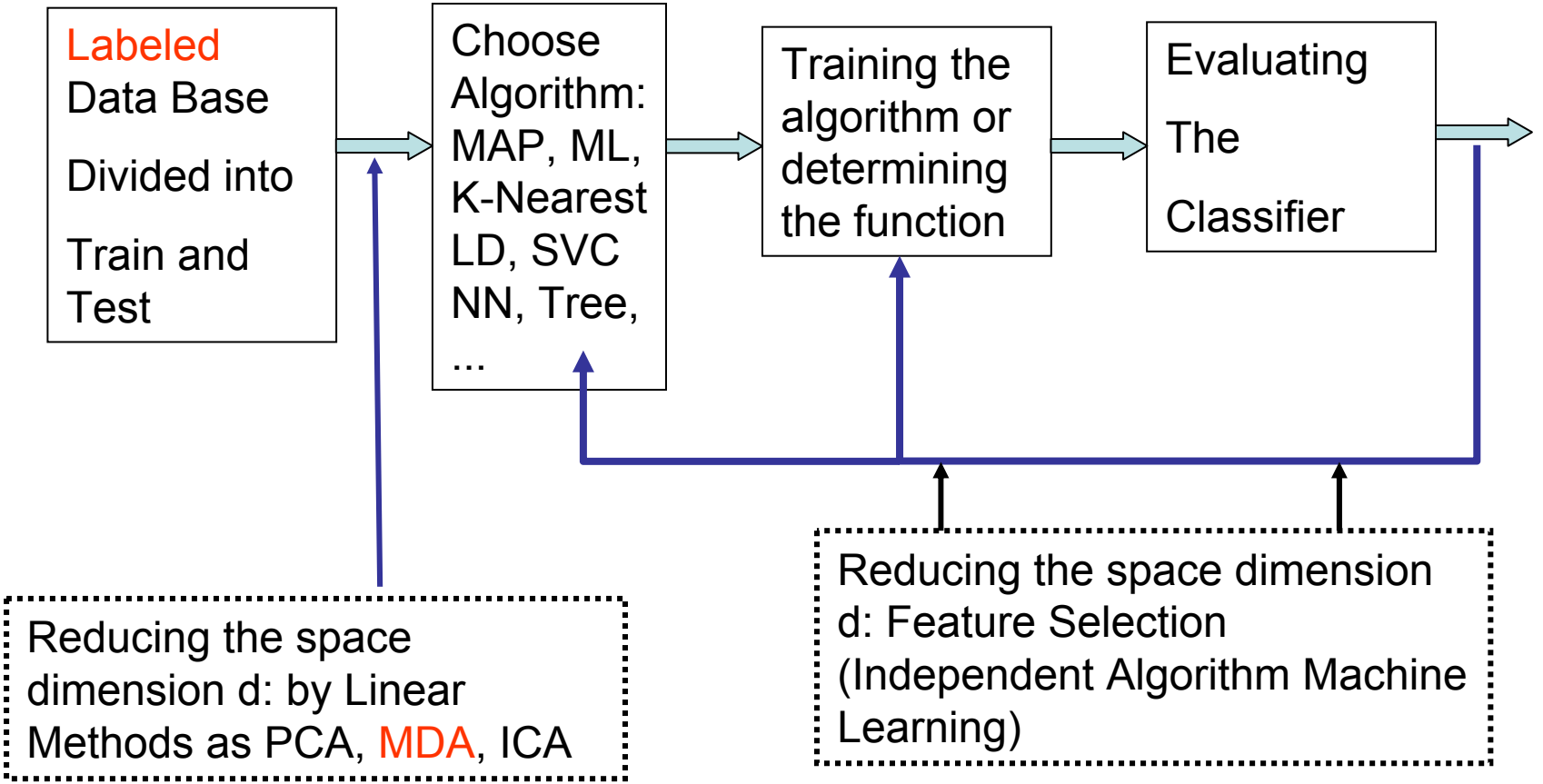
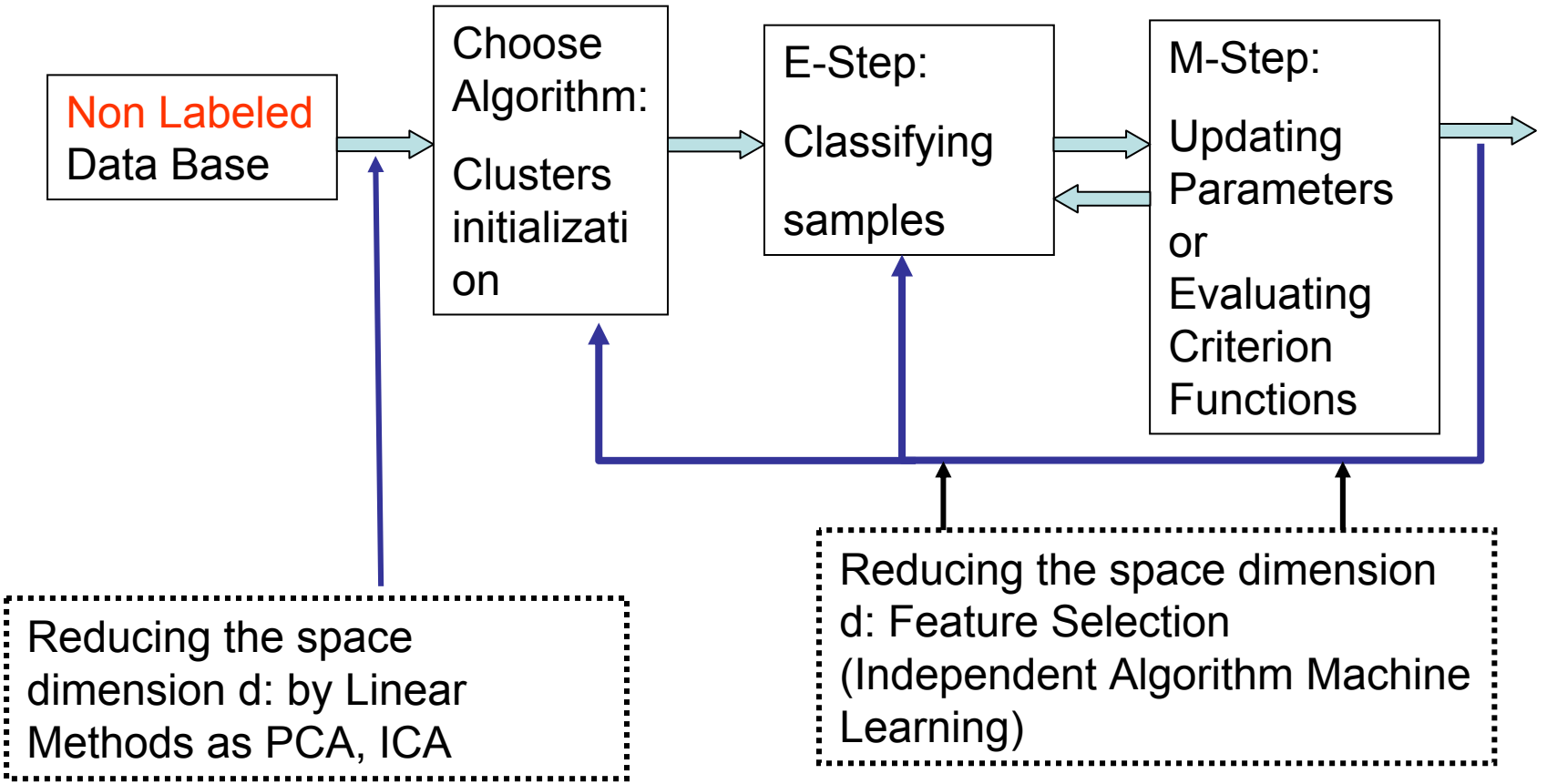
Tema 5:
Aprendizaje NO Supervisado:
CLUSTERING
Unsupervised Learning:
CLUSTERING

Febrero-Mayo 2005

SUPERVISED METHODS:
LABELED Data Base



UNSUPERVISED METHODS:
Non LABELED Data Base



LOOKING FOR STRUCTURE INSIDE THE DATA

- *Parametric Methods: They assume some f.d.p. for the clusters.*
- *Non Parametric Methods: Formal Clustering Procedures*

INDEX (*Parametric Methods*)

1 MIXTURE DENSITIES AND IDENTIFIABILITY

2 MAXIMUM LIKELIHOOD ESTIMATES: EM

3 K-Means Clustering

1 MIXTURE DENSITIES AND IDENTIFIABILITY

Assumptions:

1. The samples come from a known number c of classes
2. Prior probabilities for each class are known (Mixing Parameters).
3. The form of the class-conditional probabilities densities are known
4. The values for parameters are unknown
5. The category labels are unknown: UNSUPERVISED

$$\Pr\{\omega_j\};\quad j=1..c$$

$$f_{\mathbf{x}|\omega_j,\theta_j}\{\mathbf{x}|\omega_j,\theta_j\}$$

$$\theta_j;\quad j=1..c$$

1 MIXTURE DENSITIES AND IDENTIFIABILITY

MIXTURE DENSITY:

$$f_{\mathbf{x}|\theta}\left\{\mathbf{x}\middle|\theta\right\} = \sum_{j=1}^{c} f_{\mathbf{x}|\omega_j,\theta_j}\left\{\mathbf{x}\middle|\omega_j,\theta_j\right\}\Pr\left\{\omega_j\right\}$$

