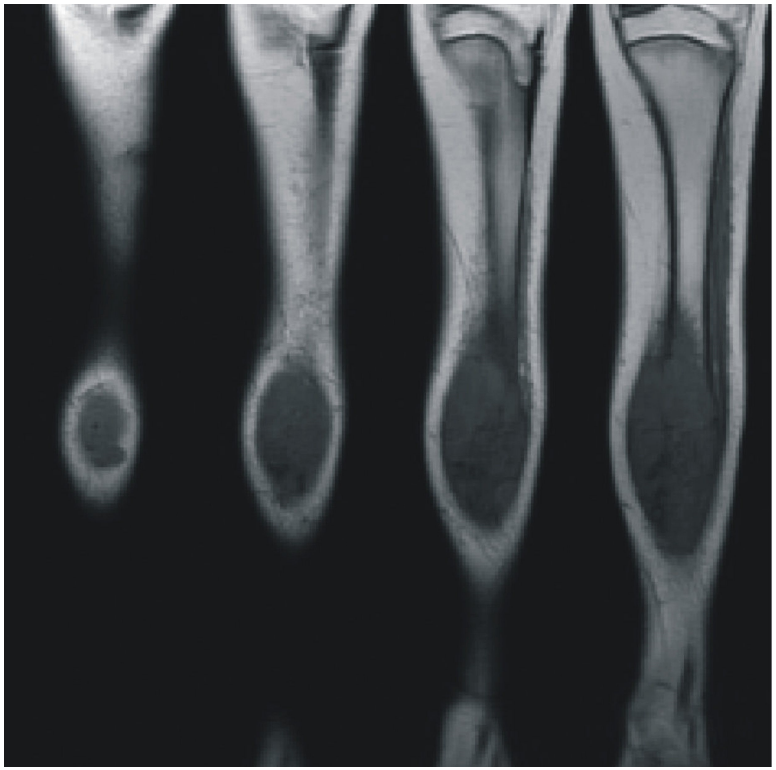
Two-dimensional data:

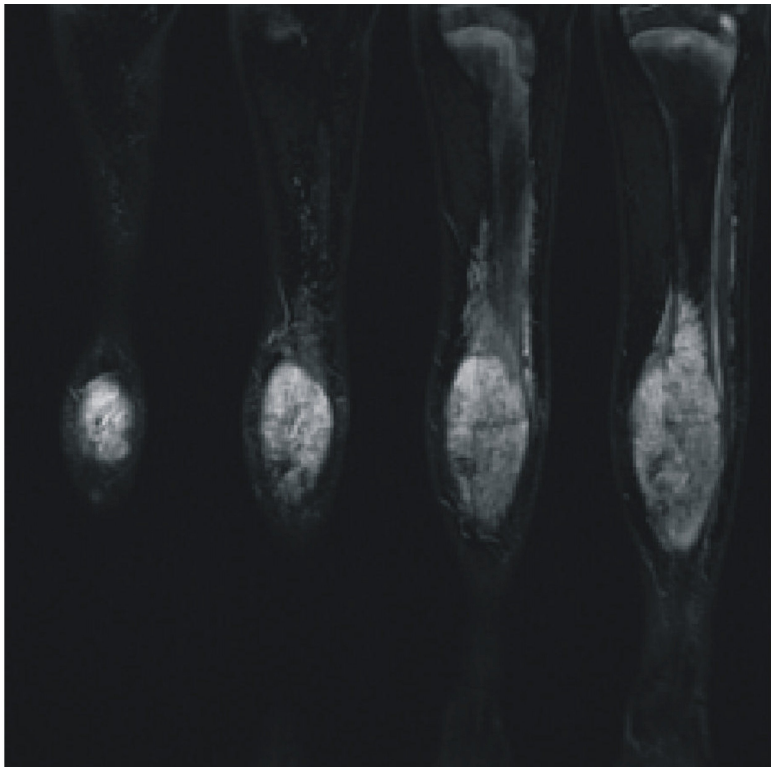


http://tle.geoscienceworld.org/content/30/1/54/F3.expansion.html

Clustering

Data representation:





- Data: different image series (T1-weighted, T2-weighted)
- Features: gray intensity levels

Data representation, X:

- $\underline{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_D^{(i)}]^T$, $i \in \{1, \ldots, m\}$ – m independent probes.
- The size of each probe is given as D.
- Example:

	Features	
	T1+C	T2
$x^{(1)}$	0.5	0.9
$x^{(2)}$	0.1	0.7
$\vdots$	$\vdots$	$\vdots$
$x^{(m)}$	0.3	0.2

	Features
	$T1 + C$
$x^{(1)}$	0.5
$x^{(2)}$	0.1
$\vdots$	$\vdots$
$x^{(m)}$	0.3

GMM

Assumptions:

- The data set is drawn form k populations
- The strength of the assignment to the j-th population depends on distance – soft-assignment
- We would like to estimate/identify the parameters of each population
- We do not have the *a priori* knowledge concerning the populations
- We "strongly believe" that their probability density functions are Gaussian – Shapiro-Wilk test ;D

