MAIA

Image Segmentation

Xavier Lladó, Robert Martí

Image segmentation

1. Definitions, representation and evaluation results
2. Region based methods
3. Clustering based methods
4. Other methods
5. Actual methods

Today (1,2,3): we will see very easy and simple methods!

Next class (4,5): more complex!

Definitions

Image segmentation [Haralick&Shapiro]

An image segmentation is the partition of an image into a set of nonoverlapping regions whose union is the entire image. The purpose of segmentation is to decompose the image into parts that are meaningful with respect to a particular application....

It is very difficult to tell a computer program what constitutes a "meaningful" segmentation. Instead, general segmentation procedures tend to obey the following rules.

Definitions

Rules:

1. Regions of an image segmentation should be uniform and homogeneus with respect to some characteristics, such as grey level or texture.
2. Region interiors should be simple and without many holes
3. Adjacent regions of a segmentation should have different values with respect to the characteristic on which they are uniform.
4. Boundaries of each segment should be simple, not ragged, and must be spatially accurate.

Definitions

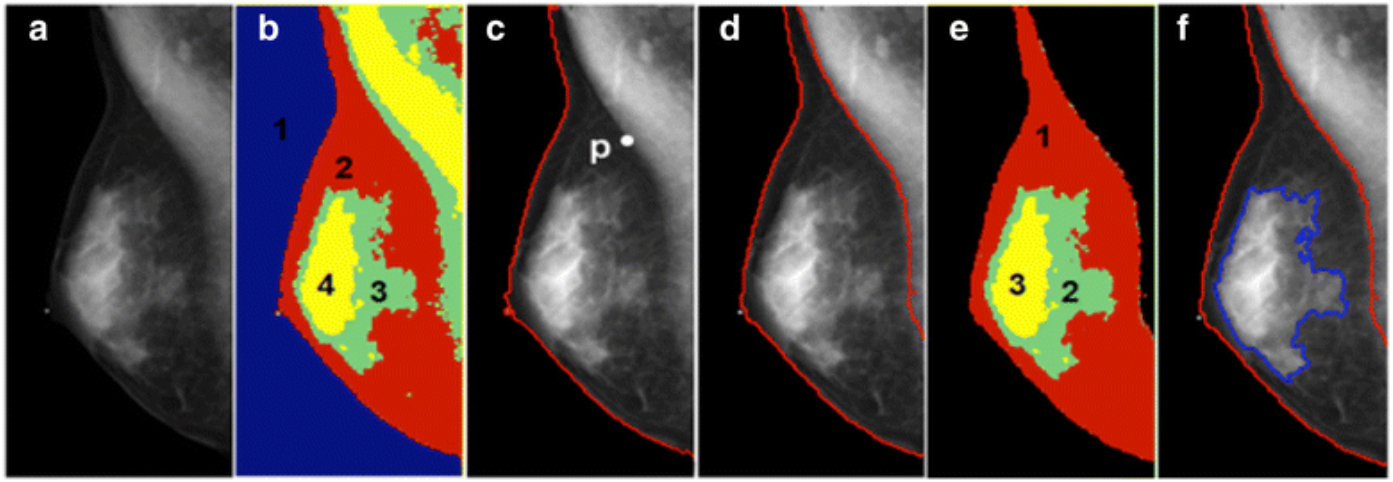
- The literature on Segmentation is large and generally inconclusive
- In some applications Segmentation is the crucial step

[*Hager*]

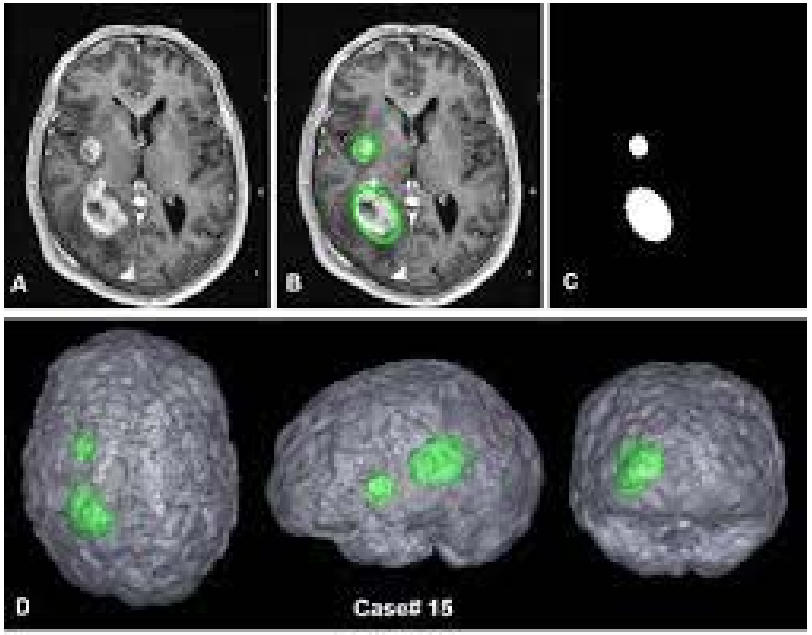
- "Old problem" in Computer Vision, but it is a very open field …
- "Segmentation is one of the oldest, and still unsolved, areas of Computer Vision"
- "In complex cases the segmentation problem can be very difficult and might require application of a great deal of domain knowledge"

Definitions

Brain tissue segmentation



Tumor / Lesoin segmentation

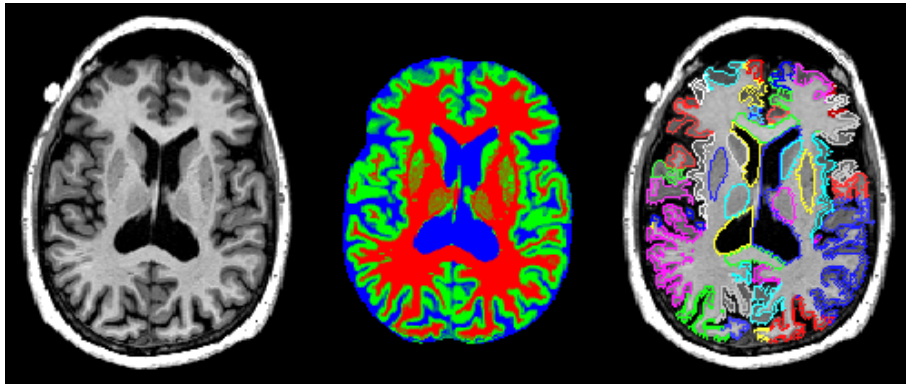


Universitat de Girona

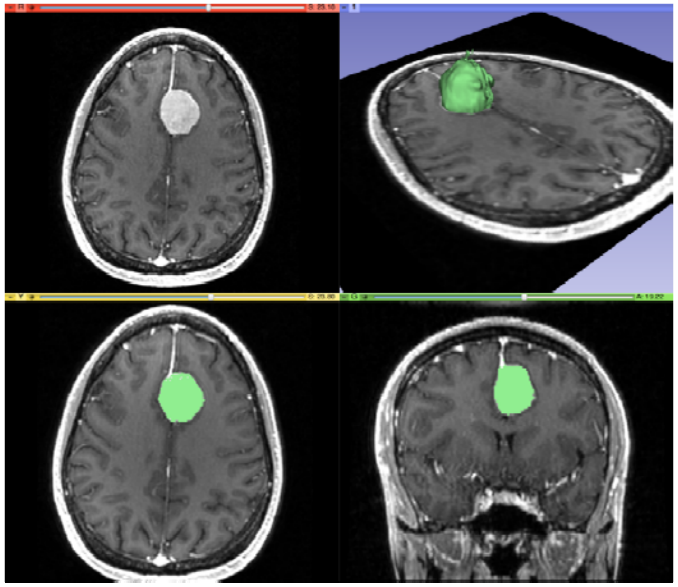
Definitions

An initial classification:

- Unsupervised segmentation



- Purposive segmentation



Evaluation results

How to evaluate? The common practice is the comparison with manual segmentation (called ground truth):

- Using real images → hard work, non objective task
- Using synthetic images → other problems! That are not real

Two typical measures:

- Contour Criteria: To compare obtained region limits vs. original image contours
- Region Criteria: Area intersection (obtained region and original image region). Pixels belonging to a object and classified as a background (and viceversa)

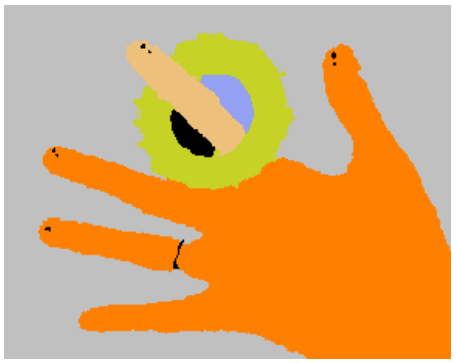
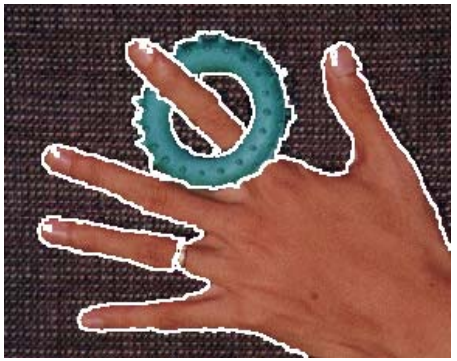
Evaluation results

Oversegmentation / Undersegmentation



How to show segmentation results:

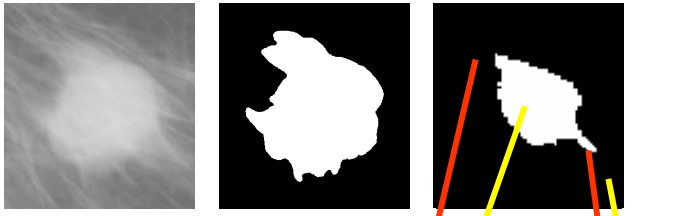


Universitat de Girona

Evaluation results

ROC (Receiver Operated Curve): comparing the ground truth with the algorithm result



Algorithm \ doctor	Mass	No Mass
Mass	TP	FP
No Mass	FN	TN

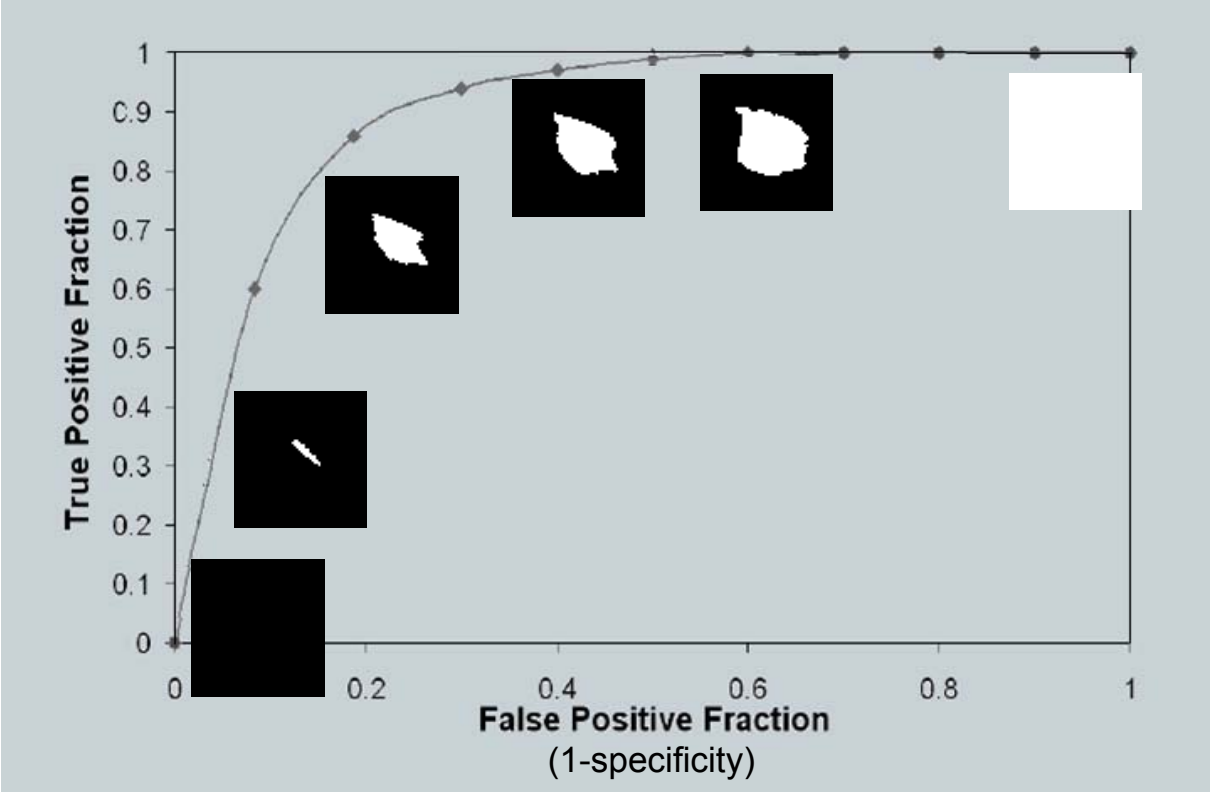
Sensitivity Specificity

$$\text{sensitivity} = \frac{\text{number of True Positives}}{\text{number of True Positives} + \text{number of False Negatives}}.$$

$$\text{specificity} = \frac{\text{number of True Negatives}}{\text{number of True Negatives} + \text{number of False Positives}}$$