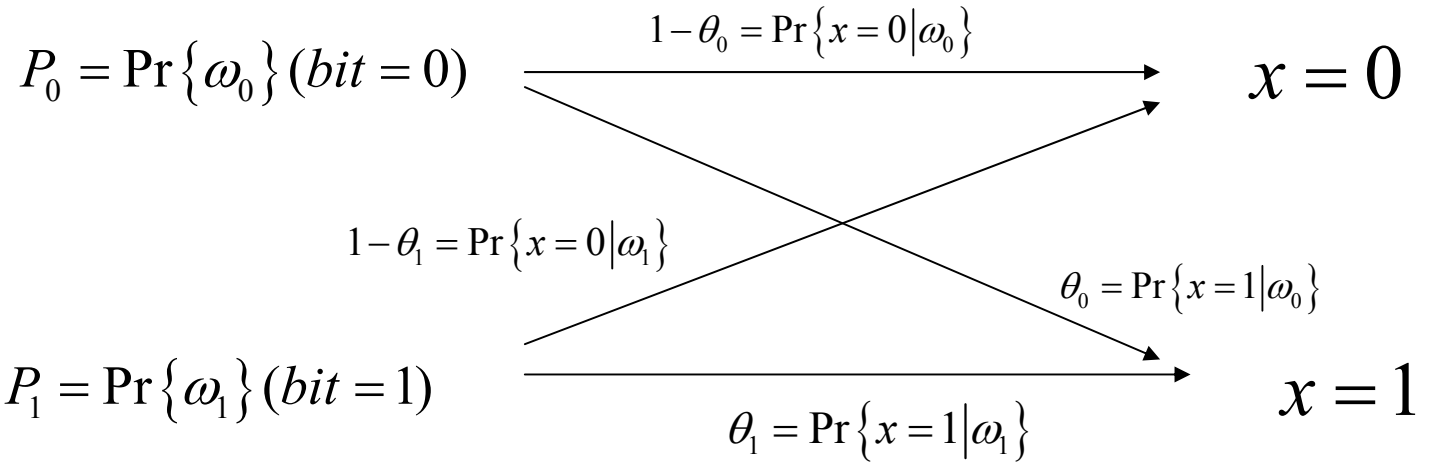
1. For the moment it is assumed that only parameter vector θ is unknown.
2. Necessary conditions for identifiability:

$$f_{\mathbf{x}|\theta}\left\{\mathbf{x}\middle|\boldsymbol{\theta}\right\} \neq f_{\mathbf{x}|\theta}\left\{\mathbf{x}\middle|\boldsymbol{\theta}'\right\} \qquad \boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_c \end{pmatrix}$$

1 MIXTURE DENSITIES AND IDENTIFIABILITY

Example: identifiability problem:

BINARY (SYMMETRIC) CHANNEL $P_0 + P_1 = 1$



1 MIXTURE DENSITIES AND IDENTIFIABILITY

Example: identifiability problem:
BINARY (SYMMETRIC) CHANNEL
Parameter Vector:

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix}$$

MIXTURE DENSITY (PROBABILITY)

$$\Pr\{\mathbf{x}|\theta\} = P_0\theta_0^{\,x}\left(1-\theta_0\right)^{1-x} + P_1\theta_1^{\,x}\left(1-\theta_1\right)^{1-x}$$

2 MAXIMUM LIKELIHOOD ESTIMATES

Likelihood of the statistical independent observed samples: $D = \{\mathbf{x}_1, \mathbf{x}_2, ..\mathbf{x}_n\}$

$$f_D\left(D|\boldsymbol{\theta}\right) = \prod_{k=1}^{n} f_{\mathbf{x}_k}\left(\mathbf{x}_k|\boldsymbol{\theta}\right); \qquad l = \sum_{k=1}^{n} \ln f_{\mathbf{x}_k}\left(\mathbf{x}_k|\boldsymbol{\theta}\right)$$

$$f_{\mathbf{x}_k}\left(\mathbf{x}_k|\boldsymbol{\theta}\right) = \sum_{j=1}^{c} f_{\mathbf{x}_k}\left(\mathbf{x}_k|\boldsymbol{\theta}_j\omega_j\right)\Pr\left(\omega_j\right)$$

Assuming statistical independence between $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$, ML solutions is one of the multiple solutions of:

$$\nabla_{\boldsymbol{\theta}_i} l = \sum_{k=1}^{n} \Pr\left\{\omega_i|\mathbf{x}_k, \boldsymbol{\theta}\right\} \nabla_{\boldsymbol{\theta}_i} \ln f_{\mathbf{x}_k}\left(\mathbf{x}_k|\omega_i, \boldsymbol{\theta}_i\right) = 0; \qquad i = 1..c$$

2 MAXIMUM LIKELIHOOD ESTIMATES

Demo:

$$\nabla_{\boldsymbol{\theta}_i} l = \sum_{k=1}^{n} \frac{1}{f_{\mathbf{x}_k}\left(\mathbf{x}_k \middle| \boldsymbol{\theta}\right)} \nabla_{\boldsymbol{\theta}_i} f_{\mathbf{x}_k}\left(\mathbf{x}_k \middle| \boldsymbol{\theta}\right) =$$

$$\sum_{k=1}^{n} \frac{1}{f_{\mathbf{x}_k}\left(\mathbf{x}_k \middle| \boldsymbol{\theta}\right)} \nabla_{\boldsymbol{\theta}_i} \left(\sum_{j=1}^{c} f_{\mathbf{x}_k}\left(\mathbf{x}_k \middle| \omega_j, \boldsymbol{\theta}_i\right) \Pr(\omega_j) \right) =$$

$$\sum_{k=1}^{n} \frac{1}{f_{\mathbf{x}_k}\left(\mathbf{x}_k \middle| \boldsymbol{\theta}\right)} \nabla_{\boldsymbol{\theta}_i} \left(f_{\mathbf{x}_k}\left(\mathbf{x}_k \middle| \omega_i, \boldsymbol{\theta}_i\right) \Pr(\omega_i) \right) =$$

$$\sum_{k=1}^{n} \Pr\left\{\omega_i \middle| \mathbf{x}_k, \boldsymbol{\theta}\right\} \nabla_{\boldsymbol{\theta}_i} \ln f_{\mathbf{x}_k}\left(\mathbf{x}_k \middle| \omega_i, \boldsymbol{\theta}_i\right) = 0; \quad i = 1..c$$

2 MAXIMUM LIKELIHOOD ESTIMATES

Generalizing to the unknown prior probability case: (No demo is included here)

1. To compute prior probability estimates
2. To compute vector parameter estimates
3. To compute conditioned probability for classes.

$$\hat{\Pr}\{\omega_i\} = \frac{1}{n}\sum_{k=1}^{n} \hat{\Pr}\left\{\omega_i \left| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right.\right\}$$

$$\sum_{k=1}^{n} \hat{\Pr}\left\{\omega_i \left| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right.\right\} \nabla_{\boldsymbol{\theta}_i} \ln f_{\mathbf{x}_k}\left(\mathbf{x}_k \left| \omega_i, \hat{\boldsymbol{\theta}}_i\right.\right); \quad i = 1..c$$

$$\hat{\Pr}\left\{\omega_i \left| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right.\right\} = \frac{f_{\mathbf{x}_k}\left(\mathbf{x}_k \left| \omega_i, \hat{\boldsymbol{\theta}}_i\right.\right)\hat{\Pr}\{\omega_i\}}{\sum_{j=1}^{c} f_{\mathbf{x}_k}\left(\mathbf{x}_k \left| \omega_j, \hat{\boldsymbol{\theta}}_j\right.\right)\hat{\Pr}\{\omega_j\}}$$

2 MAXIMUM LIKELIHOOD ESTIMATES

For Gaussian Distributions:

$$\ln\left(f_{\mathbf{x}_k}\left(\mathbf{x}_k\left|\omega_i,\boldsymbol{\theta}_i\right.\right)\right)=\ln\left(\frac{\left|\boldsymbol{\Sigma}_i^{-1}\right|^{\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}}\right)-\tfrac{1}{2}\left(\mathbf{x}_k-\boldsymbol{\mu}_i\right)^T\boldsymbol{\Sigma}_i^{-1}\left(\mathbf{x}_k-\boldsymbol{\mu}_i\right)$$

Parameters to estimate:

$$\boldsymbol{\theta}=\left(\boldsymbol{\theta}_1,..;\boldsymbol{\theta}_c\right)$$

$$\boldsymbol{\theta}_i=\left(\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i\right)$$

2 MAXIMUM LIKELIHOOD ESTIMATES

ML is solved appliying the SOFT Expectation-Maximization algorithm:
Soft Assignment. Iterations stop when the p.d.f. does not vary.