Goal: Fit a Set of k Gaussians to the data – Maximum Likelihood estimator

Mixture

- For m independent probes $\underline{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_D^{(i)}]^T$, $j \in \{1, \ldots, m\}$ of size D probability density function of a probe $\underline{x}^{(i)}$ in a mixture is given as

$$p(\underline{x}) = \sum_{j=1}^{k} \pi_j p_j(\underline{x}),$$

- where $p_j(\underline{x})$ is probability density function of j-th form k components, and $\pi_j \in [0, 1] : j \in \{1, \ldots, k\}$ is its mixing proportions coefficient, such that

$$\sum_{j=1}^{k} \pi_j = 1.$$

Mixture of Gaussians

- The Gaussian Mixture Model assumes that each group of the data is generated by normal probability distribution

$$p(\underline{x};\theta_j) = \frac{1}{(2\pi)^{\frac{D}{2}} det(\Sigma_j)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\underline{x}-\underline{\lambda}_j)^T \Sigma_j^{-1}(\underline{x}-\underline{\lambda}_j)\right\},$$

- where $\underline{\lambda}_j$ and Σ_j are the parameters of D-dimensional normal probability distribution $N(\underline{\lambda}_j, \Sigma_j)$, mean values vector $(\underline{\lambda}_j)$ and covariance matrix(Σ_j).

Covariance Matrix:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{n1} & \cdots & \sigma_n^2 \end{bmatrix}$$

- $\sigma_i^2 = D^2(X_i)$ – variance of random variable X_i
- $\sigma_{ij} = cov(X_i, X_j)$ – covariance between variables X_i i X_j
- Matrix Σ is symmetric

Variance of random variable

Measures the spread, or variability, of the distribution of X.
Variance estimators:

- biased sample variance

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \lambda)^2$$

- unbiased sample variance

$$\sigma^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x^{(i)} - \lambda)^2$$

Covariance

The value describing linear dependence between random variables X_i and X_j.

Covariance matrix

- Estimate the mean values λ_i and λ_j of vectors X_i and X_j
- $cov(X_i, X_j) = \frac{1}{m-1} \sum_{k=1}^{m} ((X_i)^{(k)} - \lambda_i)((X_j)^{(k)} - \lambda_j)$

Expected value

Weighted average of all possible values. Expected value of discrete random variable X of probability density function $p(x)$, $(P(X = x^{(i)}) = P(x^{(i)}))$ is given as

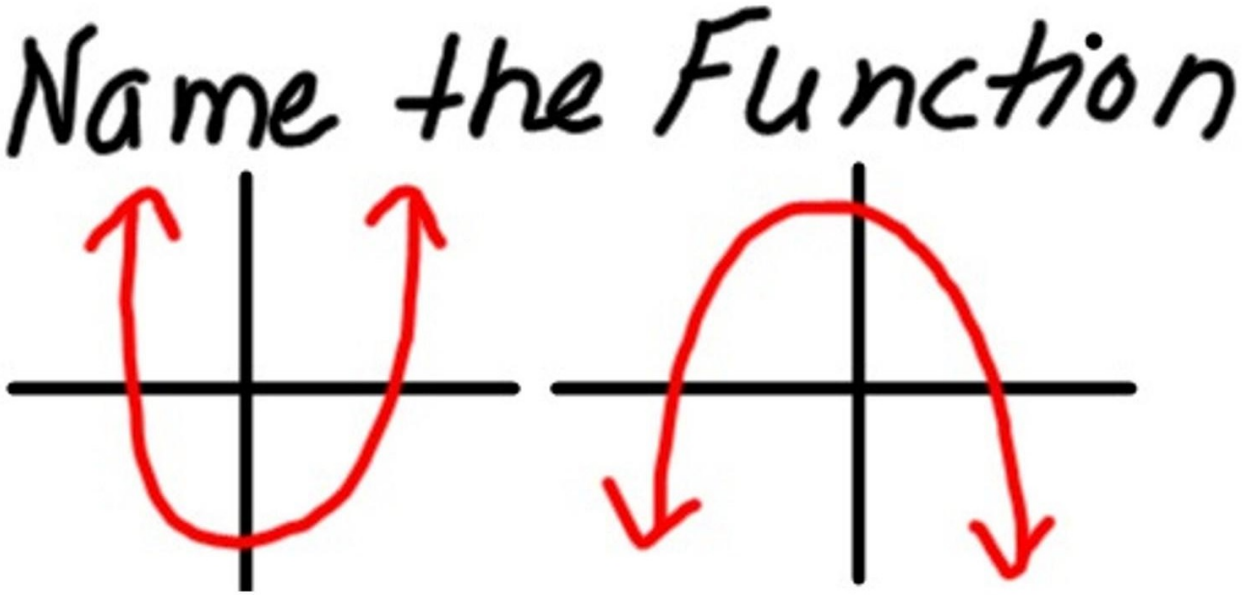
$$E[X] = \sum_{i=1}^{m} x^{(i)} P(x^{(i)}).$$