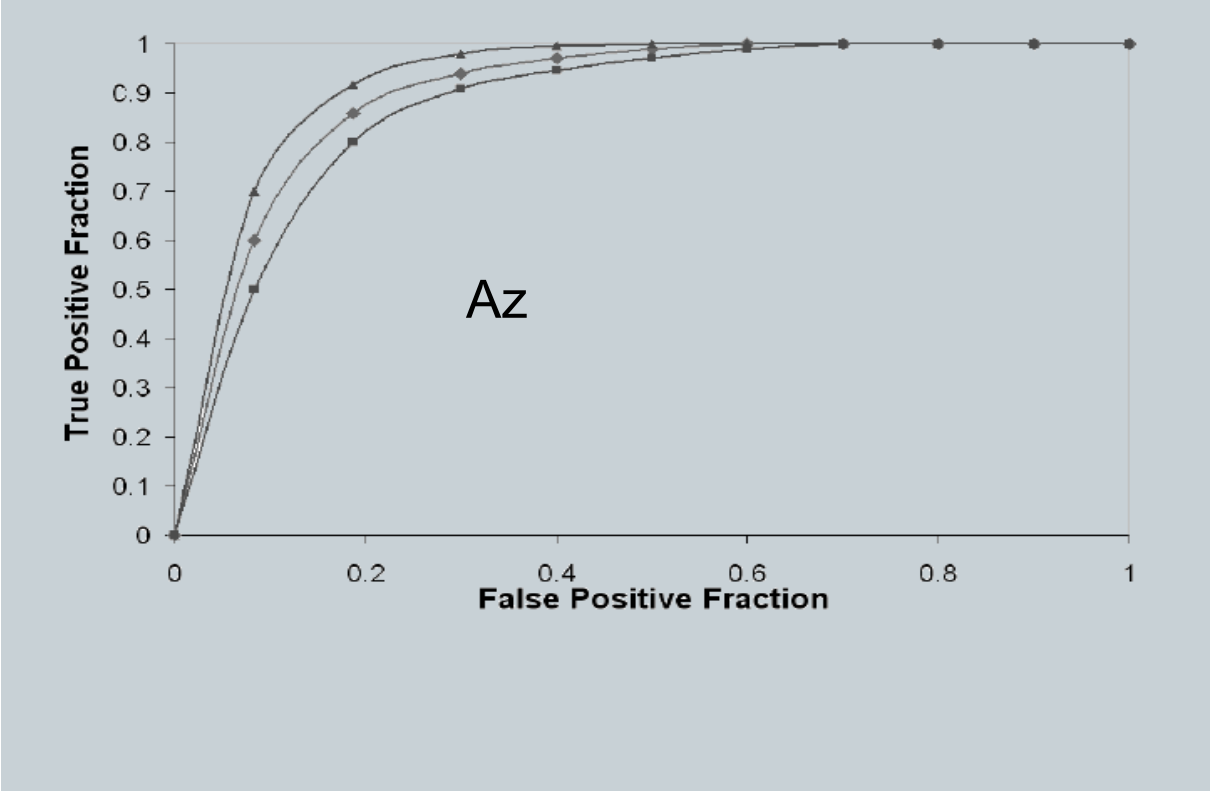
Evaluation results

ROC (Receiver Operated Curve): comparing the hand segmented with the algorithm result



Evaluation results

ROC (Receiver Operated Curve): comparing the hand segmented with the algorithm result



Evaluation results

Common evaluation metrics:

Accuracy: (TP + TN) / (TP + TN + FP + FN)

$$tp\ rate = \frac{TP}{P} = \frac{TP}{TP + FN} = recall$$

$$fp\ rate = \frac{FP}{N} = \frac{FP}{FP + TN}$$

$$\text{sensitivity} = \text{recall}$$

$$\text{specificity} = \frac{\text{True negatives}}{\text{False positives} + \text{True negatives}} = 1 - fp\ rate$$

$$\text{positive predictive value} = \text{precision}$$

$$precision = \frac{TP}{Y} = \frac{TP}{TP + FP}$$

Hypothesized class \ True class	p	n
Y	True Positives	False Positives
N	False Negatives	True Negatives

Evaluation results

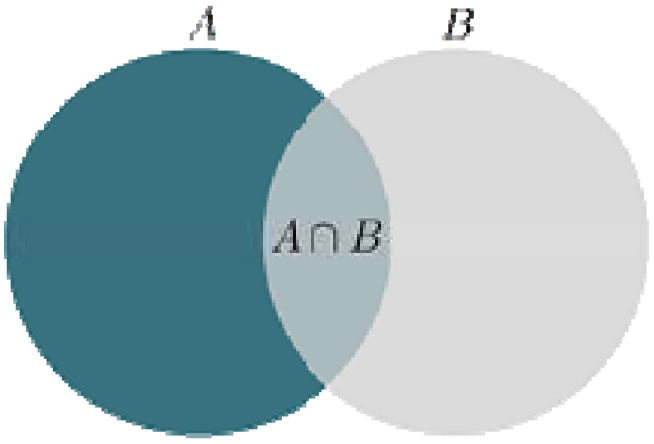
Area overlap and DICE measures

Jaccard Index is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Dice Index, also know as Dice Similarity Coefficient (DSC) is a similarity measure over sets:

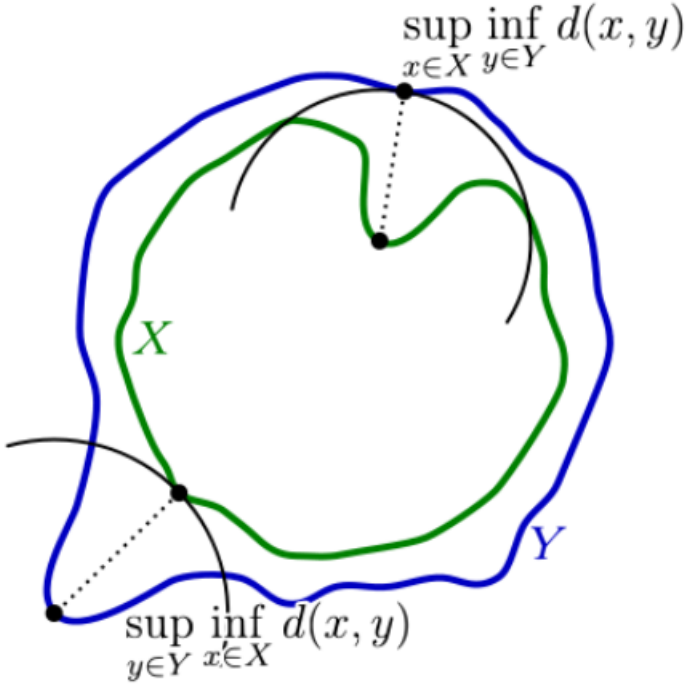
$$DSC = \frac{2|A \cap B|}{|A| + |B|}$$



Evaluation results

Distance measures:

- **Hausdorff Distance** between two sets gives an idea about their dissimilarity
- Two sets are close in the Hausdorff distance if every point of either set is close to some point of the other set
- The Hausdorff distance is the greatest of all the distances from a point in one set to the closest point in the other set.



$$d_{\mathrm{H}}(X, Y) = \max\{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \}$$

Image segmentation

2. Region based methods

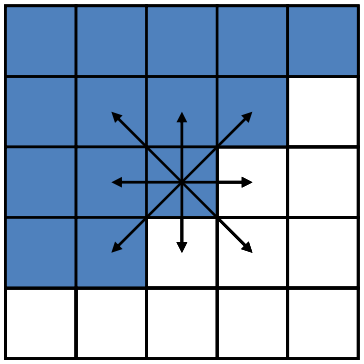
- Region Growing
- Split & Merge

3. Clustering based methods

- Thresholding methods
- K-means
- EM algorithms

Region based methods

Region Growing



Considerations:

- Implementation: recursive, sequential, concurrent,...
- Seed distribution: how many and placement
- Aggregation criteria: intensity, color, texture. A typical one is:

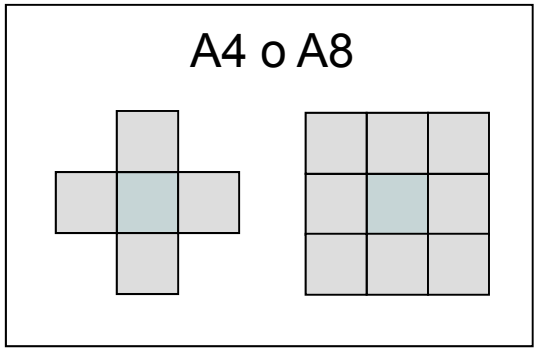
$$If\ |f(x,y) - \mu_{Ri}| <= \Delta \qquad then\ add\ f(x,y)\ to\ Ri,\ update\ \mu_{Ri}$$

Region based methods

PROGRAM Region_Growing

Mark all the pixels as not considered

FOR all the pixels (x,y) of the image DO
IF pixel(x,y) no considered THEN
Begin statistics new region Ri
Mark pixel(x,y) as a considered
Explore(x,y)
Increase the number of seeds
ENDIF



ACTION Explore(x,y)
WHILE ((x',y') is an adjacent pixel respect to (x,y) not considered) and ((x',y') belongs to actual region) DO
Mark pixel(x',y') as a considered
Recompute statistics of region Ri
Explore(x',y')

Aggregation Criteria: intensity

If $|f(x,y) - \mu Ri| <= \Delta$

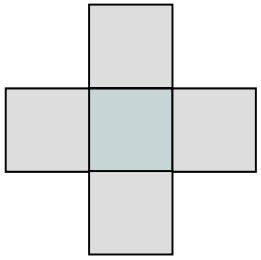
Region based methods

Region based methods

Implementation example: Visiting the neighbouring pixels using a queu

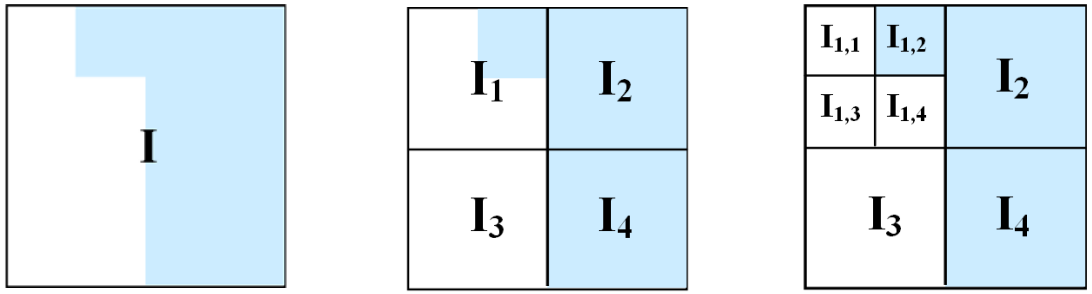
			14			
		16	6	15		
	18	8	2	7	17	
25	13	5	1	3	9	19
	24	12	4	10	20	
		23	11	21		
			22			

connectivity A4

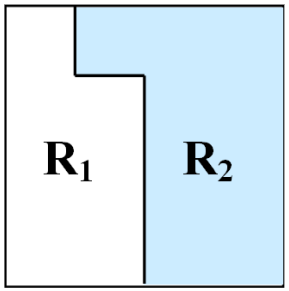


Region based methods

Split & Merge (Pavlidis method)



Split



Merging

Considerations :

- Criteria and features for splitting and merging: intensity,color,texture
- Quad-tree representation
- Usually has high cost on merging, and pixeled aspect of result