1. Expectation (E-Step)

$$\hat{\Pr}\left\{\omega_i \middle| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right\} = \frac{\left|\hat{\boldsymbol{\Sigma}}_i^{-1}\right|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i\right)^T \hat{\boldsymbol{\Sigma}}_i^{-1}\left(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i\right)\right)\hat{\Pr}\left\{\omega_i\right\}}{\sum_{j=1}^{c}\left|\hat{\boldsymbol{\Sigma}}_j^{-1}\right|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_j\right)^T \hat{\boldsymbol{\Sigma}}_j^{-1}\left(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_j\right)\right)\hat{\Pr}\left\{\omega_j\right\}}$$

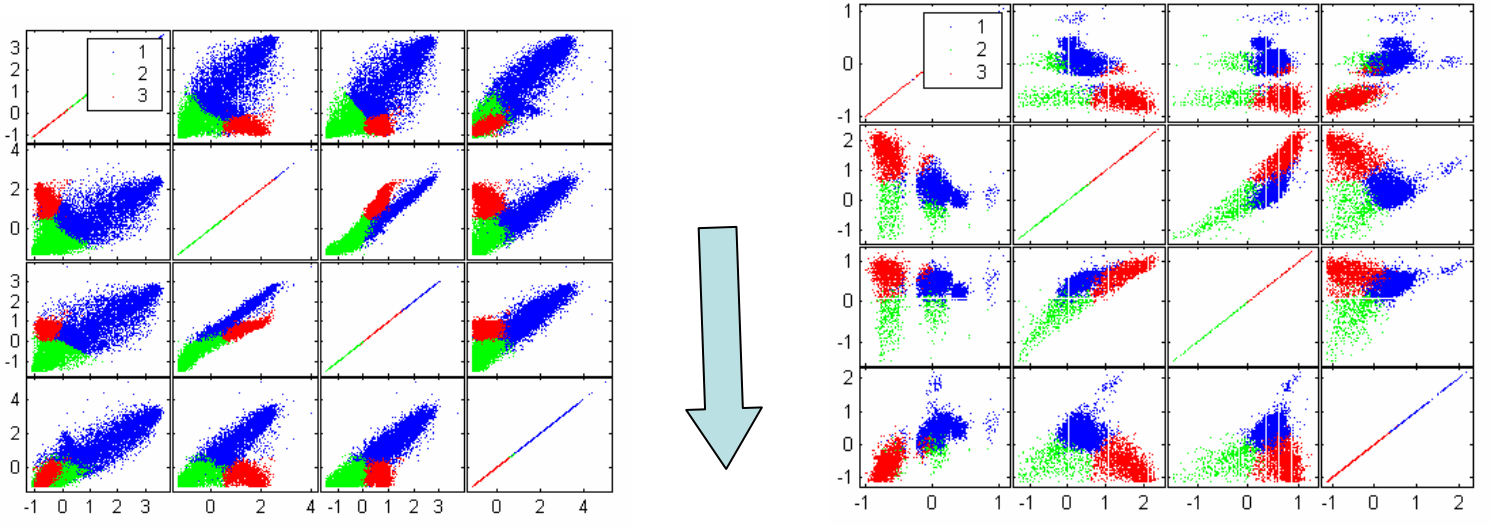
2. Maximization (M-Step)

$$\hat{\Pr}\left\{\omega_i\right\} = \frac{1}{n}\sum_{k=1}^{n}\hat{\Pr}\left\{\omega_i \middle| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right\};\quad \hat{\boldsymbol{\mu}}_i = \frac{\sum_{k=1}^{n}\hat{\Pr}\left\{\omega_i \middle| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right\}\mathbf{x}_k}{\sum_{k=1}^{n}\hat{\Pr}\left\{\omega_i \middle| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right\}};\quad \hat{\boldsymbol{\Sigma}}_i = \frac{\sum_{k=1}^{n}\hat{\Pr}\left\{\omega_i \middle| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right\}\left(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i\right)\left(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i\right)^T}{\sum_{k=1}^{n}\hat{\Pr}\left\{\omega_i \middle| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right\}}$$

2 MAXIMUM LIKELIHOOD ESTIMATES

Pb: Starting Point: Brain Images;

Full DataBase vs Labeled Data Base Pr>0.95



$$\boldsymbol{\mu}_i[0]; \quad \dim: dx1$$

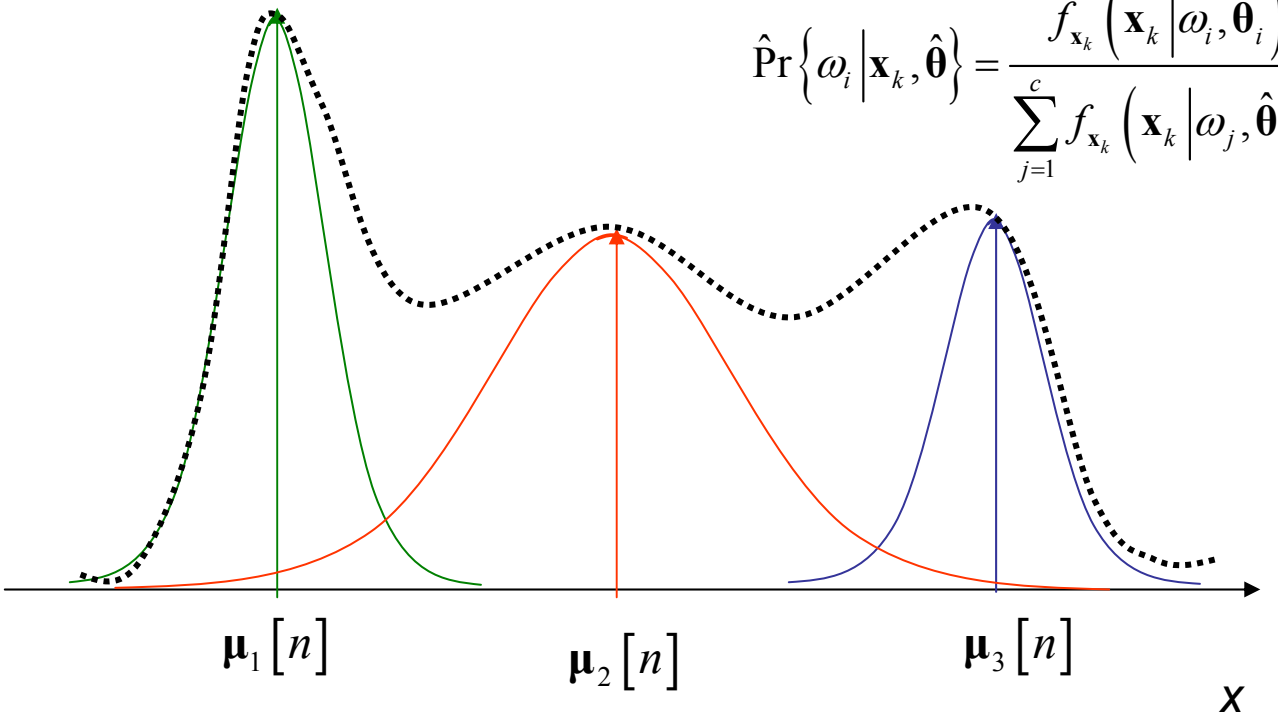
$$\Sigma_i[0]; \quad \dim: dxd$$

$$\Pr\{\omega_i\} \quad i = 1,2,3$$

2 MAXIMUM LIKELIHOOD ESTIMATES

E-step: For a given x_k estimate:

$$\hat{\Pr}\left\{\omega_i \left| \mathbf{x}_k, \hat{\boldsymbol{\theta}} \right.\right\} = \frac{f_{\mathbf{x}_k}\left(\mathbf{x}_k \left| \omega_i, \hat{\boldsymbol{\theta}}_i \right.\right)\hat{\Pr}\{\omega_i\}}{\sum_{j=1}^{c} f_{\mathbf{x}_k}\left(\mathbf{x}_k \left| \omega_j, \hat{\boldsymbol{\theta}}_j \right.\right)\hat{\Pr}\{\omega_j\}}$$



M-STEP: Parameters are updated (ML estimation)

3. K-Means Clustering

HARD Classification: Simplification of the ML (EM) estimates for a Normal Multivariable (Optimum for CASE 1 Multivariable Gaussian Variable seen with MAP). $\boldsymbol{\theta}_i = \boldsymbol{\mu}_i \qquad \boldsymbol{\Sigma}_i = \sigma^2 \mathbf{I}$

$$\hat{\Pr}\left\{\omega_i \left| \mathbf{x}_k, \hat{\boldsymbol{\mu}} \right.\right\} = \begin{cases} 1 & d_e\left(\mathbf{x}_k, \hat{\boldsymbol{\mu}}_i\right) < d_e\left(\mathbf{x}_k, \hat{\boldsymbol{\mu}}_j\right); j \neq i \\ 0 & other \end{cases}$$

$$\hat{\Pr}\left\{\omega_i\right\} = \frac{1}{n}\sum_{k=1}^{n} \hat{\Pr}\left\{\omega_i \left| \mathbf{x}_k, \hat{\boldsymbol{\theta}} \right.\right\} = \frac{n_k}{n}$$

Centroid

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{n_i}\sum_{k=1}^{n_i} \mathbf{x}_k$$