Jensen's inequality

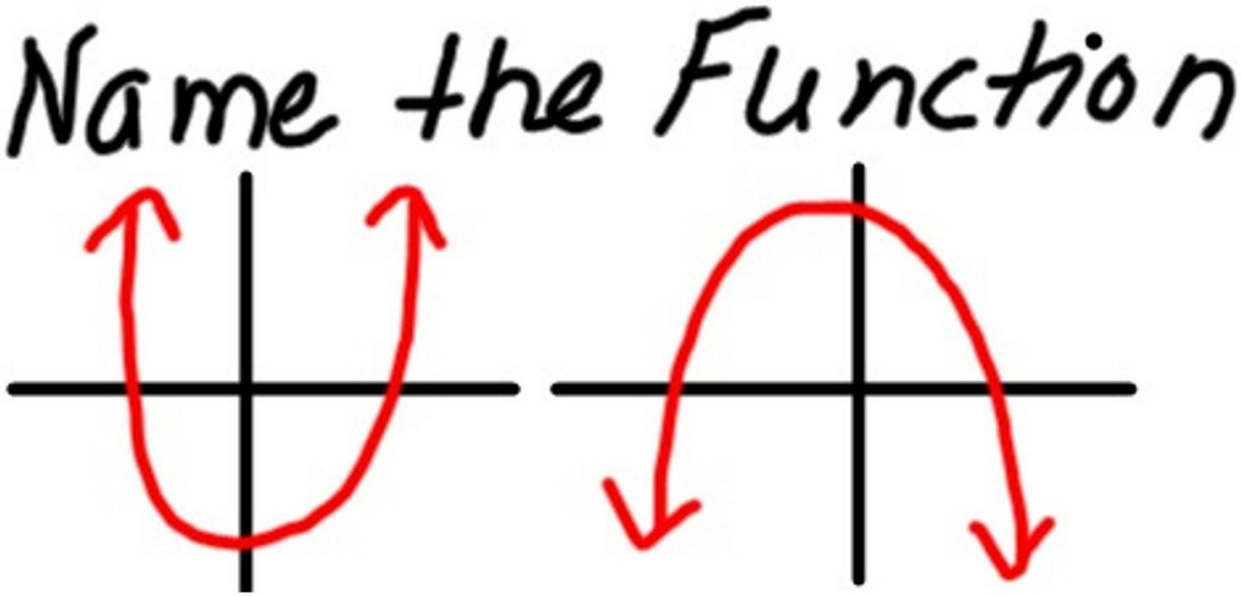
Let f be a convex function defined on a given interval ($f''(x) \geq 0$). For any $x_1, x_2, \ldots, x_n$, $n \geq 2$ from the given interval and for any constants $a_1, a_2, \ldots, a_n$, such that $a_1 + a_2 + \ldots + a_n = 1$ inequality

$$a_1 f(x_1) + a_2 f(x_2) + \ldots + a_n f(x_n) \geq f(a_1 x_1 + a_2 x_2 + \ldots + a_n x_n).$$

is true.



http://www.studyblue.com/notes/note/n/geometry-elephant-quiz-cards/deck/1085351



http://www.studyblue.com/notes/note/n/geometry-elephant-quiz-cards/deck/1085351

convex concave

Jensen's inequality

Let f be a convex function ($f''(x) \geq 0$), and $X \in \{x^{(i)} : i = 1, \ldots, m\}$ be a random variable with probabilities $P(x^{(i)})$ where $\sum P(x^{(i)}) = 1$, then

$$E[X] = \sum_{i=1}^{m} x^{(i)} P(x^{(i)}).$$

$$f(E[X]) \leq E[f(X)]$$

$$f\left(\sum_{i=1}^{m} x^{(i)} P(x^{(i)})\right) \leq \sum_{i=1}^{m} f(x^{(i)}) P(x^{(i)})$$

Jensen's inequality

Let f be a convex function ($f''(x) \geq 0$), and $X \in \{x^{(i)} : 1, \ldots, m\}$ be a random variable with probabilities $P(x^{(i)})$ where $\sum P(x^{(i)}) = 1$, then

$$f(E[X]) \leq E[f(X)]$$

Moreover, if $f''(x) > 0$ (f is a strictly convex function)