Region based methods

Split & Merge results

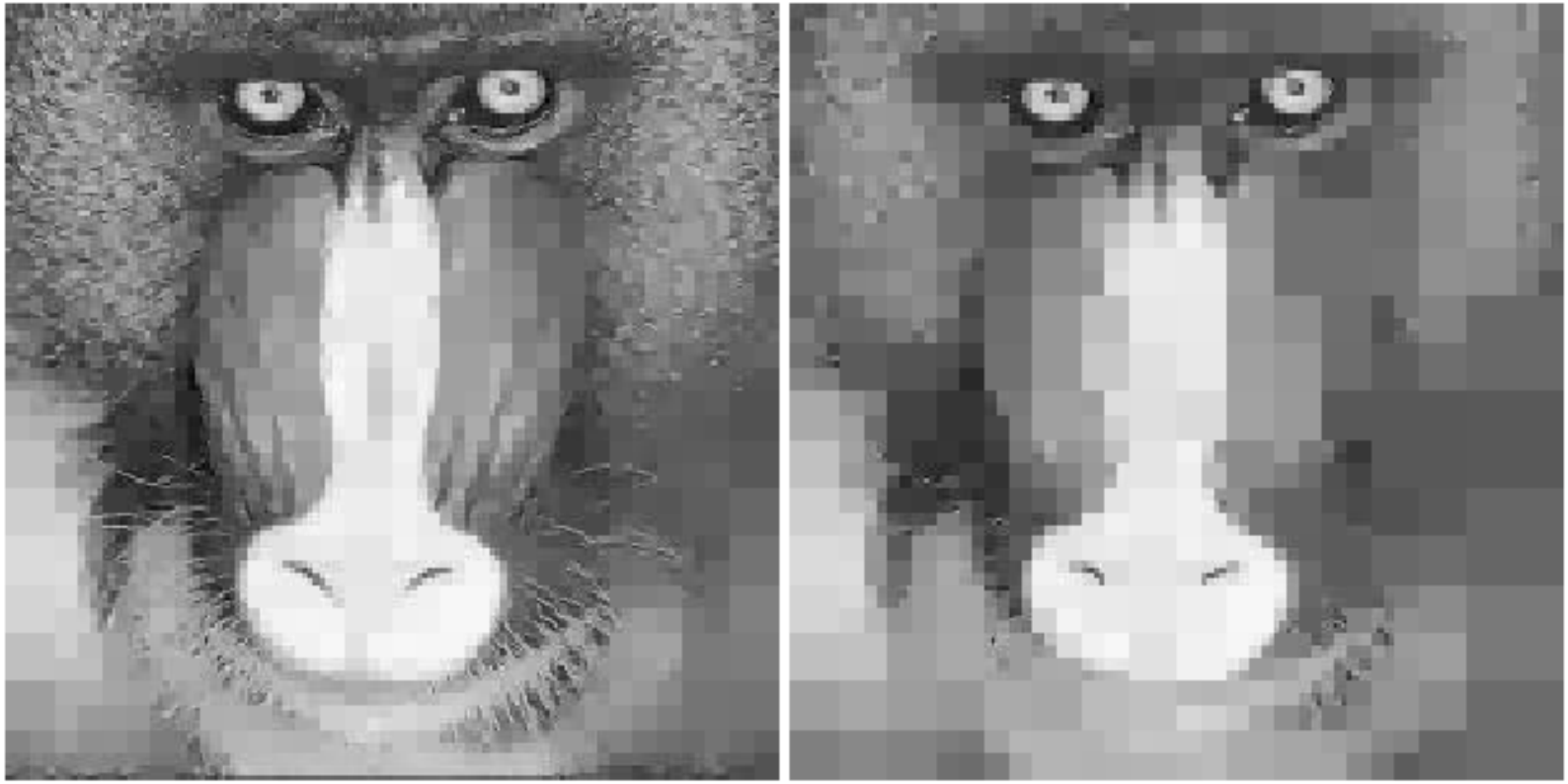


Image segmentation

Region based methods

- Region Growing
- Split & Merge

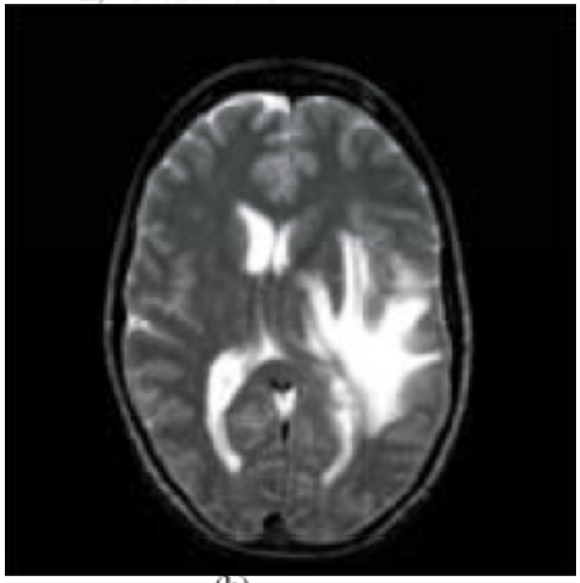
Clustering based methods

- Thresholding methods
- K-means
- EM algorithms

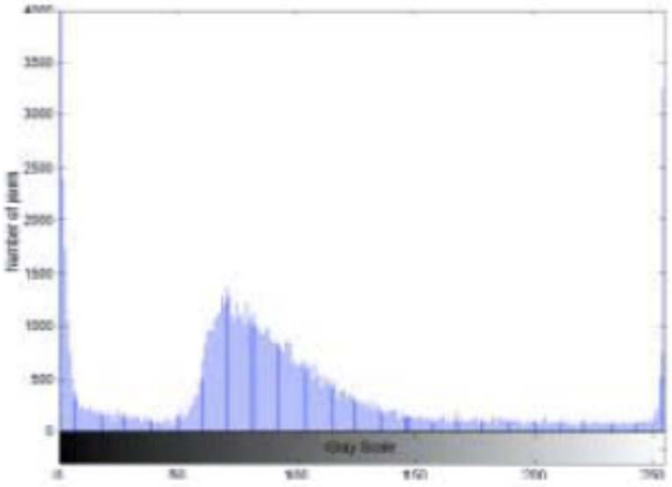
Clustering based methods

Image segmentation [Haralick&Shapiro]

The difference between image segmentation and clustering is that in clustering, the grouping is done in measurement space: In image segmentation, the grouping is done in spatial domain of the image.



(b)



T=166

T=225

Clustering based methods

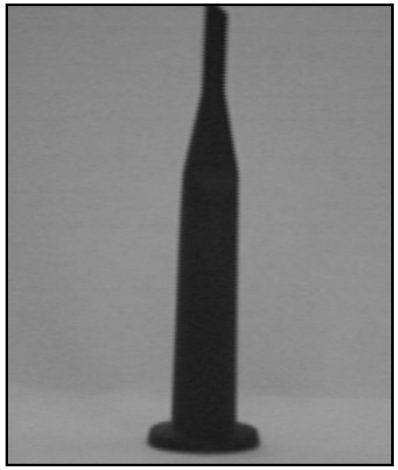
Cluster: grouping of pixels considering 1 o more features

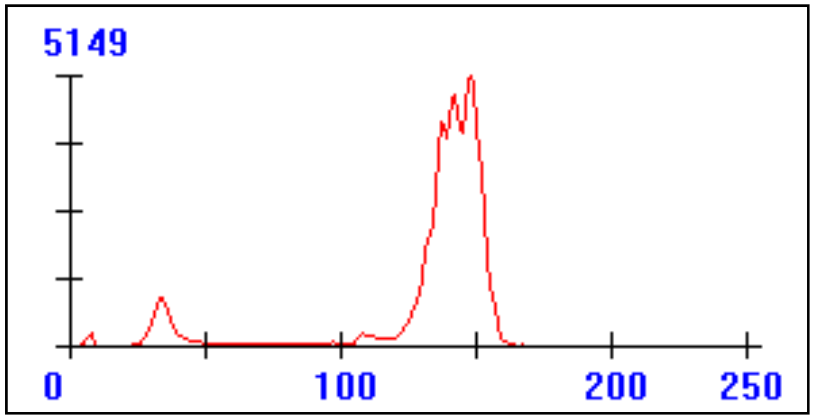
- Feature selection: Feature Space (R,G and B, in the previous Figure)
- Similar features… in a cluster
- Features significantly different between different clusters
- The spatial distribution of pixels in the image is not considered (feature space is considered)
- Shape of clusters: compact but, spherical, ellipsoidal, elongated, …
- Ownership of a pixel to a cluster depends on a proximity measure (distance). Grouping Methodology !!!
- Results may be subjective…

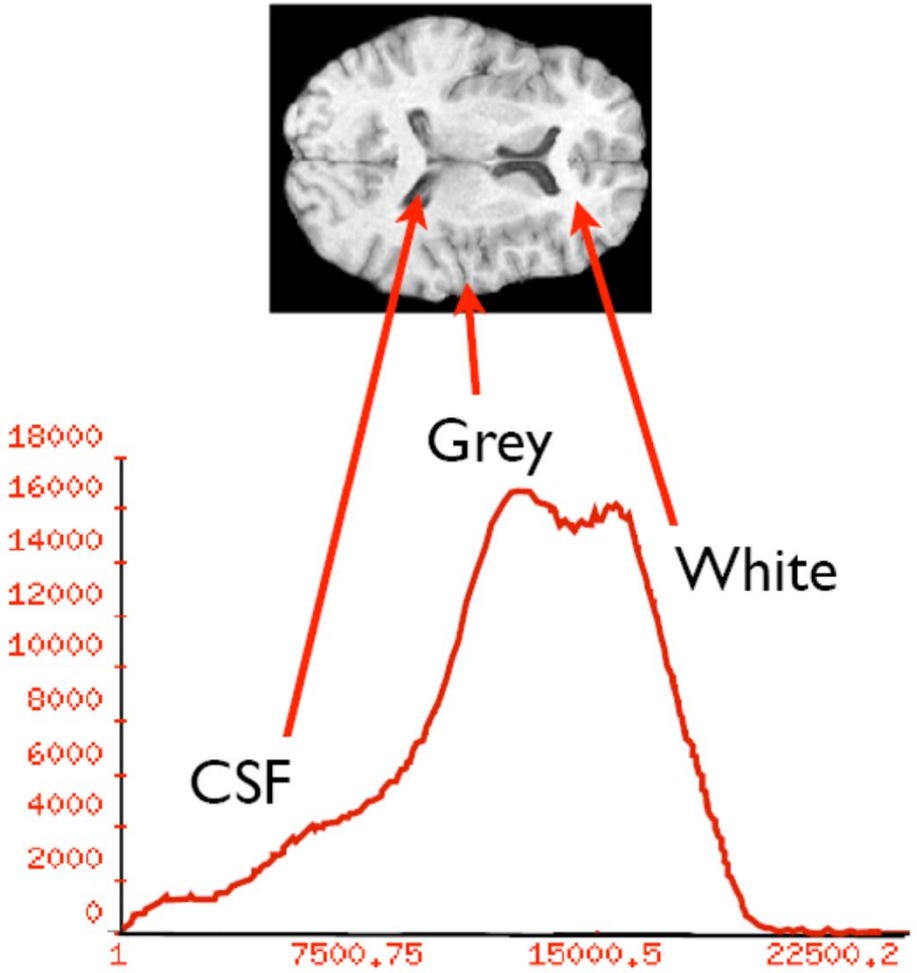
Key point: number of clusters of an image ??

Clustering based methods

Clustering en 1D = Histogram analysis → thresholding

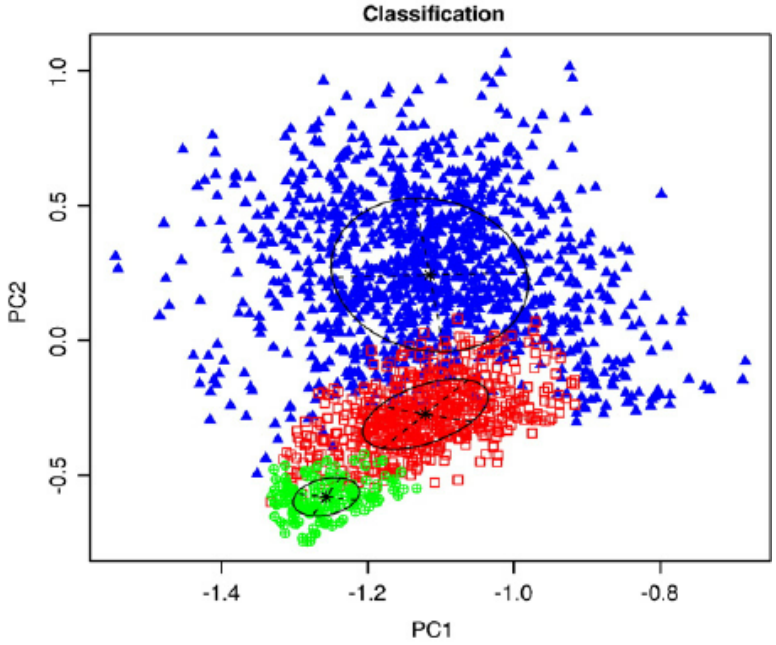


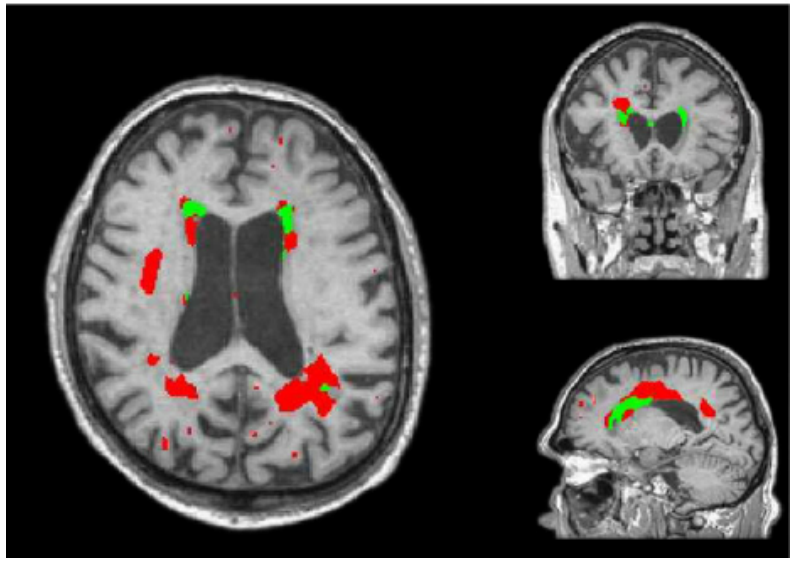




Universitat de Girona

Clustering based methods





MAIA

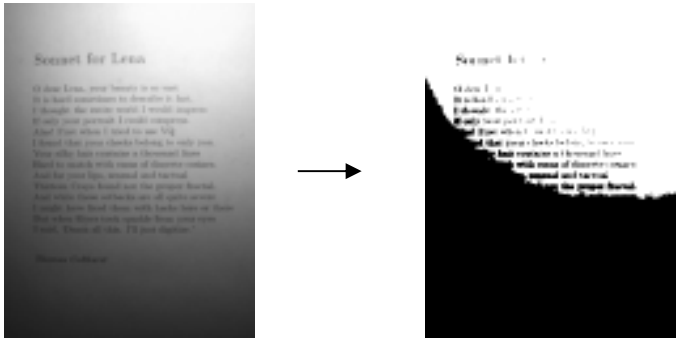
Image thresholding

Universitat de Girona

Thresholding: fixed vs adaptative

- In fixed or global thresholding, the threshold value is held constant throughout the image
- A variation could uses two or more thresholds

Global Thresholding



Local/Adaptive Thresholding

7x7; T = mean 7x7; T = mean-7 75x75; T = mean-10

Optimal thresholding

Methods able to find the optimal threshold:

- Isodata, Peak and valley, Otsu, p-tile,...
- What happens when appearing more than two modes: Ohlander,...