3. K-Means Clustering

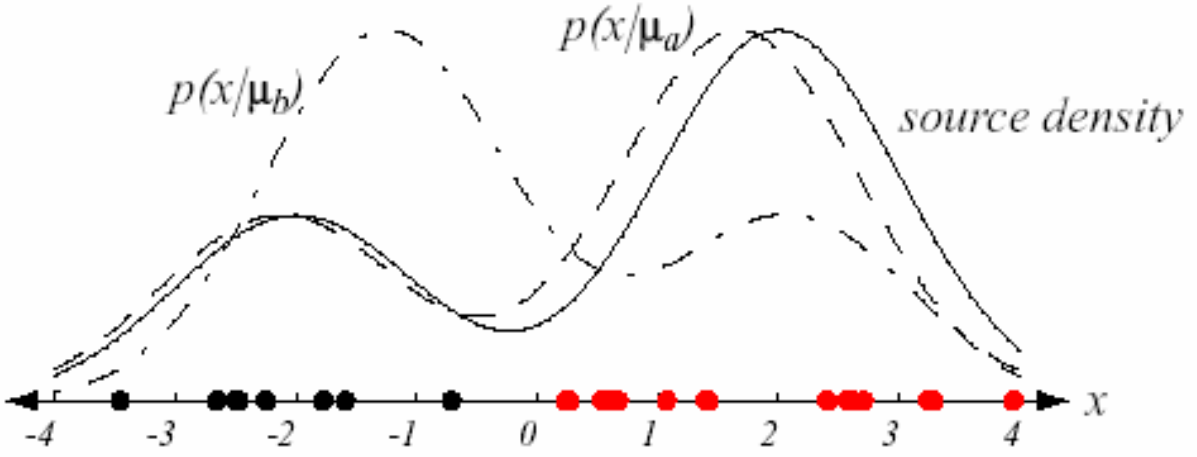
- *K-Means Clustering*

Algorithm 1 (K-means clustering)

1 **begin initialize** $n, c, \mu_1, \mu_2, \ldots, \mu_c$
2 **do** classify n samples according to nearest μ_i
3 recompute μ_i
4 **until** no change in μ_i
5 **return** $\mu_1, \mu_2, \ldots, \mu_c$
6 **end**