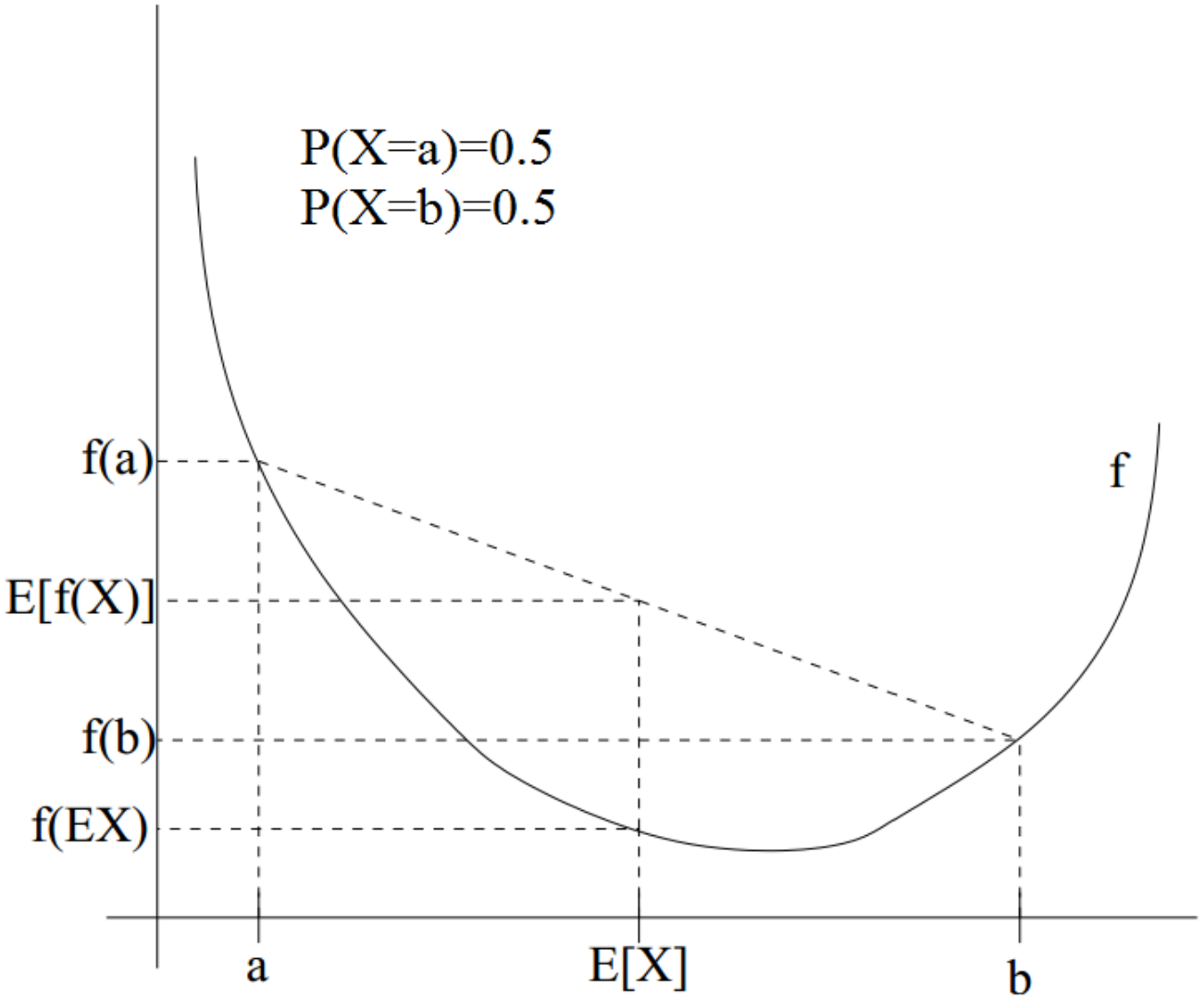
$$f(E[X]) < E[f(X)],$$

then $E[f(X)] = f(E[X])$ holds true if and only if $X = E[X]$ with probability $P(X) = 1$ (X is constant).
If function f is concave

$$f(Ex) \geq E[f(x)].$$

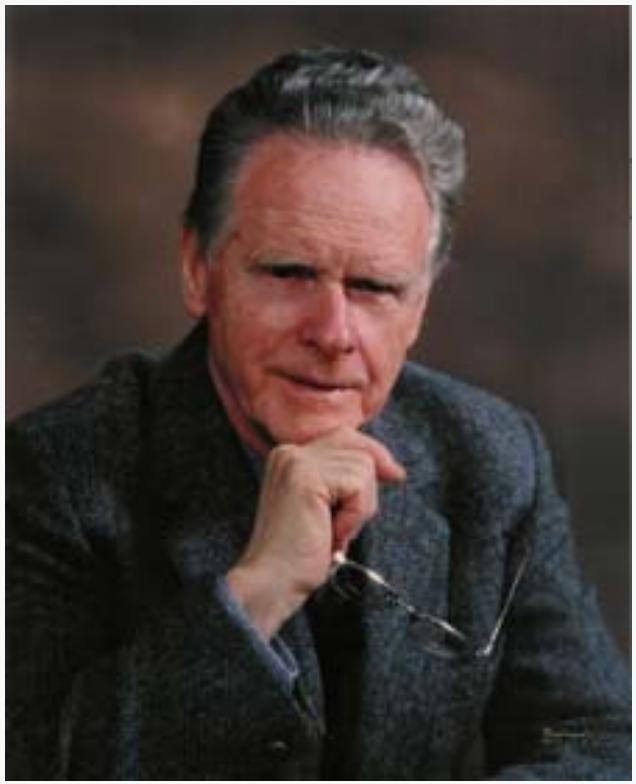
Jensen's inequality

Proof:



EM Algorithm

Algorithm developed by Dempster (1997) – finding a maximum likelihood of parameters in statistical models (of parametric probability distribution), where the model depends on unobserved latent variables (group labels).