Let's see a couple of examples

Optimal thresholding

ISODATA (Iterative Self-Organising Data Analysis Technique Algorithm)

Initially the histogram is segmented in 2 parts. Then, we compute the mean value associated to each part of the histogram m1, m2. With these values we compute a new threshold doing T= (m1+m2) / 2. We repeat the process until convergence

ISODATA step by step:

1)Chose the initial threshold Ti=T0 (median histogram, maximum gray level…)
2)Divide the image in two groups R1 and R2 using Ti
3)Compute the mean of the gray level for both parts m1 and m2
4)Select the new threshold Ti+1 = (m1 + m2)/2
5)Repeat steps 2-4 until Ti = Ti-1

Optimal thresholding

Example: H(z)=

2	5	7	16	34	22	17	4	5	5	3	23	45	56	12	3

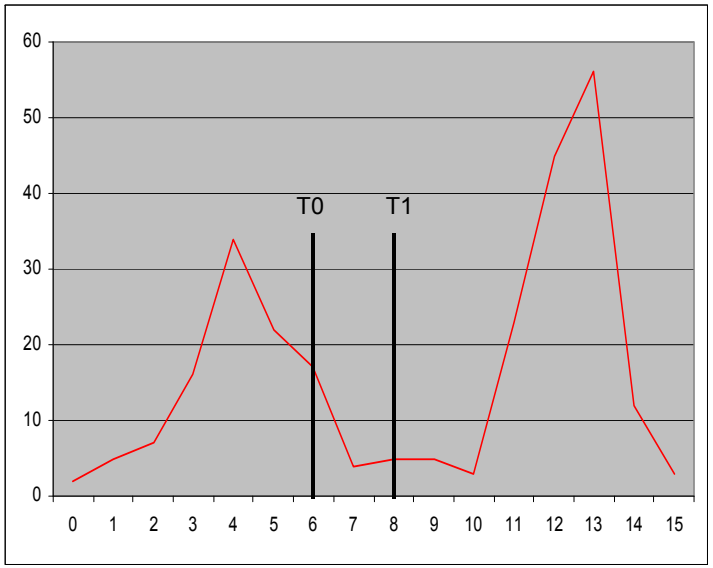
z has values from 0 a 15 (4 bits)

1) T_0=position(maximum)/2=13/2=6
   (we could also do T_0=mean of the histrogram $T_0=\Sigma\ z*H(z)/\ \Sigma H(z) \rightarrow T_0=8.85$)

2) $m_1=\Sigma\ z*H(z)/\ \Sigma H(z)$ for z = 0 to 6= 4.02
   $m_2= \Sigma\ z*H(z)/\ \Sigma H(z)$ for z = 7 to 15= 12.03
   $T_1 = (m1 + m2)/2 = 8.03$

3) $m_1=\Sigma\ z*H(z)/\ \Sigma H(z)$ for z = 0 to 8= 4.31
   $m_2= \Sigma\ z*H(z)/\ \Sigma H(z)$ per z = 9 to 15= 12.31
   $T_2 = (m_1 + m_2)/2 = 8.\ 31$

FINAL floor(T2)=floor(T3)



Universitat de Girona

Optimal thresholding

Peak and valley method

1. Assign to Z1 the gray level z which is the maximum of H(z)

2. Assign to Z2 the gray level that maximises g(z)=|Z1-z| * H(z)

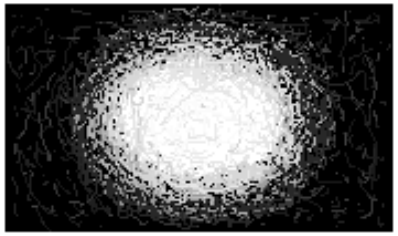
This is called to maximise the peakiness, the difference between peaks and valley

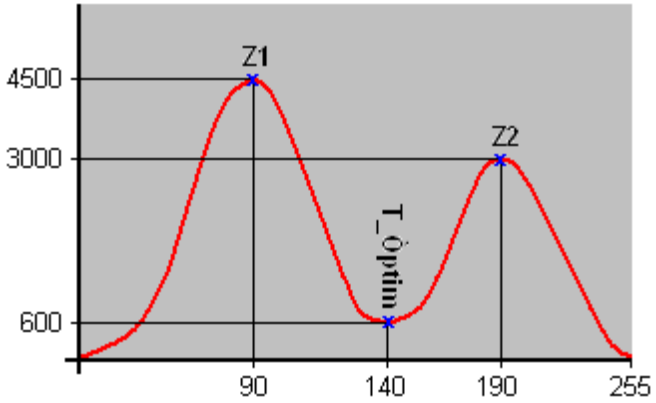
3. Assign the threshold T to the gray level that is the minimum of H(z) going:

from z=Z1+1 to Z2-1, if Z2>Z1
from z=Z2+1 to Z1-1, if Z2<Z1

Optimal thresholding

✓ It works when the histogram has 2 well defined pics

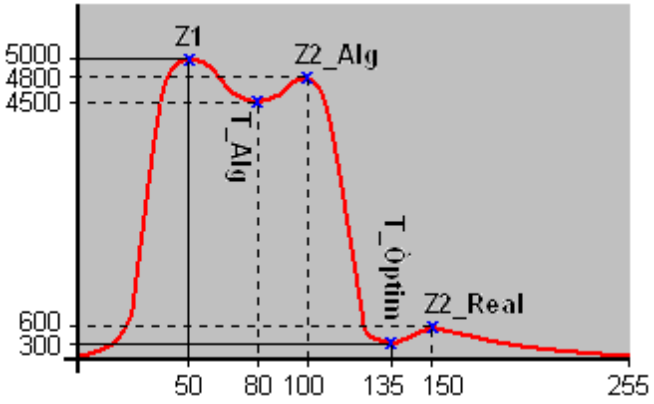




Computing Z2:

➡ g(z)=|Z1-z|*H(z)
Z2→ g(z)max =
= |90-190|*3.000 = 300.000

✗ If **1 peak is very small** we can have problems computing Z2



Computing Z2:

➡ g(z)=|Z1-z|*H(z)
Alg→ |50-100|*4.800 = 240.000
Real→ |50-150|*600 = 60.000

Universitat de Girona

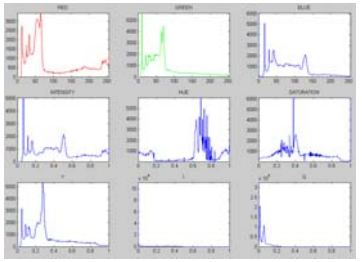
Optimal thresholding

Ohlander method

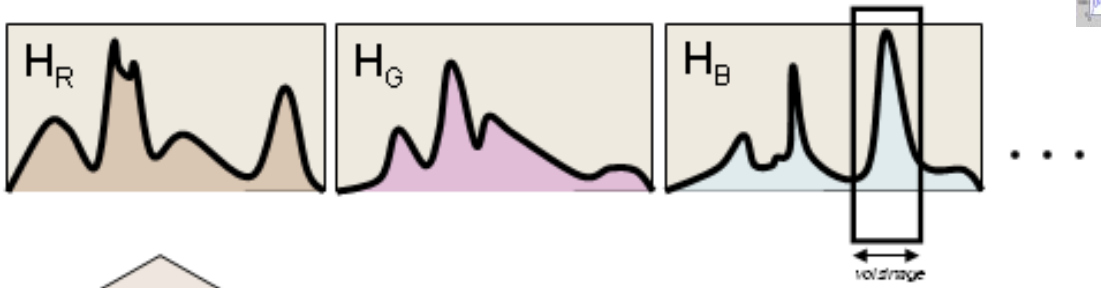
- Region based algorithm using thresholding technique
- Based on exploiting the chromatic information by constructing colour and hue histograms
- Regions are split recursively based upon histogram analysis
- Picture is thresholded at its most clearly separated peak

Optimal thresholding

Ohlander method

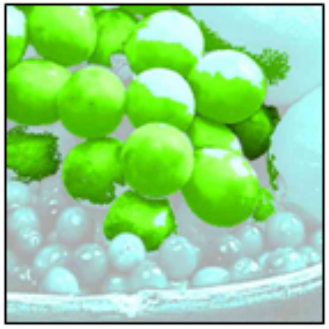


Initial image



Re-injection of sufficient-sized regions

Retro projection of the histogram window



Suppression of the extracted region

Image segmentation

2. Region based methods

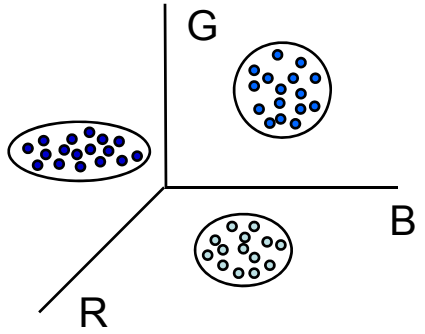
- Region Growing
- Split & Merge

3. Clustering based methods

- Thresholding methods
- K-means
- EM algorithms

Clustering based methods

K-means algorithm



-Successive generation of clusters, minimizing a cost function J, obtaining the “best” clusters

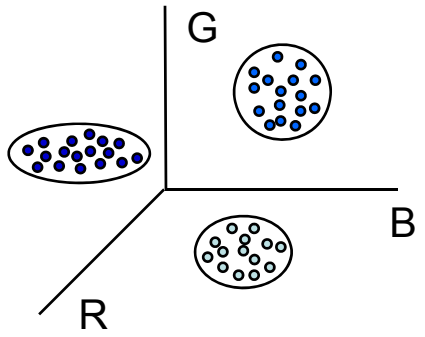
- “a priori” knowledge of: # of clusters, ownership of every pixel to only one cluster Cj (minimum distance to the centroid Φj)

$$\phi j = \frac{1}{N} \sum_{Xi \in Cj} Xi \qquad d(Xi, \phi j) = \left\| Xi - \phi j \right\|$$

Clustering based methods

K-means algorithm

Cost Function J:



$$J = \sum_{i=1}^{N} \sum_{j=1}^{K} u_{ij} \times d(x_i, \phi_j)$$

$$where\ \phi_j\ is\ the\ centroid\ of\ cluster\ C_j$$

- *Uij* is a coefficient related to the ownership of a pixel *i*

to a cluster *j*:

Uij = 1, if *d* is the minimum distance

Uij = 0, otherwise

K-means

4 steps, iterative algorithm:

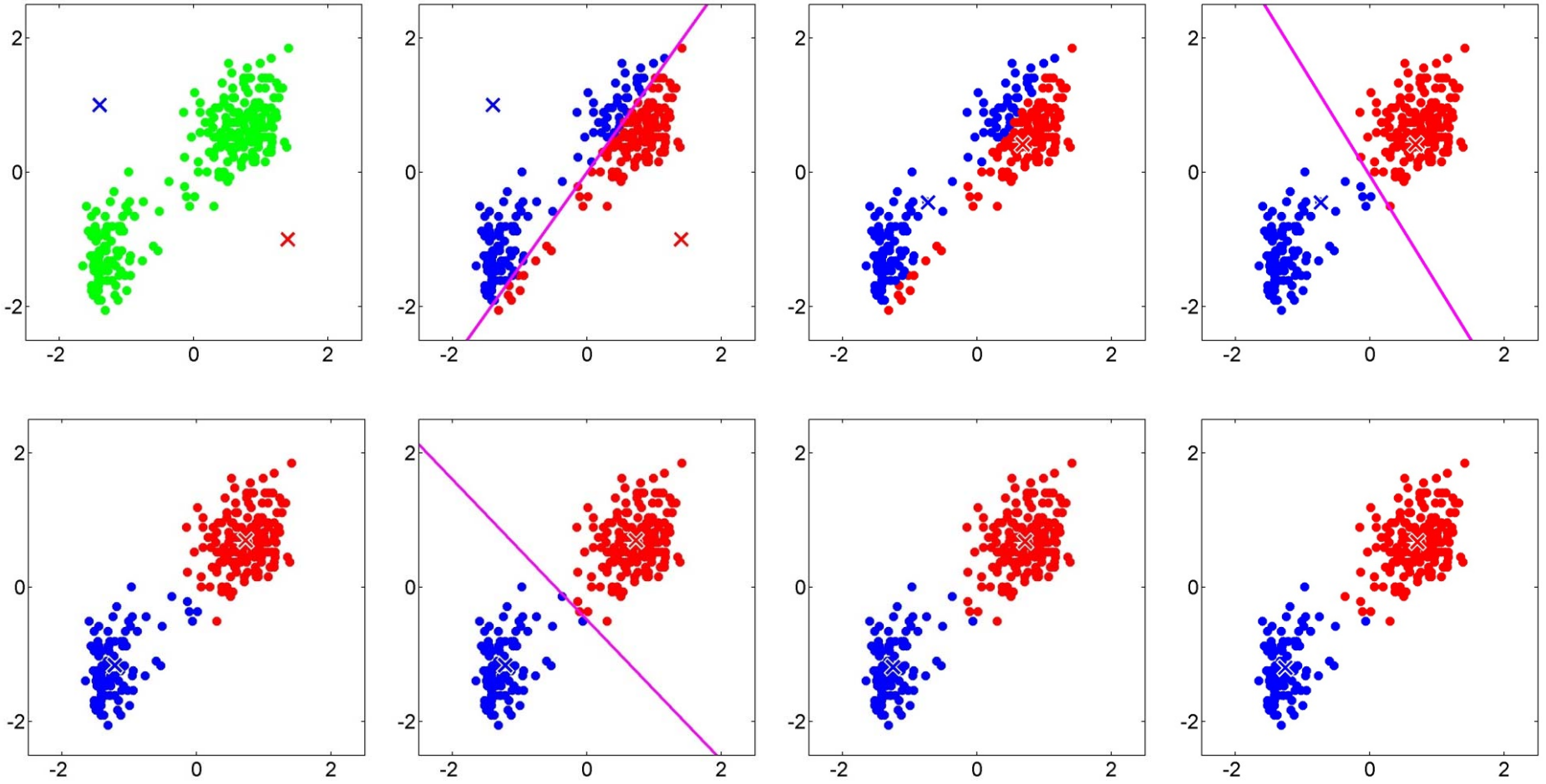
- Phase 1
  K is known
  Determination of the initial centers of each cluster ϕ_j, j = 1…K
- Phase 2
  Each pixel x_i is assigned to its nearest cluster C_j:
- Phase 3
  Compute the "new" centers of the clusters ϕ_j
- Phase 4
  If any pixel has changed the cluster (phase 2) then go to phase 2, else the algorithm finish.

Final goal, minimize:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{K} u_{ij} \times d(x_i, \phi_j)$$

$on \quad \phi_j \quad és \quad el \quad centroid \quad del \quad cluster \quad C_j$

K-means



Example taken from Christopher M. Bishop BCS Summer school, Exeter 2003

http://research.microsoft.com/~cmbishop/talks.htm

K-means

```
PROGRAM K-Means(number_clusters,inicial_centers)
        centers=Compute_inicial_centers(number_clusters,inicial_centers)
        REPEAT
        FOR all the pixels (x,y) of the image  DO
    changes = Compute_cluster_pixel(x,y,centers) ENDFOR
        centers = Compute_centers(number_clusters)
        UNTIL NO changes
ENDPROGRAM

FUNCTION Compute_cluster_pixel(x,y,centers)
  cluster_previous = cluster_actual
  cluster_actual = nearest_cluster(x,y,centers)
  IF cluster_previous != cluster_actual THEN  RETURN true
  ELSE  RETURN false
ENDFUNCTION

FUNCTION Compute_centers(number_clusters)
  FOR all number_clusters i DO
    centers(i) = Mean_pixels_cluster(i)
  ENDFOR
  RETURN centers
ENDFUNCTION
```

MAIA

EM- Expectation Maximization

Xavier Lladó, Robert Martí

Contents

- **Introduction to the classification problem**
- **Description E-M Algorithm**
- **Evaluation**

EM algorithm

- The Expectation-Maximization (EM) algorithm is an iterative technique designed for probabilistic models.

- It is often used for finding the unknown parameters of a mixture model.

INITIALIZE randomly the model parameters.

REPEAT

Expectation step:

Compute labels for all the dataset given the current cluster parameters.

Maximization step:

Use that classification to reestimate the parameters

UNTIL convergence is achieved → not significant changes on the parameters

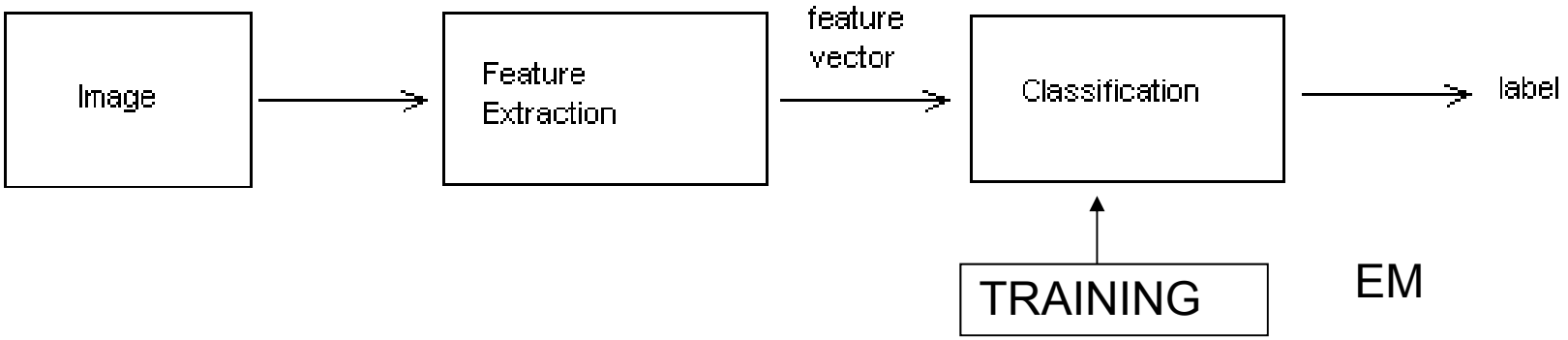
Introduction to classification

- Objectives

  To segment image by means of classification of feature vectors extracted from images