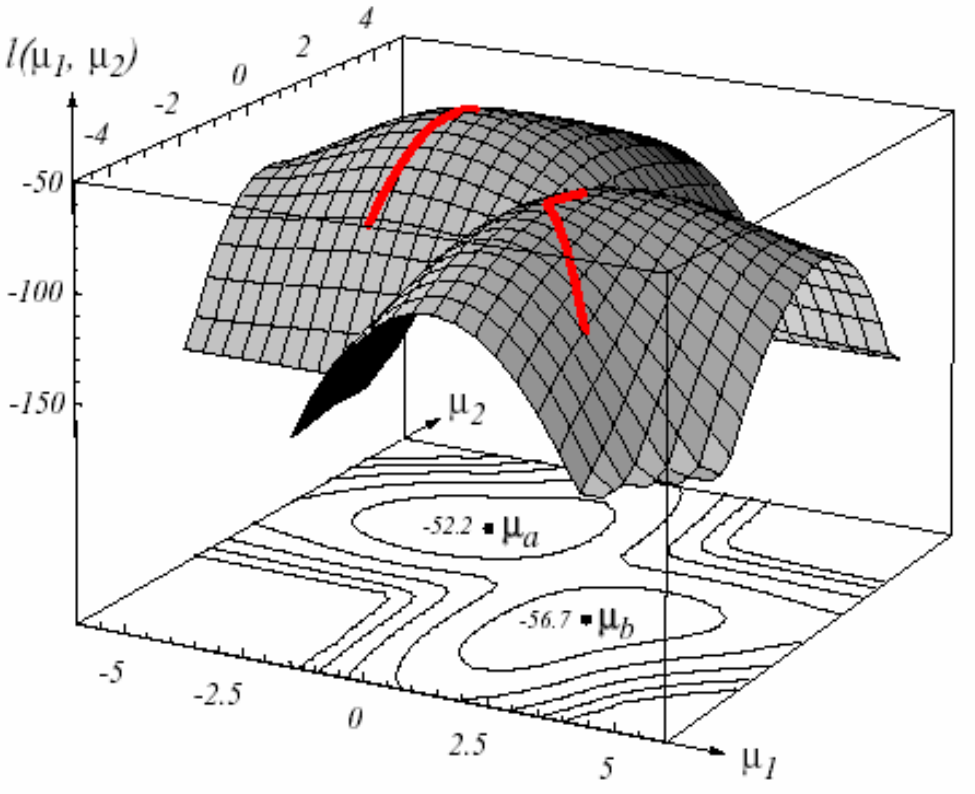
3. K-Means Clustering

- *K-Means Clustering*



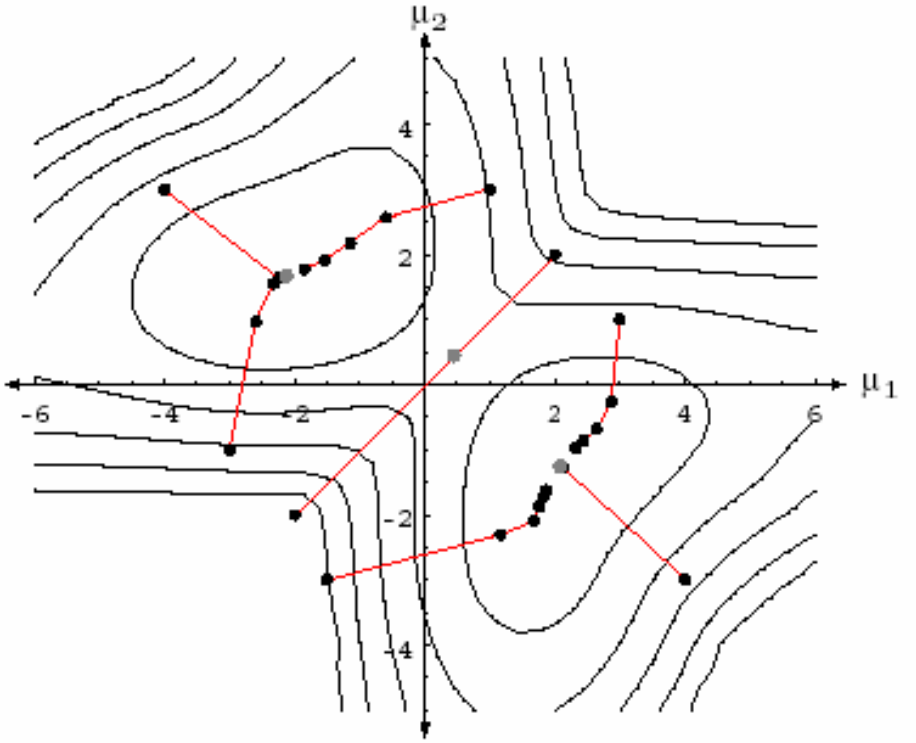
3. K-Means Clustering

- *K-Means Clustering*

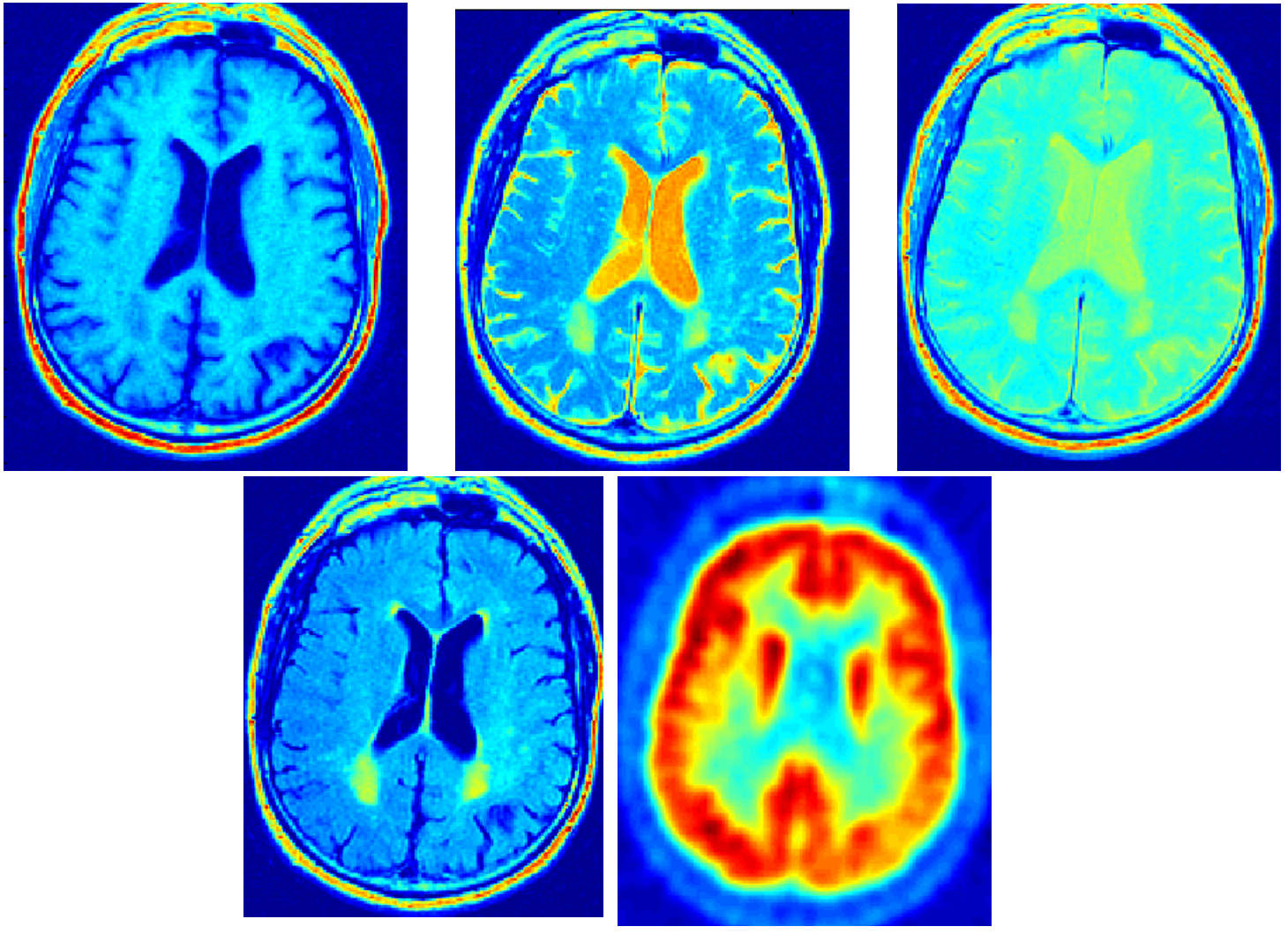


3. K-Means Clustering

- *K-Means Clustering*

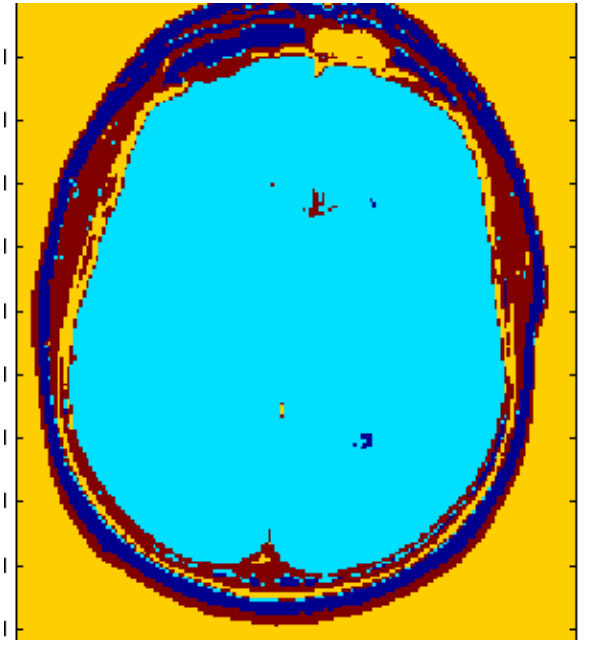


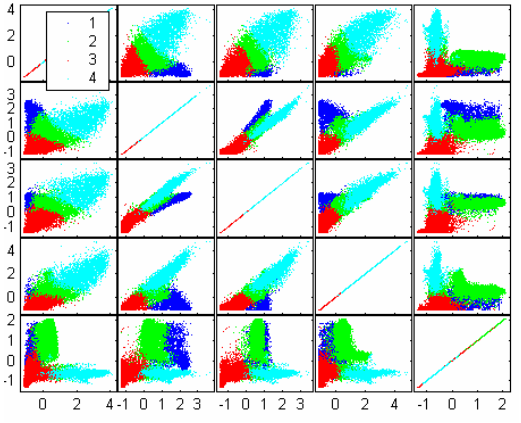
Brain Images

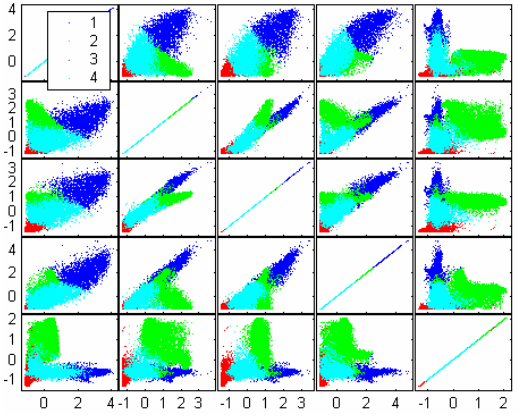


Brain Images: K-Means

Different Starting Points

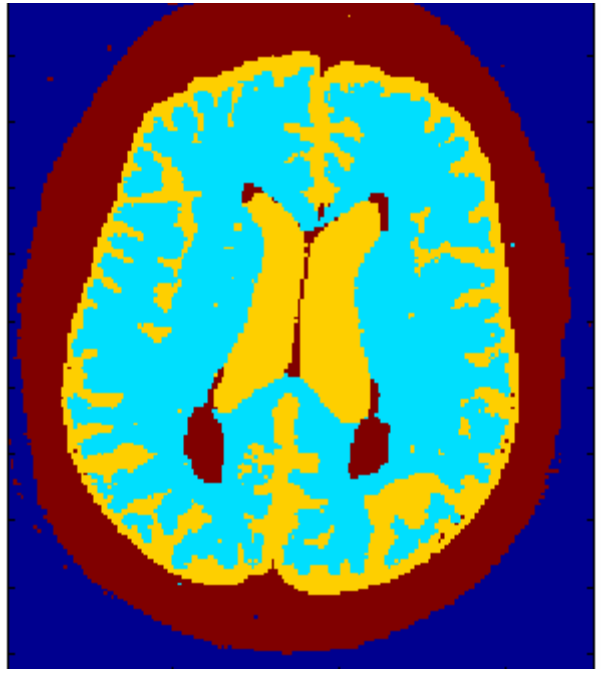




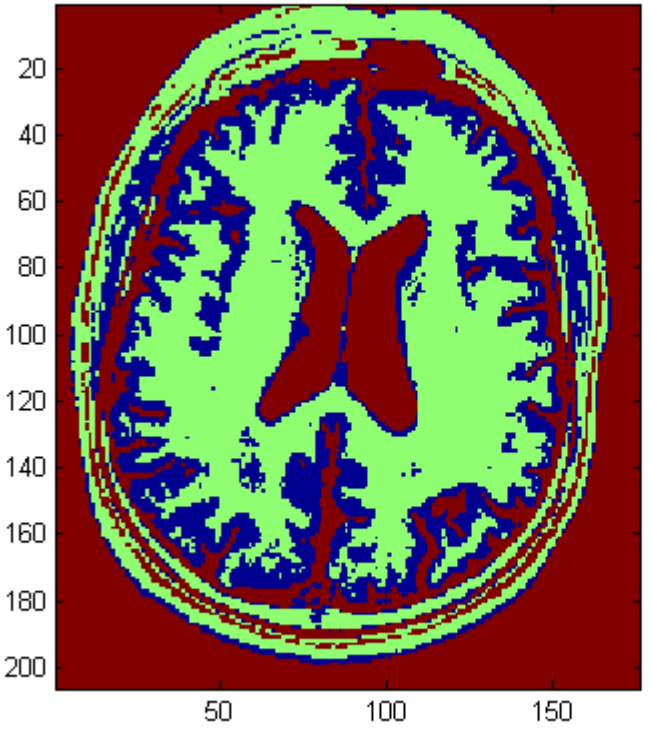


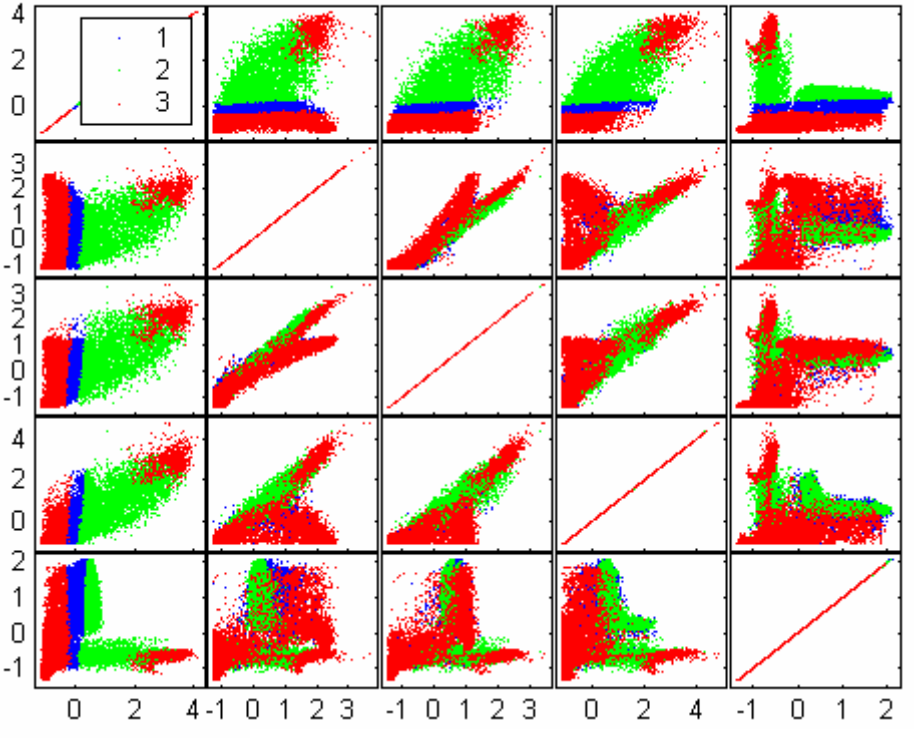
Brain Images: Expectation-Maximization

Different Starting Points



Brain Images: NN





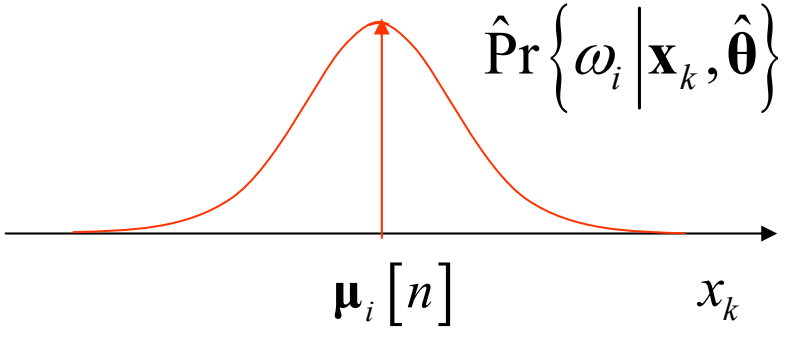
3. K-Means Clustering

APPLICATION: Vector Quantization of a n-dimensional real valued vector. *See: Proakis: "Digital Communications" Chapter 3: Source Coding.*