Applications:

- learning an optimal mixture of fixed parameters
- estimating the parameters of a compound Dirichlet distribution
- dis-entangling superimposed signals

Example:

- Consider the temperature outside for each 24 hours a day $x \in \mathbb{R}^{24}$
- Let say that the temperature depends on season $\theta \in \{sumer, fall, winter, spring\}$
- The seasonal temperature distribution $p(x;\theta)$ - known
- Measure: the average temperature $y = \widehat{x}$ for some day
- Question: what is the season?
- We seek the maximum likelihood estimate of $\widehat{\theta}$, that is, the value that maximizes $p(y;\theta)$

General EM Algorithm

- Assume initially that the entire training data set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ consisting of m independent examples.
- θ – all the parameters of the model $p(x, z)$
- Goal – choose θ to maximize the likelihood function $\ell(\theta)$

- Let Z be any discrete auxiliary random variable, whose distribution is a function of θ
- Let z range over the possible outcomes of Z
- By definition

$$p(x; \theta) = \sum_z p(x, z; \theta).$$

General EM Algorithm

- The likelihood is then given by

$$\ell(\theta) = \sum_{i=1}^{m} \log p(x;\theta) = \sum_{i=1}^{m} \log \sum_{z} p(x,z;\theta)$$

- Explicitly finding the maximum likelihood estimates of the parameters θ may be hard
- $z^{(i)}$'s are the latent random variables and (often) if the $z^{(i)}$'s were observed, the maximum likelihood estimation would be easy

EM algorithm

- The EM algorithm is an efficient method for maximum likelihood estimation
- Idea: Instead of maximizing $\ell(\theta)$ explicitly, repeatedly construct a lower-bound on ℓ and then optimize that lower bound
- Therefore:
  - E-step: construct a lower-bound
  - M-step: optimize the lower-bound

EM algorithm

- Let be Q_i (for each i) – a distribution over the z's

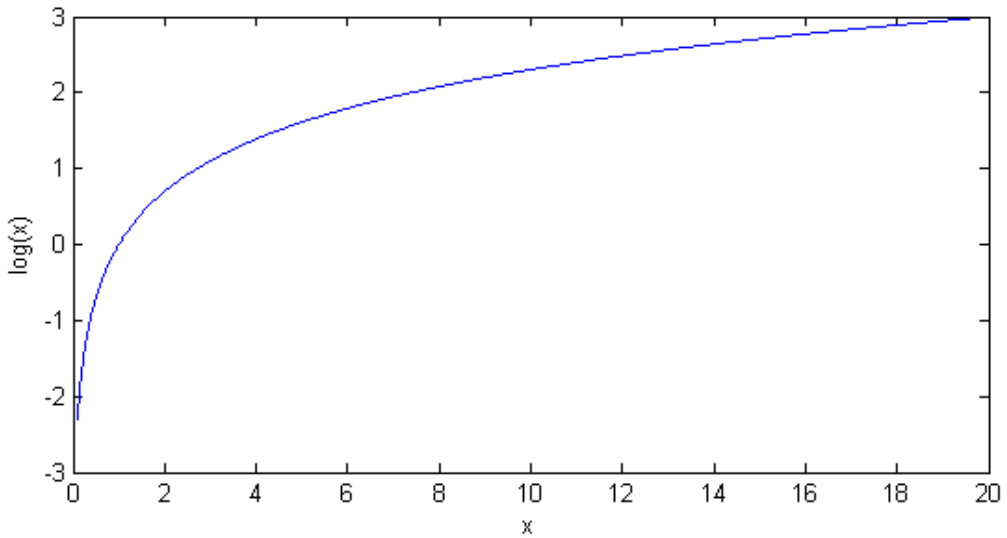
$$\sum_z Q_i(z) = 1, \quad Q_i(z) \geq 0$$

- Then

$$\begin{aligned}
\sum_i \log p(x^{(i)}; \theta) &= \sum_i \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta) \\
&= \sum_i \log \sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \\
&\geq \sum_i \sum_{z^{(i)}} \log Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}
\end{aligned}$$

Proof: Jensen's inequality

- $f(x) = \log x$ is a concave function – $f''(x) = \frac{-1}{x^2} < 0$ over its domain $x \in \mathbb{R}^+$



Proof: Jensen's inequality

- The term

$$\sum_{z^{(i)}} Q_i(z^{(i)}) \left[\frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \right]$$

  is an expectation of the quantity

$$\left[\frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \right]$$

  with respect to $z^{(i)}$ drawn according to the distribution given by Q_i ($z^{(i)} \sim Q_i$)

- By Jensen's inequality

$$f\left(E_{z^{(i)} \sim Q_i} \left[\frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \right] \right) \geq E_{z^{(i)} \sim Q_i} \left[f\left(\frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \right) \right]$$

EM algorithm

- Then, for any set of distributions Q_i the formula

$$\sum_i \log p(x^{(i)};\theta) \quad \geq \sum_i \sum_{z^{(i)}} \log Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}$$

  gives a lower-bound on $\ell(\theta)$
- Assuming that $\ell(\theta)$ increases monotonically, if we have some current guess θ of the parameters, we can try to make lower-bound tight at that value of θ

$$\ell(\theta^{(t)}) \quad = \sum_i \sum_{z^{(i)}} \log Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}$$