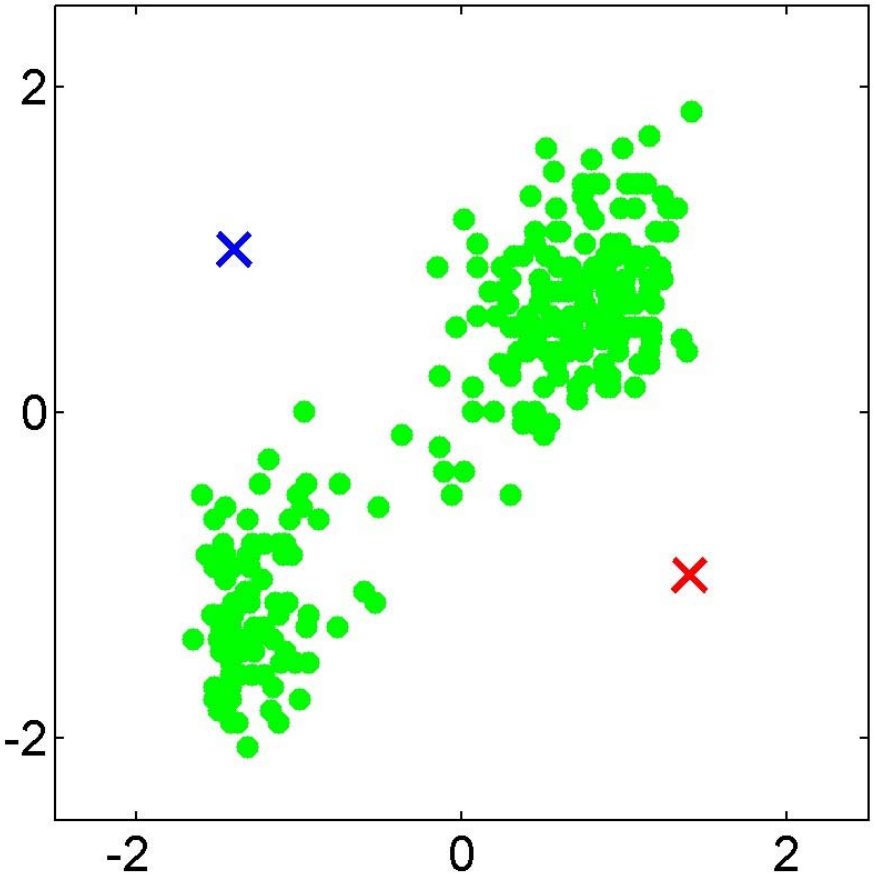
- Procedure

  The whole procedure can be described as follows



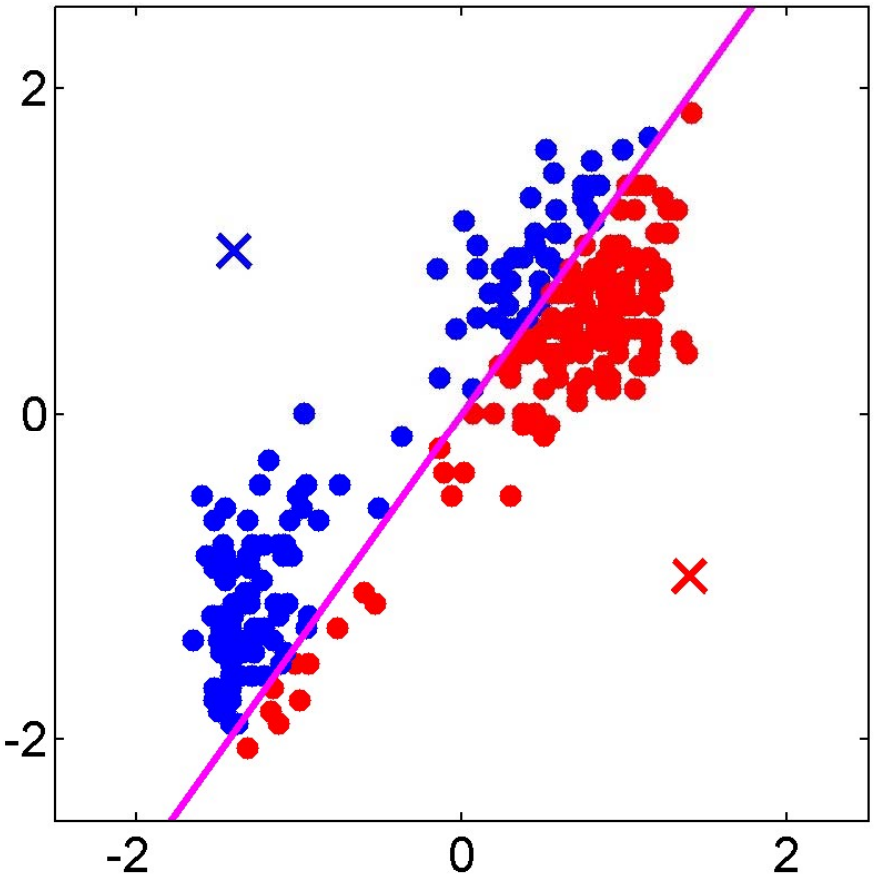
Clustering example

- K-means example. Initial centers



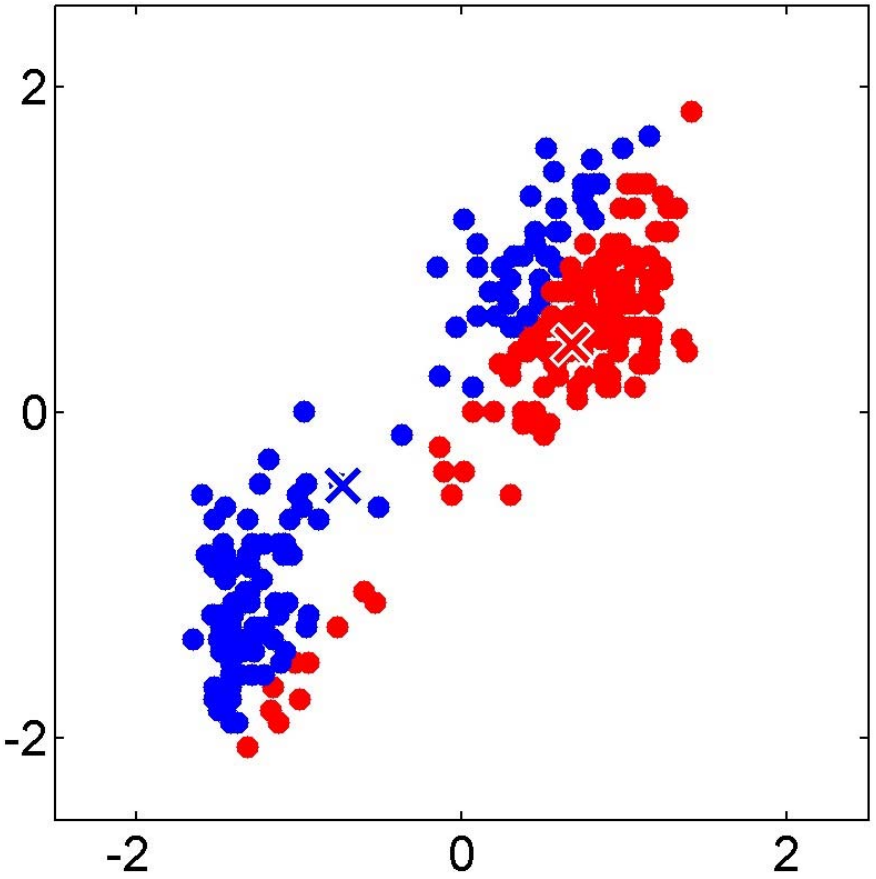
Clustering example

- First step assign points to the nearest centroid



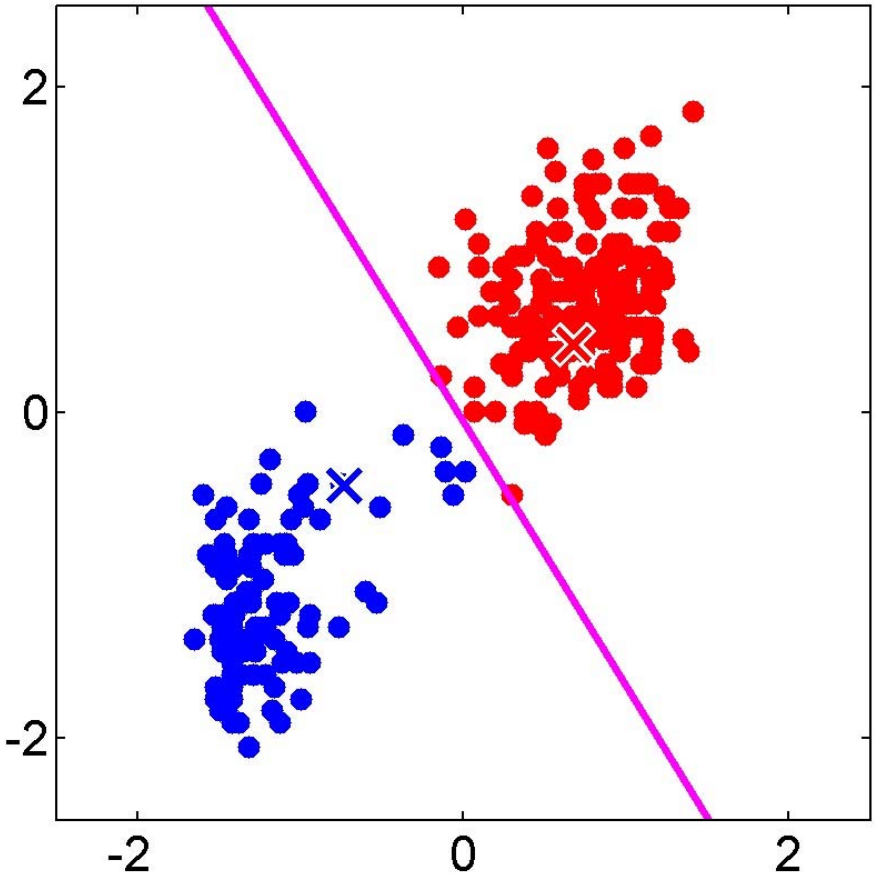
Clustering example

- Compute new position of centroids given assigned points



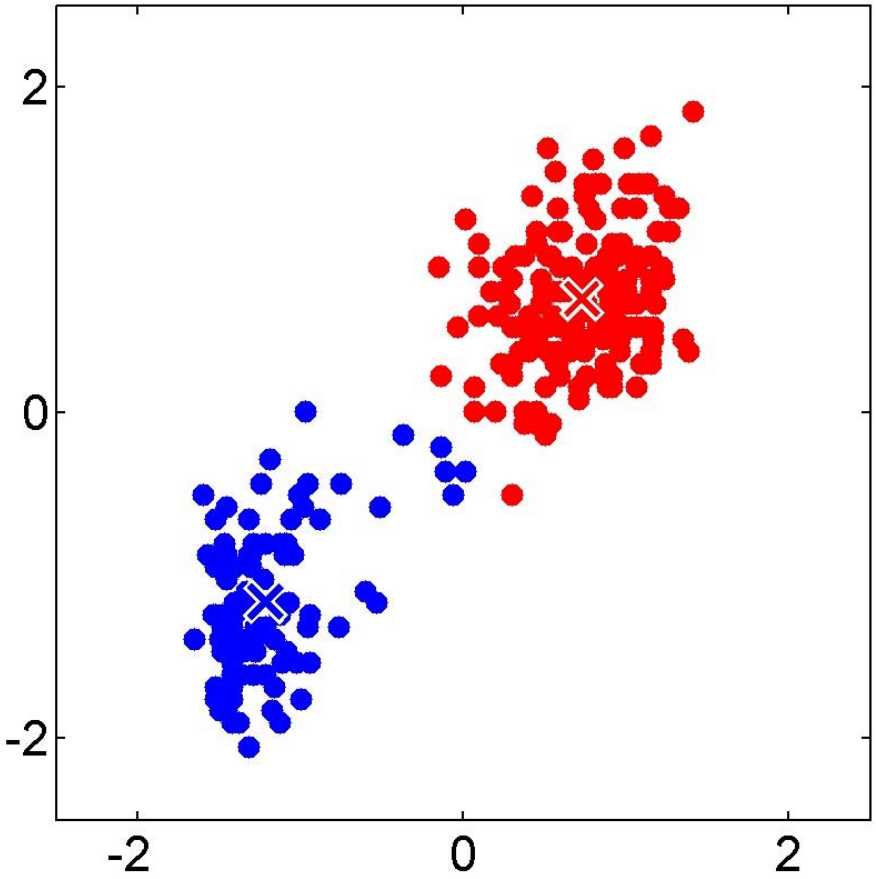
Clustering example

- Iterate re-assigning points to nearest centroids...



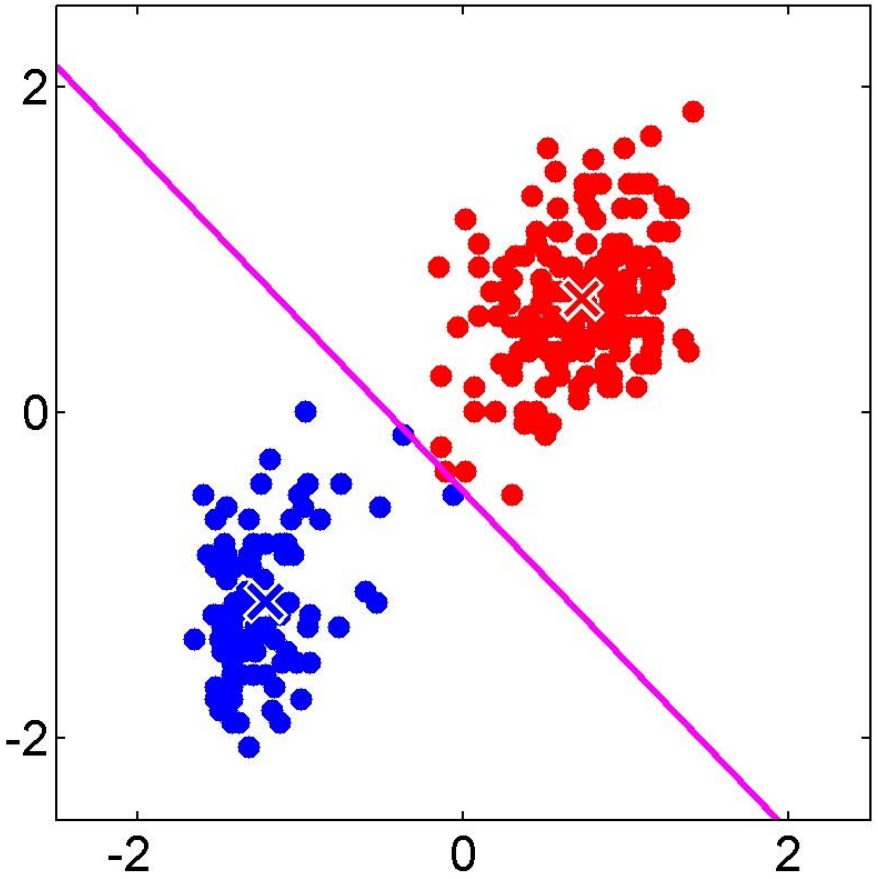
Clustering example

- ... and re-compute new centroids



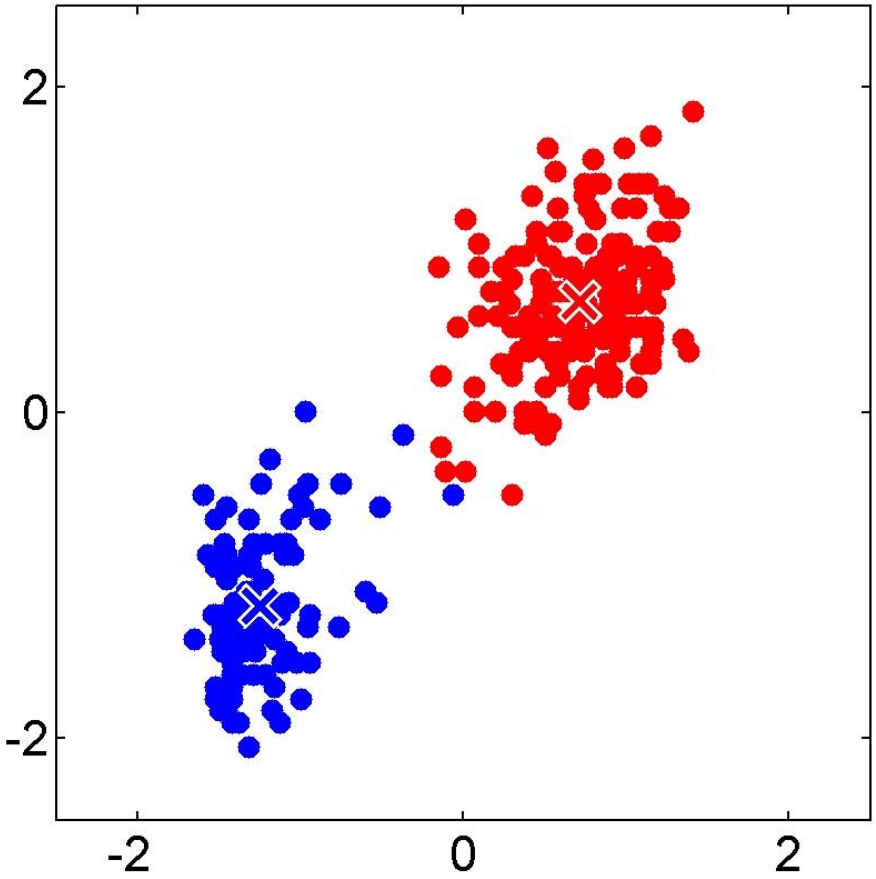
Clustering example

- Iterate until convergence



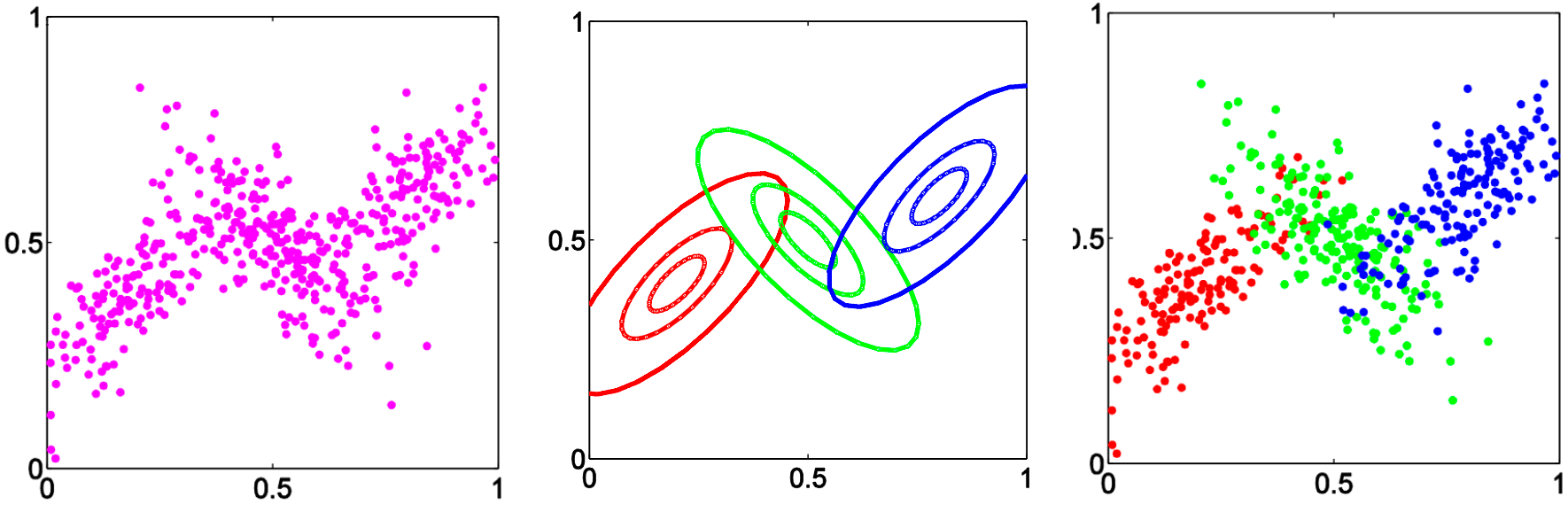
Clustering example

- Convergence is assumed when no significant changes are seen



Gaussian mixture models

- Other situations lead to Gaussian distributions



EM algorithm

- EM algorithm is iterative technique designed for probabilistic models
- It is often used for finding the unknown parameters of a mixture model
- Likelihood is the probability of a sample of belong to a class, given the parameters of that class
- EM maximizes the log-likelihood of the sample as represented by the mixture model: maximizing the likelihood we can fit a mathematical model to the data

EM algorithm

- INITIALIZE the parameters of the model, randomly

- REPEAT

  Expectation step:

  Recompute labels for all the dataset given the current cluster parameters.