FUZZY K-Means Soft Classification. *b* is a free blending parameter

$$J_{Fuzzy} = \sum_{i=1}^{c}\sum_{j=1}^{n}\left(\hat{\Pr}\left\{\omega_i \middle| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right\}\right)^b d_e\left(\mathbf{x}_j, \hat{\boldsymbol{\mu}}_i\right)$$

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{k=1}^{n}\left(\hat{\Pr}\left\{\omega_i \middle| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right\}\right)^b \mathbf{x}_k}{\sum_{k=1}^{n}\left(\hat{\Pr}\left\{\omega_i \middle| \mathbf{x}_k, \hat{\boldsymbol{\theta}}\right\}\right)^b}$$



INDEX: *Formal Clustering Procedures*

1 INTRODUCTION:
FORMAL CLUSTERING PROCEDURES
2 SIMILARITY MEASURES
3 CRITERION FUNCTIONS
4 ITERATIVE OPTIMIZATION
5 CONCLUSIONS

1. INTRODUCTION

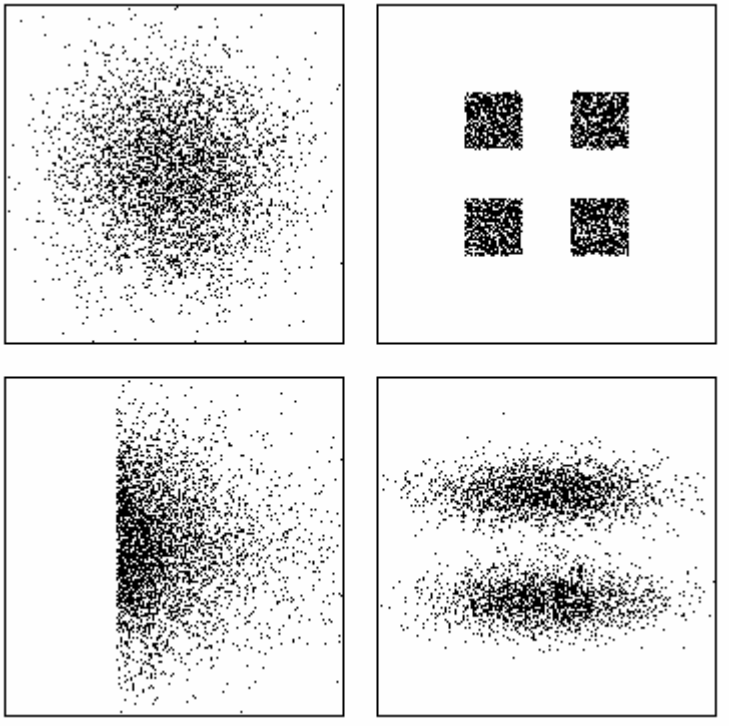
- *Clusters* may form clouds of points in a d-dimensional space.
- *Normal Distribution*: Sample Mean and Sample Covariance Matrix form a Sufficient Statistics
- *Mean Sample* **m**: Locates de Center of gravity of the cloud and it best represents all of the data in the sense of minimizing the sum of squared distances from **m** to the samples.
- *Sample Covariance Matrix* **C**: denotes the amount the data scatters along various directions around **m**.

1. INTRODUCTION

Sample mean vector and Sample Covariance Matrix aren't a suficcient statistical in a general case: Distributions with identical Mean and Covariance:

$$\mathbf{m}_N = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k$$

$$\mathbf{C} = \frac{1}{N-1}\sum_{k=1}^{N}(\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T$$



1. INTRODUCTION

Formal Clustering Procedures: Two Key Steps

- Data are grouped in clusters or groups of data points that posses strong internal similarities.
- A Criterion Function is used to seek the grouping that extremizes it. To evaluate the partitioning of a set of samples into clusters, the similarity is measured between samples.

2. SIMILARITY MEASURES

Similarity is measured using distance between samples

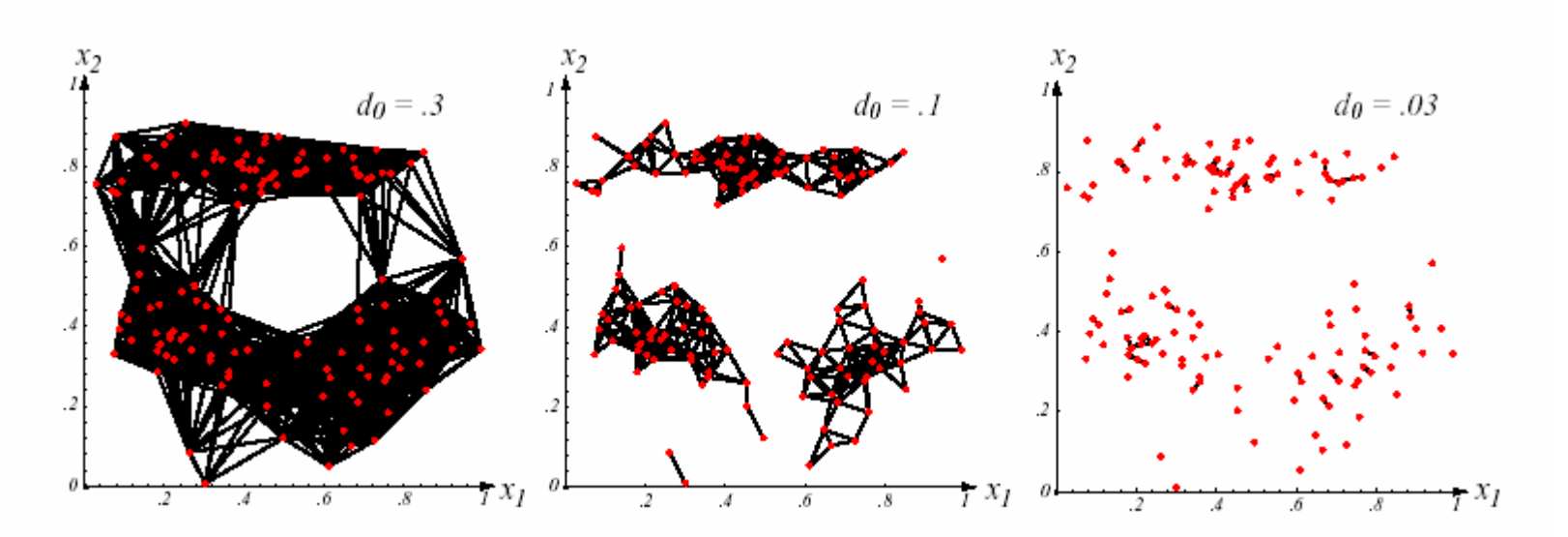
- Example: Euclidean distance $d(x_i, x_j)$.

$$d_e(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{n=1}^{d} \left(x_i[n] - x_j[n]\right)^2} = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|$$

- Two samples belongs to the same cluster if $d(x_i, x_j) < d_o$.
- Threshold d_o is critical.

2. SIMILARITY MEASURES

Distance threshold affects the number and size of clusters:

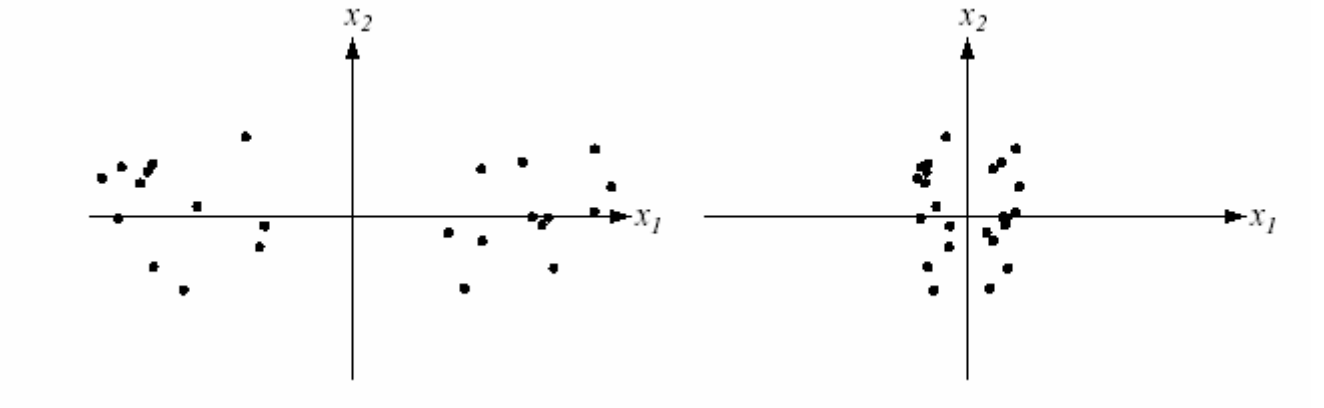


typical within clusters distance $< d_0 <$ typical between clusters distance

2. SIMILARITY MEASURES

Euclidean distance d_{ij}.