EM algorithm

- The Jensen's inequality (in our case) is true if the expectation is taken over a "constant"-valued random variable, what means that

$$\frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} = c$$

  is required, where c does not depend on $z^{(i)}$.
- Therefore, we can choose

$$Q_i(z^{(i)}) \propto p(x^{(i)}, z^{(i)}; \theta)$$

EM algorithm

- Since $\sum_z Q_i(z^{(i)}) = 1$

$$
\begin{aligned}
Q_i(z^{(i)}) &= \frac{p(x^{(i)}, z^{(i)}; \theta)}{\sum_z p(x^{(i)}, z; \theta)} \\
&= \frac{p(x^{(i)}, z^{(i)}; \theta)}{p(x^{(i)}; \theta)} \\
&= p(z^{(i)} | x^{(i)}; \theta)
\end{aligned}
$$

- It means: set the Q_i's to be the posterior distribution of the $z^{(i)}$'s given $x^{(i)}$ and the given parameters θ

EM algorithm

- Step E – *Expectation* – compute the

$$Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)};\theta)$$

for each i

- Step M – *Maximization* – find θ to maximize

$$\theta := arg\max_{\theta} \sum_i \sum_{z^{(i)}} \log Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}$$

- Repeat until convergence

$$|\theta^{(t+1)} - \theta^{(t)}| < \epsilon$$

EM algorithm

- Does the algorithm really converge?
- Assume the parameters $\theta^{(t)}$ and $\theta^{(t+1)}$ are the parameters from two successive iterations of EM
- Prove, that $\ell(\theta^{(t)}) \leq \ell(\theta^{(t+1)})$
- "At the beginning" we would have chosen

$$Q_i^{(t)}(z^{(i)}) := p(z^{(i)}|x^{(i)}; \theta^{(t)})$$

based on the parameters $\theta^{(t)}$

EM algorithm

- It was proven that this choice ensures that

$$\ell(\theta^{(t)}) = \sum_i \sum_{z^{(i)}} \log Q_i^{(t)}(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{Q_i^{(t)}(z^{(i)})}$$

- Thus, to obtain the parameters $\theta^{(t+1)}$

$$\begin{aligned} \ell(\theta^{(t+1)}) &\geq \sum_i \sum_{z^{(i)}} \log Q_i^{(t)}(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta^{(t+1)})}{Q_i^{(t)}(z^{(i)})} \\ &\geq \sum_i \sum_{z^{(i)}} \log Q_i^{(t)}(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{Q_i^{(t)}(z^{(i)})} \\ &= \ell(\theta^{(t)}) \end{aligned}$$

EM algorithm

- But why?
- It is true, that

$$\ell(\theta) \quad \geq \sum_i \sum_{z^{(i)}} \log Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

  hold for any values of Q_i and θ: $Q_i = Q_i^{(t)}$ and $\theta = \theta^{(t+1)}$
- We use the fact that $\theta^{(t+1)}$ is chosen so that

$$\theta^{(t+1)} := arg \max_\theta \sum_i \sum_{z^{(i)}} \log Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

- and that the formula evaluated at $\theta^{(t+1)}$ must be equal or larger than the same formula evaluated at $\theta^{(t)}$

Coming back to GMM...

- For N independent probes $\underline{x}_n = [x_{i1}, x_{i2}, \ldots, x_{iD}]^T$, $i \in \{1, \ldots, m\}$ of size D probability density function of a probe $\underline{x}_i$ in a mixture is given as

$$p(\underline{x}) = p(\underline{x}^{(i)}, z^{(i)}; \theta) = \sum_{j=1}^{k} \pi_j p(\underline{x}^{(i)} | z^{(i)}; \theta) = \sum_{j=1}^{k} \pi_j N(\underline{x}; \underline{\lambda}_j, \Sigma_j),$$

- where $N(\underline{x}; \underline{\lambda}_j, \Sigma_j)$ is probability density function of j-th form k components,

$$N(\underline{x}; \underline{\lambda}, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}} det(\Sigma)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\underline{x} - \underline{\lambda})^T \Sigma^{-1} (\underline{x} - \underline{\lambda})\right\}$$

Coming back to GMM...