  Maximization step:

  Use that classification to reestimate the parameters

- UNTIL converges to a local optimum → not significant changes on the parameters

EM: Maths

given a data set $D = \{\underline{x}_1, \ldots, \underline{x}_N\}$ where $\underline{x}_i$ is a d-dimensional vector measurement.

$p(\underline{x})$ is a finite mixture model with K components:

$$p(\underline{x}|\Theta) = \sum_{k=1}^{K} \alpha_k p_k(\underline{x}|z_k, \theta_k)$$

where:

- The $p_k(\underline{x}|z_k, \theta_k)$ are *mixture components*, $1 \leq k \leq K$. Each is a density or distribution defined over $p(\underline{x})$, with parameters θ_k.
- $z = (z_1, \ldots, z_K)$ is a vector of K binary indicator variables that are mutually exclusive and exhaustive (i.e., one and only one of the z_k's is equal to 1, and the others are 0). z is a K-ary random variable representing the identity of the mixture component that generated $\underline{x}$. It is convenient for mixture models to represent z as a vector of K indicator variables.
- The $\alpha_k = p(z_k)$ are the mixture weights, representing the probability that a randomly selected $\underline{x}$ was generated by component k, where $\sum_{k=1}^{K} \alpha_k = 1$.

The complete set of parameters for a mixture model with K components is

$$\Theta = \{\alpha_1, \ldots, \alpha_K, \theta_1, \ldots, \theta_K\}$$

EM: Maths

Membership Weights

We can compute the "membership weight" of data point $\underline{x}_i$ in cluster k, given parameters Θ as

$$w_{ik} = p(z_{ik} = 1|\underline{x}_i, \Theta) = \frac{p_k(\underline{x}_i|z_k, \theta_k) \cdot \alpha_k}{\sum_{m=1}^{K} p_m(\underline{x}_i|z_m, \theta_m) \cdot \alpha_m}, \quad 1 \leq k \leq K, \quad 1 \leq i \leq N.$$

Gaussian Mixture Models

For $\underline{x} \in \mathcal{R}^d$ we can define a Gaussian mixture model by making each of the K components a Gaussian density with parameters $\underline{\mu}_k$ and Σ_k. Each component is a multivariate Gaussian density

$$p_k(\underline{x}|\theta_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu}_k)^t \Sigma_k^{-1}(\underline{x}-\underline{\mu}_k)}$$

with its own parameters $\theta_k = \{\underline{\mu}_k, \Sigma_k\}$.

EM: Maths

The EM Algorithm for Gaussian Mixture Models

We define the EM (Expectation-Maximization) algorithm for Gaussian mixtures as follows. The algorithm is an iterative algorithm that starts from some initial estimate of Θ (e.g., random), and then proceeds to iteratively update Θ until convergence is detected. Each iteration consists of an E-step and an M-step.

E-Step: Denote the current parameter values as Θ. Compute w_{ik} (using the equation above for membership weights) for all data points $\underline{x}_i$, $1 \leq i \leq N$ and all mixture components $1 \leq k \leq K$. Note that for each data point $\underline{x}_i$ the membership weights are defined such that $\sum_{k=1}^{K} w_{ik} = 1$. This yields an $N \times K$ matrix of membership weights, where each of the rows sum to 1.

M-Step: Now use the membership weights and the data to calculate new parameter values. Let $N_k = \sum_{i=1}^{N} w_{ik}$, i.e., the sum of the membership weights for the kth component—this is the effective number of data points assigned to component k.

Specifically,

$$\alpha_k^{new} = \frac{N_k}{N}, \quad 1 \leq k \leq K.$$

EM: Maths

These are the new mixture weights.

$$\underline{\mu}_k^{new} = \left(\frac{1}{N_k}\right) \sum_{i=1}^{N} w_{ik} \cdot \underline{x}_i \quad 1 \leq k \leq K.$$

The updated mean is calculated in a manner similar to how we could compute a standard empirical average, except that the ith data vector $\underline{x}_i$ has a fractional weight w_{ik}. Note that this is a vector equation since $\underline{\mu}_k^{new}$ and $\underline{x}_i$ are both d-dimensional vectors.

$$\Sigma_k^{new} = \left(\frac{1}{N_k}\right) \sum_{i=1}^{N} w_{ik} \cdot (\underline{x}_i - \underline{\mu}_k^{new})(\underline{x}_i - \underline{\mu}_k^{new})^t \quad 1 \leq k \leq K.$$

Again we get an equation that is similar in form to how we would normally compute an empirical covariance matrix, except that the contribution of each data point is weighted by w_{ik}. Note that this is a matrix equation of dimensionality $d \times d$ on each side.

The equations in the M-step need to be computed in this order, i.e., first compute the K new α's, then the K new $\underline{\mu}_k$'s, and finally the K new Σ_k's.

After we have computed all of the new parameters, the M-step is complete and we can now go back and recompute the membership weights in the E-step, then recompute the parameters again in the E-step, and continue updating the parameters in this manner. Each pair of E and M steps is considered to be one iteration.

EM: Maths

Initialization and Convergence Issues for EM

The EM algorithm can be started by either initializing the algorithm with a set of initial parameters and then conducting an E-step, or by starting with a set of initial weights and then doing a first M-step. The initial parameters or weights can be chosen randomly (e.g. select K random data points as initial means and select the covariance matrix of the whole data set for each of the initial K covariance matrices) or could be chosen via some heuristic method (such as by using the k-means algorithm to cluster the data first and then defining weights based on k-means memberships).

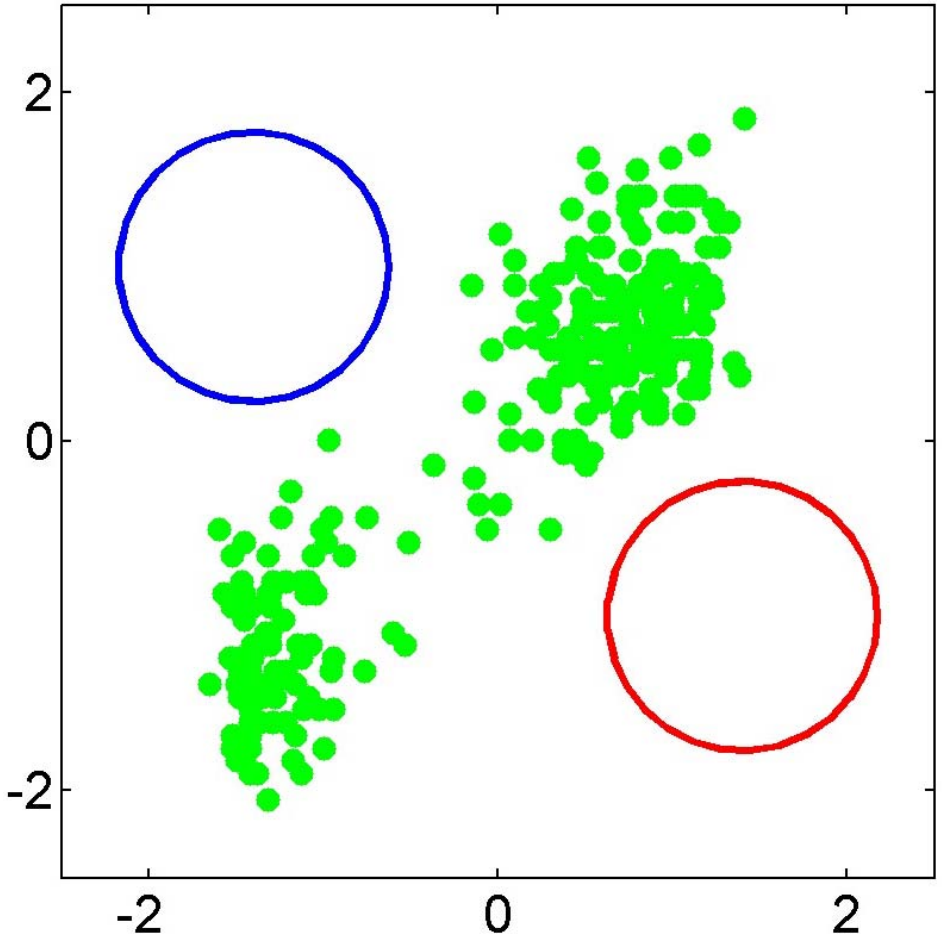
Convergence is generally detected by computing the value of the log-likelihood after each iteration and halting when it appears not to be changing in a significant manner from one iteration to the next. Note that the log-likelihood (under the IID assumption) is defined as follows:

$$\log l(\Theta) = \sum_{i=1}^{N} \log p(\underline{x}_i|\Theta) = \sum_{i=1}^{N} \left(\log \sum_{k=1}^{K} \alpha_k p_k(\underline{x}_i|z_k, \theta_k) \right)$$

where $p_k(\underline{x}_i|z_k, \theta_k)$ is the Gaussian density for the kth mixture component.

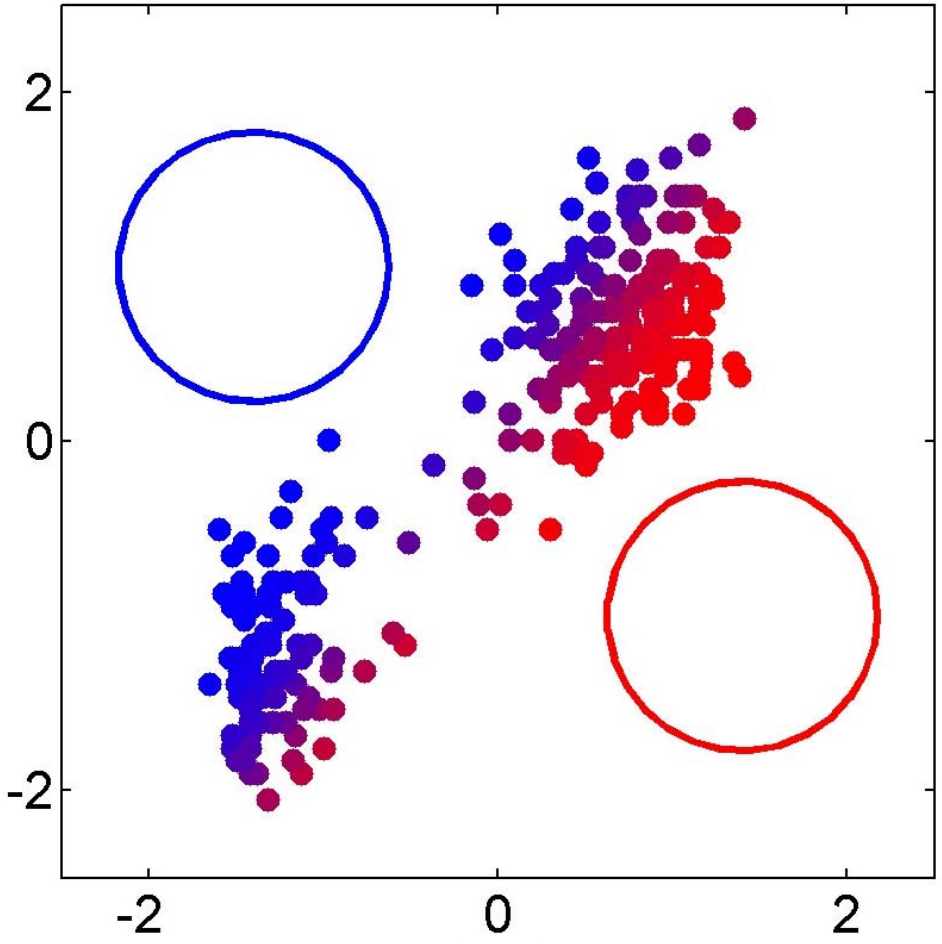
EM algorithm

- Example of initialization: two classes normal distribution.
- μ=random
- Σ=Identity