- Clusters are invariant to Rotation.
- Clusters are invariant to Translation.
- Clusters are variant to Linear Transformations in general.



2. SIMILARITY MEASURES

Normalization prior to clustering.

- Each feature is translated to have zero mean
- Each feature is scaled to have unit variance.
  - (These two previous actions are recommended with Neural Nets).
- PCA Principal Components Analysis (Axes coincide with the eigenvectors of the sample covariance matrix).

AFTER NORMALIZATION AND PCA , CLUSTERS ARE INVARIANT TO DISPLACEMENTS, SCALE CHANGE AND ROTATIONS.

2. SIMILARITY MEASURES

Other Metrics.

- Minkowski Distance

$$d_q(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{n=1}^{d} \left(x_i[n] - x_j[n] \right)^q \right)^{\frac{1}{q}}$$

- Mahalanobis Distance

$$d_M^2(\mathbf{x}_i, \mathbf{x}_j) = \left(\mathbf{x}_i - \mathbf{x}_j \right)^{\mathbf{T}} \boldsymbol{\Sigma}^{-1} \left(\mathbf{x}_i - \mathbf{x}_j \right)$$

2. SIMILARITY MEASURES

Similarity Functions:

- It compares two vectors

$$s_e(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}$$

- It is invariant to Rotation and Dilation
- It is no invariant to translation and general linear transformation

2. SIMILARITY MEASURES

If the found clusters are used to a posterior problem of classification:

- Metric (distance) is used as classification criteria

or

- Similarity function is used as classification criteria

3. CRITERION FUNCTIONS

Criterion Functions for Clustering:

- Initial Set $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

- Partition into exactly c subsets.

$$D_1, D_2, \ldots, D_c$$

- Objective: To find the partition that extremizes the criterion function

3. CRITERION FUNCTIONS

3.1 Criterion Function Sum Of Squared Error Criterion:

$$J_e = \sum_{i=1}^{c} \sum_{\mathbf{x} \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2$$

- $\mathbf{m}_i$ is the best representative of the samples in D_i.
- It is appropriated when the clusters form compact clouds and uniform number of samples per cluster.

3. CRITERION FUNCTIONS

Related Minimum Variance Criteria:

- J_e

$$J_e = \frac{1}{2}\sum_{i=1}^{c} n_i \bar{s}_i; \qquad \bar{s}_i = \frac{1}{n_i^2}\sum_{\mathbf{x}\in D_i}\sum_{\mathbf{x}'\in D_i} \left\|\mathbf{x}-\mathbf{x}'\right\|^2$$

- Suggestion to obtain other criterion function:

$$\bar{s}_i = \max_{\mathbf{x},\mathbf{x}'\in D_i} d_e(\mathbf{x},\mathbf{x}'); \qquad \bar{s}_i = \frac{1}{n_i^2}\sum_{\mathbf{x}\in D_i}\sum_{\mathbf{x}'\in D_i} s_e(\mathbf{x},\mathbf{x}'); \qquad \bar{s}_i = \min_{\mathbf{x},\mathbf{x}'\in D_i} s_e(\mathbf{x},\mathbf{x}');$$

3. CRITERION FUNCTIONS

3.2 Scatter Criteria:

- *Mean Vectors and Scatter matrices used in clustering criteria*
- *Mean Vector for the i cluster*

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{x}$$

- *Total mean vector*

$$\mathbf{m} = \frac{1}{n} \sum_{\mathbf{x} \in D} \mathbf{x} = \frac{1}{n} \sum_{i=1}^{c} n_i \mathbf{m}_i$$

- *Scatter matrix for the I cluster*

$$\mathbf{S}_i = \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T$$

- *Within-cluster scatter matrix*

$$\mathbf{S}_W = \sum_{i=1}^{c} \mathbf{S}_i$$

- *Between-cluster scatter matrix*

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$$

- *Total Scatter Matrix*

$$\mathbf{S}_T = \frac{1}{n} \sum_{\mathbf{x} \in D} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T = \mathbf{S}_W + \mathbf{S}_B$$

3. CRITERION FUNCTIONS

3.2 Scatter Criteria: TRACE CRITERION

- It measures the square of the scattering radius
- Minimize the trace of the Within Cluster Scatter Matrix

$$Tr\left[\mathbf{S}_W\right] = \sum_{i=1}^{c} Tr\left[\mathbf{S}_i\right] = \sum_{i=1}^{c} \sum_{\mathbf{x} \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2 = J_e$$

- **¡¡¡¡ It results function J_e.**
- It is equivalent to maximize between cluster scattering matrix trace.

$$Tr\left[\mathbf{S}_W\right] = Tr\left[\mathbf{S}_T\right] - Tr\left[\mathbf{S}_B\right] \qquad Tr\left[\mathbf{S}_B\right] = \sum_{i=1}^{c} n_i \left\| \mathbf{m}_i - \mathbf{m} \right\|^2$$

3. CRITERION FUNCTIONS

3.2 Scatter Criteria: DETERMINANT CRITERION

- It measures the square of the scattering volume.
- S_B is singular if c<=d; rank(S_B)<=c-1
- S_W is singular if n-c<d
- Assuming n>d+c

$$J_d = |\mathbf{S}_W| = \left| \sum_{i=1}^{c} \mathbf{S}_i \right|$$

- ¡¡¡ It no changes if the axes are scaled

3. CRITERION FUNCTIONS

3.2 Scatter Criteria: Invariant Criteria

- Eigenvalues of $\text{inv}(S_W)S_B$ are invariant to nonsigular linear transformations of the data.

$$\max : Tr\left[\mathbf{S}_W{}^{-1}\mathbf{S}_B\right] = \sum_{i=1}^{d}\lambda_i; \quad \frac{|\mathbf{S}_W|}{|\mathbf{S}_T|} = \prod_{i=1}^{d}\frac{1}{1+\lambda_i}$$

- Proposed Criteria $\min : J_f = Tr\left[\mathbf{S}_T{}^{-1}\mathbf{S}_W\right] = \sum_{i=1}^{d}\frac{1}{1+\lambda_i}$

They are equivalent for c=2

3. CRITERION FUNCTIONS

3.2Invariant Criteria

$$\lambda_1,..,\lambda_i,..,\lambda_d = eigenvalues(\mathbf{S}_W{}^{-1}\mathbf{S}_B) \Rightarrow$$

$$\frac{1}{1+\lambda_1},..,\frac{1}{1+\lambda_i},..,\frac{1}{1+\lambda_d} = eigenvalues(\mathbf{S}_T{}^{-1}\mathbf{S}_B)$$

- Demo:

$$\mathbf{S}_B\mathbf{v}_i = \lambda_i\mathbf{S}_W\mathbf{v}_i \Rightarrow$$

$$\mathbf{S}_T\mathbf{v}_i = \mathbf{S}_B\mathbf{v}_i + \mathbf{S}_W\mathbf{v}_i = \lambda_i\mathbf{S}_W\mathbf{v}_i + \mathbf{S}_W\mathbf{v}_i = (1+\lambda_i)\mathbf{S}_W\mathbf{v}_i \Rightarrow$$

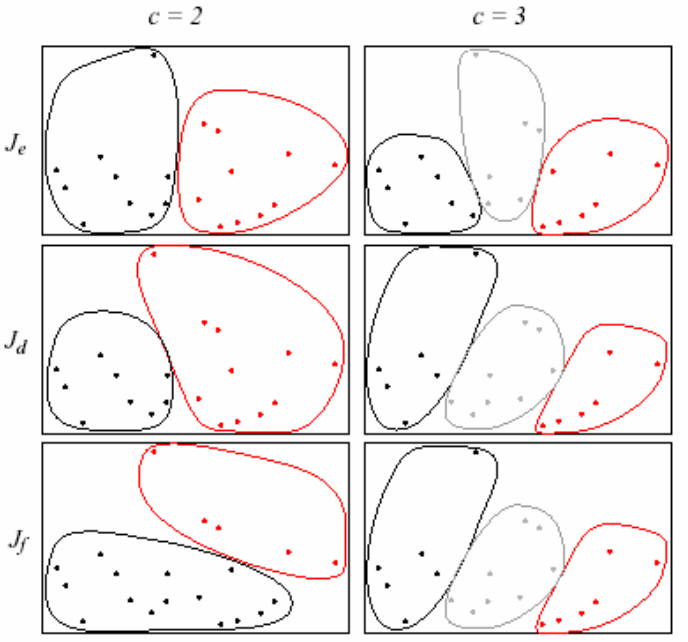
$$\mathbf{v}_i = (1+\lambda_i)\mathbf{S}_T{}^{-1}\mathbf{S}_W\mathbf{v}_i \Rightarrow$$

$$\mathbf{S}_T{}^{-1}\mathbf{S}_W\mathbf{v}_i = \frac{1}{1+\lambda_i}\mathbf{v}_i$$

3. CRITERION FUNCTIONS

3.2 Scatter Criterion: Invariant Criteria

- Trace Criteria.
- Determinant Criteria.
- Invariant Criteria.



CLUSTERING PROCEDURES

CONCLUSIONS

- Underlying Model: assumes that samples form c fairly well separated clouds of points.
- S_W measures the compactness of these clouds.

- Problem: Computational complexity to evaluate the overall number of possibilities in partitioning is impracticable.