- The E-step:

$$w_j^{(i)} = Q_i(z^{(i)} = j) = P(z^{(i)} = j | \underline{x}^{(i)}; \pi, \underline{\lambda}, \Sigma)$$

where $Q_i(z^{(i)} = j)$ denotes the probability of $z^{(i)}$ taking the value j under the distribution Q_i

- The conditional probability of $p(z^{(i)} | \underline{x}^{(i)}; \pi, \underline{\lambda}, \Sigma)$ can be derived using Bayes rule

$$p(z^{(i)} = j | \underline{x}^{(i)}; \theta) = \frac{\pi_j p(\underline{x}^{(i)} | z^{(i)}; \theta_j)}{\sum_{l=1}^{k} \pi_l p(\underline{x}^{(i)} | z^{(i)}; \theta_l)}$$

EM algorithm

- The M-step: maximize the quantity

$$\sum_{i=1}^{m}\sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(\underline{x}^{(i)}, z^{(i)}; \pi, \underline{\lambda}, \Sigma)}{Q_i(z^{(i)})}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{k} Q_i(z^{(i)} = j) \log \frac{p(\underline{x}^{(i)} | z^{(i)} = j; \underline{\lambda}, \Sigma) p(z^{(i)} = j; \pi)}{Q_i(z^{(i)} = j)}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{k} w_j^{(i)} \log \frac{\frac{1}{(2\pi)^{\frac{D}{2}} det(\Sigma_j)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\underline{x}^{(i)} - \underline{\lambda}_j)^T \Sigma_j^{-1} (\underline{x}^{(i)} - \underline{\lambda}_j)\right\} \cdot \pi_j}{w_j^{(i)}}$$

EM algorithm

- The M-step: maximize the quantity with respect to $\underline{\lambda}_l$

$$\nabla_{\underline{\lambda}_l} \sum_{i=1}^{m} \sum_{j=1}^{k} w_j^{(i)} \log \frac{\frac{1}{(2\pi)^{\frac{D}{2}} det(\Sigma_j)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\underline{x}^{(i)} - \underline{\lambda}_j)^T \Sigma_j^{-1} (\underline{x}^{(i)} - \underline{\lambda}_j)\right\} \cdot \pi_j}{w_j^{(i)}}$$

$$= -\nabla_{\underline{\lambda}_l} \sum_{i=1}^{m} \sum_{j=1}^{k} w_j^{(i)} \frac{1}{2} (\underline{x}^{(i)} - \underline{\lambda}_j)^T \Sigma_j^{-1} (\underline{x}^{(i)} - \underline{\lambda}_j)$$

$$= \frac{1}{2} \sum_{i=1}^{m} w_j^{(i)} \nabla_{\underline{\lambda}_l} (2\underline{\lambda}_l^T \Sigma_l^{-1} \underline{x}^{(i)} - \underline{\lambda}_l^T \Sigma_l^{-1} \underline{\lambda}_l)$$

$$= \sum_{i=1}^{m} w_j^{(i)} (\Sigma_l^{-1} \underline{x}^{(i)} - \Sigma_l^{-1} \underline{\lambda}_l)$$

Therefore,

$$\underline{\lambda}_l := \frac{\sum_{i=1}^{m} w_l^{(i)} \underline{x}^{(i)}}{\sum_{i=1}^{m} w_l^{(i)}}$$

EM

- The M-step: maximize the quantity with respect to π_j
- $w_l^{(i)}$ does not depend on π_j, then we need to maximize

$$\sum_{i=1}^{m}\sum_{j=1}^{k} w_j^{(i)} \log \pi_j,$$

where $\sum_{j=1}^{k} \pi_j = 1$

- the Lagrangian

$$\mathcal{L}(\pi) = \sum_{i=1}^{m}\sum_{j=1}^{k} w_j^{(i)} \log \pi_j + \beta(\sum_{j=1}^{k} \pi_j - 1),$$

- taking derivatives $\frac{\partial}{\partial \pi_j}\mathcal{L}(\pi) = \sum_{i=1}^{m} \frac{w_j^{(i)}}{\pi_j} - \beta$

EM

- The M-step: setting the derivatives to 0 and solving

$$\pi_j = \frac{\sum_{i=1}^{m} w_j^{(i)}}{-\beta}$$

- we can see, that $\pi_j \propto \sum_{i=1}^{m} w_j^{(i)}$ and we know, that $\sum_{j=1}^{k} \pi_j = 1$, then

$$-\beta = \sum_{i=1}^{m} \sum_{j=1}^{k} w_j^{(i)} = \sum_{i=1}^{m} 1 = m$$

- therefore,

$$\pi_j := \frac{1}{m} \sum_{i=1}^{m} w_j^{(i)}$$

EM

- Step E

$$p(z^{(i)} = j|\underline{x}^{(i)}; \theta^{(t)}) = \frac{\pi_j p(\underline{x}^{(i)}|z^{(i)}; \theta_j^{(t)})}{\sum_{l=1}^{k} \pi_l p(\underline{x}^{(i)}|z^{(i)}; \theta_l^{(t)})}$$

where

$$p(\underline{x}^{(i)}|z^{(i)}; \theta_j^{(t)}) = N(\underline{x}_i; \underline{\lambda}_j^{(t)}, \Sigma_j^{(t)})$$

EM

- Step M

$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{m} p(z^{(i)} = j|\underline{x}^{(i)}; \theta^{(t)})}{m},$$

$$\underline{\lambda}_j^{(t+1)} = \frac{\sum_{i=1}^{m} p(z^{(i)} = j|\underline{x}^{(i)}; \theta^{(t)})\underline{x}^{(i)}}{\sum_{n=1}^{N} p(z^{(i)} = j|\underline{x}^{(i)}; \theta^{(t)})},$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^{m} p(z^{(i)} = j|\underline{x}^{(i)}; \theta^{(t)})(\underline{x}^{(i)} - \lambda_j^{(i)})(\underline{x}^{(i)} - \lambda_j^{(i)})^T}{\sum_{i=1}^{m} p(z^{(i)} = j|\underline{x}^{(i)}; \theta^{(t)})}.$$