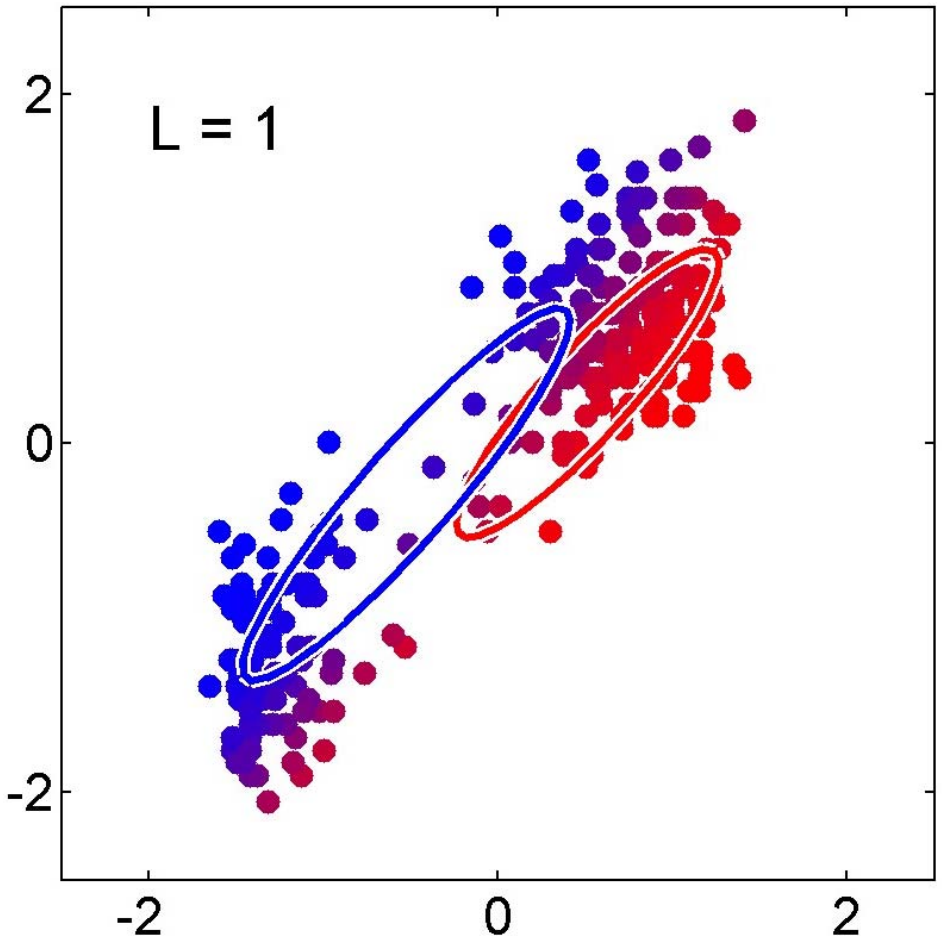
EM algorithm

- First 'expectation' step. For each point, to which class is expected to belong?



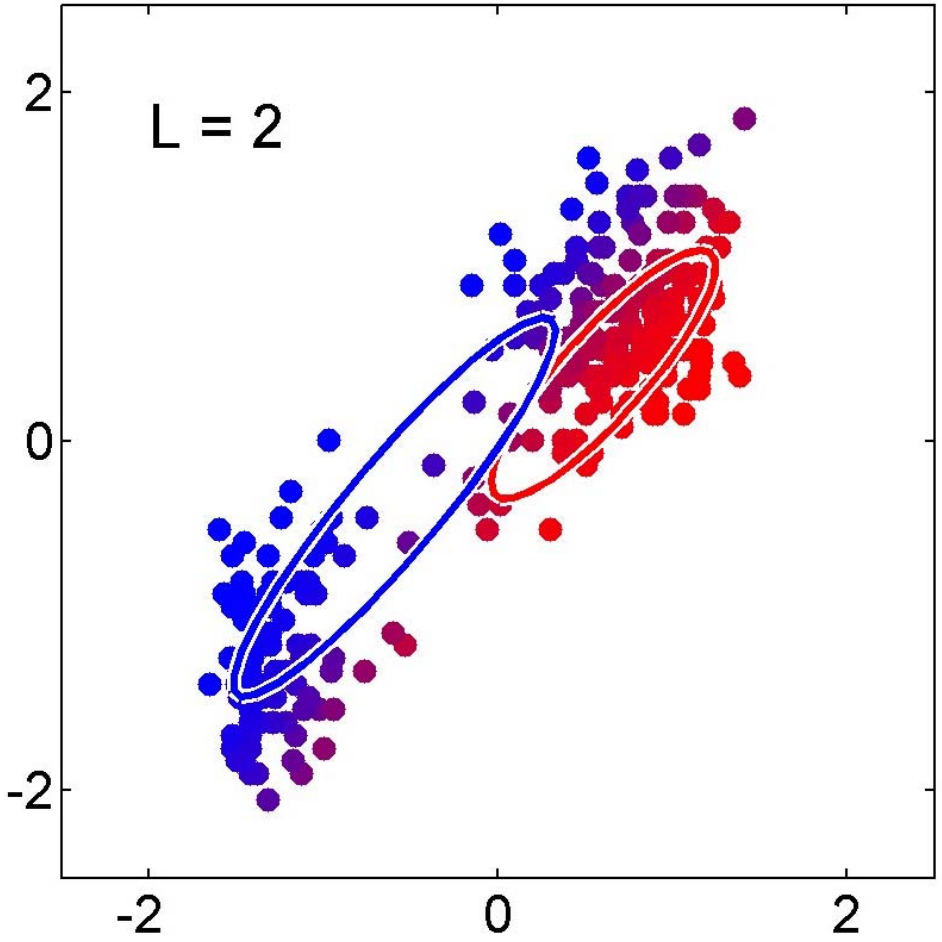
EM algorithm

- First 'maximization' step. For expected class of the points, which distribution maximizes the likelihood?
- Compute μ,Σ



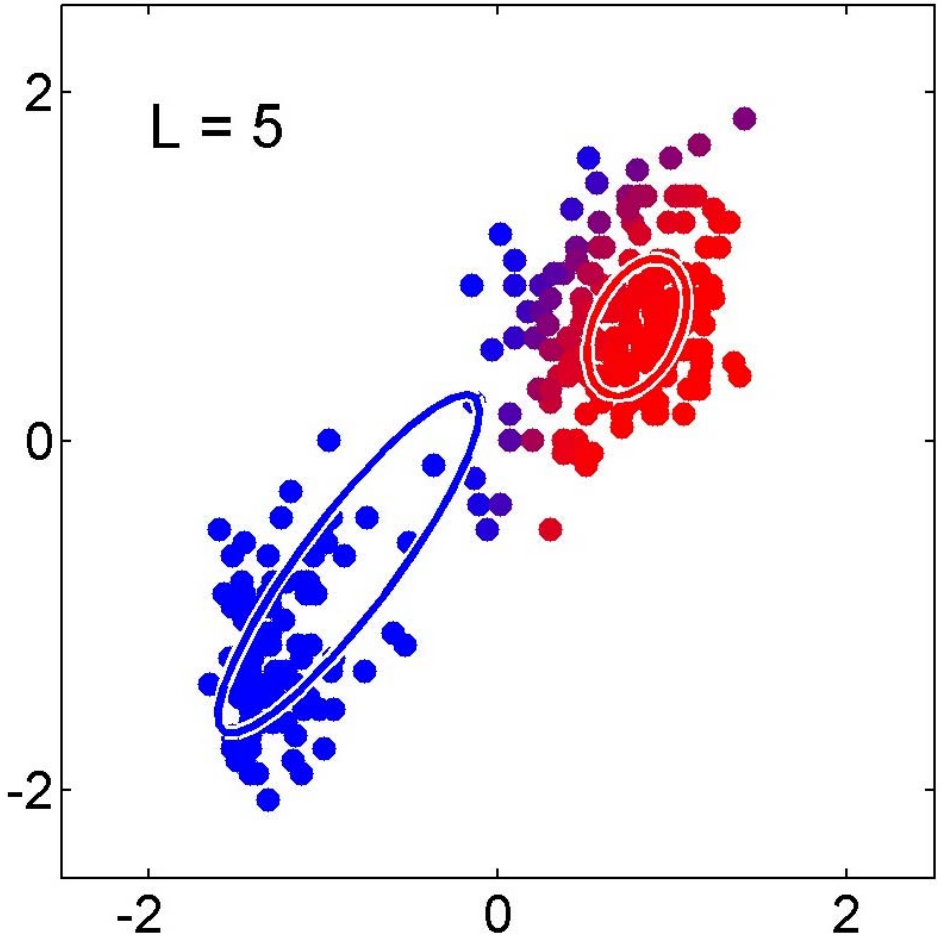
EM algorithm

- After second step.



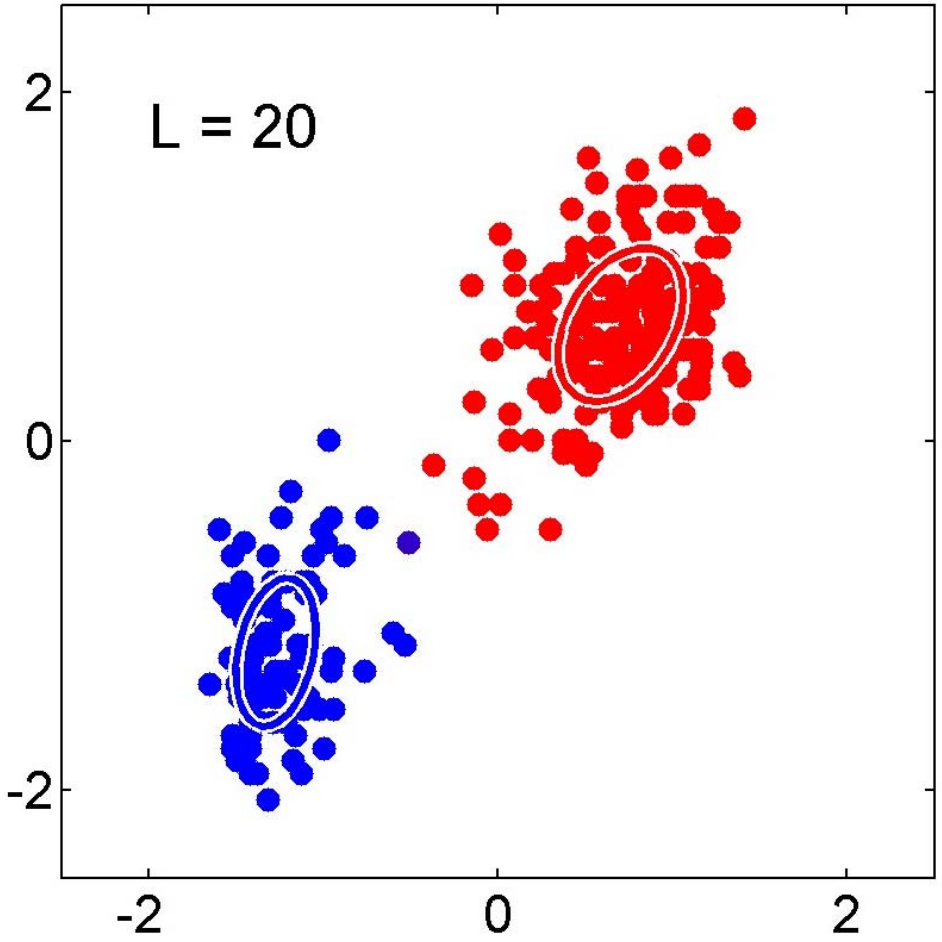
EM algorithm

- After five steps.



EM algorithm

- After convergence (twenty iterations).



Evaluation

Advantages / Disadvantages

- The EM algorithm guarantees to converge
- Having an a-priori knowledge of the data distribution can be used with in the segmentation framework

- EM algorithm depends on the parameter initialization. Some initializations may lead to local optimum (not the correct one)
- We need to know the number of classes beforehand
- No spatial information. This sometimes leads to incorrect clustering. Markov Random Fields are use to improve this issue

Conclusions

- EM is a simple and popular method for clustering in Euclidean space. It provides an iterative procedure to compute a series of log-likelihood based on incomplete data. It can be widely used for image segmentation, diagnosis, distribution discovery, etc

- Difference with K-means: EM estimates the parameters of the distribution (mean and covariance), and it does the clustering based on prior probability instead of the distance to the centroids (means)

- Because the quality of the final results depends largely on the initial set of clusters, the first parameters should be carefully selected

References

- [1] A.P.Dempster et al “Maximum-likelihood from incomplete data” Journal of the Royal Statistical Society. Series B Methodological), Vol. 39, No. 1. (1977), pp. 1-38
- [2] F. Dellaert, “The Expectation Maximization Algorithm”, Tech. Rep. GIT-GVU-02-20, 2002
- [3] K. Andersson, Presentation: Model Optimization using the EM algorithm, COSC 7373, 2001
- [4] Chris Ding and Xiaofeng He. "K-means Clustering via Principal Component Analysis". Machine Learning (ICML 2004), pp 225-232. July 2004
- Graphics and pictures from Cristopher M. Bishop. Microsoft research, Cambridge