4 ITERATIVE OPTIMIZATION

- Direct partitioning: $c^n/c!$
- Practical solution:
- Initiate with some reasonable partition and to move samples from one group to another if such a move will improve the value of the criterion function.
- It guarantees local but not global optimization.

4 ITERATIVE OPTIMIZATION

- Iterative Improvement to minimize the sum of squared error criterion J_e.
- Effective error per cluster J_i.

$$J_e = \sum_{i=1}^{c} J_i; \qquad J_i = \sum_{\mathbf{x} \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2$$

- A sample is moved from cluster i to cluster j.

$$\hat{\mathbf{x}} \in D_i \Rightarrow \hat{\mathbf{x}} \in D_j$$

$$\mathbf{m}_j^* = \mathbf{m}_j + \frac{\hat{\mathbf{x}} - \mathbf{m}_j}{n_j + 1}; \qquad \mathbf{m}_i^* = \mathbf{m}_i - \frac{\hat{\mathbf{x}} - \mathbf{m}_i}{n_i - 1}$$

$$n_j = n_j + 1; \qquad n_i = n_i - 1$$

4 ITERATIVE OPTIMIZATION

- Increasing / Decreasing “Effective error per cluster” (DEMOSTRAR COMO EJERCICIO)

$$J_j^* = \sum_{\mathbf{x}\in D_j} \left\|\mathbf{x}-\mathbf{m}_j *\right\|^2 + \left\|\hat{\mathbf{x}}-\mathbf{m}_j *\right\|^2 =$$

$$\left(\sum_{\mathbf{x}\in D_j} \left\|\mathbf{x}-\mathbf{m}_j - \frac{\hat{\mathbf{x}}-\mathbf{m}_j}{n_j+1}\right\|^2 \right) + \left\|\frac{n_j}{n_j+1}\left(\hat{\mathbf{x}}-\mathbf{m}_j\right)\right\|^2 =$$

$$J_j + \frac{n_j}{n_j+1}\left\|\hat{\mathbf{x}}-\mathbf{m}_j\right\|^2$$

4 ITERATIVE OPTIMIZATION

- Increasing / Decreasing “Effective error per cluster” (DEMOSTRAR COMO EJERCICIO)

$$J_i^* = \sum_{\mathbf{x}\in D_i} \left\|\mathbf{x}-\mathbf{m}_i\,*\right\|^2 - \left\|\hat{\mathbf{x}}-\mathbf{m}_i\,*\right\|^2 =$$

$$\left(\sum_{\mathbf{x}\in D_i} \left\|\mathbf{x}-\mathbf{m}_i+\frac{\hat{\mathbf{x}}-\mathbf{m}_i}{n_i-1}\right\|^2\right) - \left\|\frac{n_i}{n_i-1}\left(\hat{\mathbf{x}}-\mathbf{m}_i\right)\right\|^2 =$$

$$J_i - \frac{n_i}{n_i-1}\left\|\hat{\mathbf{x}}-\mathbf{m}_i\right\|^2$$

4 ITERATIVE OPTIMIZATION

- The sample moved from cluster i to cluster j is advantageous if

$$\frac{n_i}{n_i - 1}\left\|\hat{\mathbf{x}} - \mathbf{m}_i\right\|^2 > \frac{n_j}{n_j + 1}\left\|\hat{\mathbf{x}} - \mathbf{m}_j\right\|^2$$

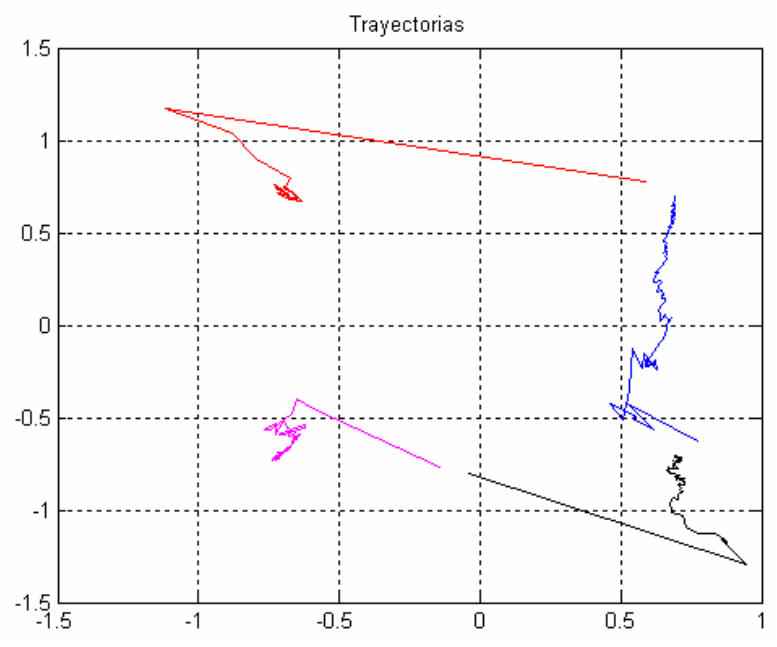
4 ITERATIVE OPTIMIZATION

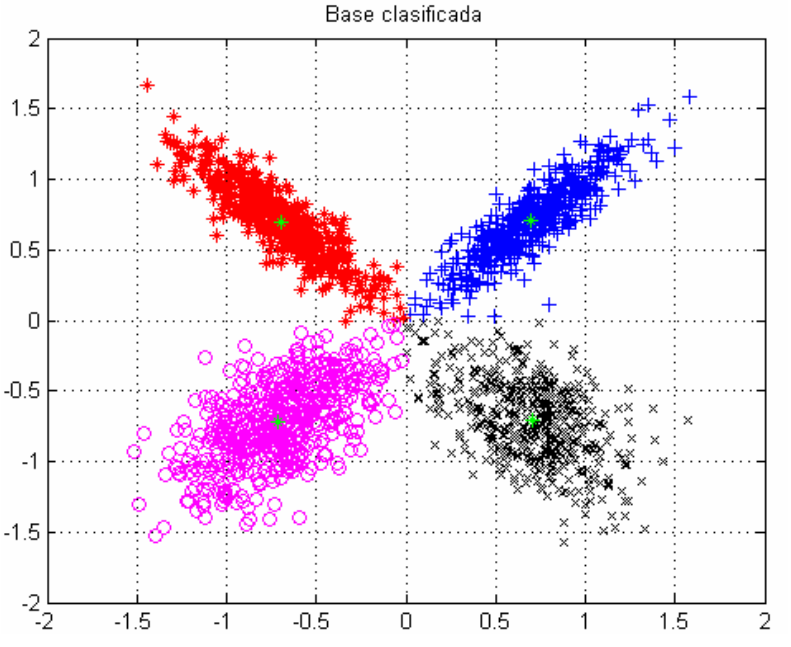
- BASIC ITERATIVE MINIMUM SQUARED ERROR CLUSTERING

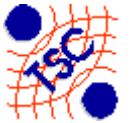
Algorithm 3 (Basic iterative minimum-squared-error clustering)

1 **begin initialize** $n, c, \mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_c$
2 **do** randomly select a sample $\hat{\mathbf{x}}$;
3 $i \leftarrow \arg\min_{i'} \|\mathbf{m}_{i'} - \hat{\mathbf{x}}\|$ (classify $\hat{\mathbf{x}}$)
4 **if** $n_i \neq 1$ **then** compute
5 $$\rho_j = \begin{cases} \frac{n_j}{n_j+1}\|\hat{\mathbf{x}} - \mathbf{m}_j\|^2 & j \neq i \\ \frac{n_j}{n_j-1}\|\hat{\mathbf{x}} - \mathbf{m}_i\|^2 & j = i \end{cases}$$
6 **if** $\rho_k \leq \rho_j$ for all j **then** transfer $\hat{\mathbf{x}}$ to $\mathcal{D}_k$
7 recompute $J_e, \mathbf{m}_i, \mathbf{m}_k$
8 **until** no change in J_e in n attempts
9 **return** $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_c$
10 **end**

4 ITERATIVE OPTIMIZATION







7 CONCLUSIONS

- When underlying distribution comes from a mixture of component densities described by a set of unknown parameters, these parameters can be estimated by Bayesian or ML (EM_algorithm) methods.
- Clustering is a more general approach.

7 CONCLUSIONS: OTHER TOPICS

- Hierarchical methods to reveal clusters and sub-clusters: Taxonomy.
- Estimation of the number of clusters
- Self-Organizing feature Maps: SOFM They preserve neighborhoods to reduce dimensionality (Kohonen Maps).

Laboratory Classes

Práctica 0: Observación de base de datos Brain, Gauss.

Práctica 1: Aplicación de métodos MAP (ldc,qdc) sobre GAUSS.

Práctica 2: Aplicación de métodos MAP (ldc,qdc) sobre PHONEME, SPAM.

Práctica 3: Aplicación de PCA y MDA sobre GAUSS.

Práctica 4: ICA como separación ciega de fuentes de audio

Práctica 5: k-Nearest Neigbour ZIP.

(Práctica 6: Discriminante Lineal (LMS-MMSE y Perceptron) sobre GAUSS y ZIP).

Práctica 7: (NN,Decisión Trees and K-means)

MULTILAYER NEURAL NETWORKS, TREE CLASSIFIERS and UNSUPERVISED Methods applied to PET and Magnetic Resonance BRAIN Images.