Classification

Based on the created mixture model the data are classified into $j \in \{1, \ldots, k\}$ groups.
For this purpose, the probabilities $\{p(z^{(i)} = 1|\underline{x}^{(i)}; \theta), \ldots, p(z^{(i)} = k|\underline{x}^{(i)}; \theta)\}$, with which the data belong to j-th group

$$p(z^{(i)} = j|\underline{x}^{(i)}; \theta) = \frac{\pi_j p(\underline{x}^{(i)}|z^{(i)}; \theta_j)}{\sum_{l=1}^{k} \pi_l p(\underline{x}^{(i)}|z^{(i)}; \theta_l)}$$

are estimated.
The data is classified to the group fulfilling

$$arg \max_{k=1,\ldots,k} p(z^{(i)} = j|\underline{x}^{(i)}; \theta)$$

Convergence

- as the EM algorithm iterates, the $(t+1)^{th}$ guess θ^{t+1} will never be less likely than the t^{th} guess θ^{t} – monotonicity of the EM algorithm
- local maximum problem

Starting parameters

- Uniform distribution of the mean values and covariance matrix close to the identity matrix
- Random values – bad – why?
- k-means or fuzzy c-means clustering
- Information criteria

Information criteria

- BIC – Bayesian Information Criterion

$$BIC_j = -2p(\underline{x}|\theta) \cong -2\ell(\theta) + \nu_j \log(m),$$

where ν_j – number of free parameters to be estimated, m – the number of data points in X.
The model is selected according to

$$m_{BIC} = arg \min_j BIC_j.$$

- AIC – Akaike information criterion

$$AIC_j = -2\ell(\theta) + 2\nu_j,$$

The model is selected according to

$$m_{AIC} = arg \min_j AIC_j.$$

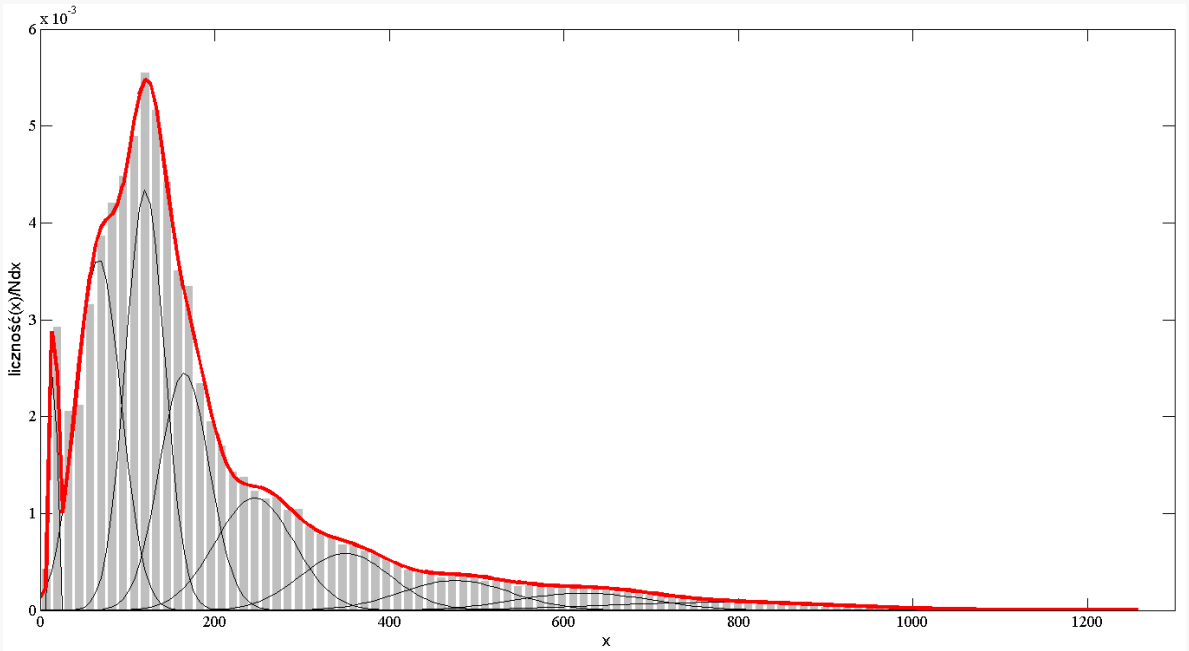
Modification of EM

- SMEM – Split and Merge EM Algorithm – splitting and merging mixture components assuming the constant number of groups
- FSMEM – Free Split and Merge EM Algorithm – enables changes in number of components

Image processing

- Histogram – a graphical representation of the distribution of data

Histogram of an exemplary series with estimated Gaussian mixture model



Results

Clustering results