

Introduction to Clustering

K-means and Hierarchical Clustering

Techniques

Chrisantha Fernando

Computational Genomics

K-means clustering

- We wish to identify groups or clusters of data points in a multidimensional space
- Suppose data set $\{x_1, x_2, \dots, x_N\}$ where x_i is a vector.
- There are N observations (points) of a random D -dimensional Random Variable x .
- Goal: Partition dataset into some number K of clusters, where K is given.

What is a cluster?

- Intuitive notion: A group of data points whose inter-point distances are small compared with the distances to points outside the cluster.
- Lets formalise this notion
- Let \mathbf{u}_k be a set of D-dimensional vectors $k = 1, 2, \dots, K$.
- \mathbf{u}_k is in some sense a prototype associated with the k^{th} cluster, a sort of cluster centre.

Goal of K-means clustering

- Find an assignment of data points to clusters, as well as a set of vectors $\{\mathbf{u}_k\}$ such that the sum of the squares of the distances of each data point to its closest vector \mathbf{u}_k is a minimum.

Notation: 1 of K coding scheme

- For each data point \mathbf{x}_n we introduce binary indicator variables r_{nk} from the elements $\{0,1\}$, where $k = 1, \dots, K$ describing which of the K clusters the data point x_n is assigned to.
- E.g. if data point \mathbf{x}_n is assigned to cluster k then
 $r_{nk} = 1$, and $r_{nj} = 0$ for j not equal to k .

Cost Function (also called objective function or distortion measure)

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \left\| x_n - \mu_k \right\|^2$$

This is the sum of squares of the distances of each data point to its assigned vector μ_k

Goal: Find values of $\{r_{nk}\}$ and $\{\mu_k\}$ so as to minimize J .

Goal (put another way): Minimize J w.r.t. the variables $\{r_{nk}\}$ and $\{\mu_k\}$.

How do we minimize J?

- Use an iterative procedure in which each iteration involves two successive steps corresponding to successive optimizations w.r.t. r_{nk} and u_k .
- First choose some initial values for the u_k 's.
 - Then in the first phase minimize J w.r.t. r_{nk} , keeping the u_k fixed.
 - In the second phase minimize J w.r.t. u_k keeping the r_{nk} fixed.
- Repeat until convergence.

This is the 2-phase K-means algorithm

- The first phase (updating the assignments r_{nk}) corresponds to the E (Expectation) step
- The second phase (updating the centers u_k) corresponds to the M (Maximization step)
- Of an algorithm called the EM algorithm which takes sense of this somewhat ad-hoc algorithm that I've told you.... For this reason we'll call these the E step and the M step from now on.

The Two Phases Again!

- E Step. Assign data points to clusters, i.e. calculate the r_{nk} values.
- M Step. Change the cluster centers, i.e. change the \mathbf{u}_k values.

E Step. How to assign r_{nk} values?

- Notice that J in this equation is a $J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$ linear function of r_{nk} .
- This optimization can easily be performed.
 - The terms for different n (data points) are independent so optimize for each n separately by choosing r_{nk} to be 1 for whichever value of k gives the minimum value of $\|x_n - \mu_k\|^2$
 - In other words, simply assign the n^{th} data point x_n to the closest cluster center μ_k .

E Step. How to assign r_{nk} values?

- Formally...

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \|x_n - \mu_j\|^2 \\ 0 & \text{otherwise} \end{cases}$$

M Step. How to update u_k centers?

- With r_{nk} assignments held fixed. We see that the objective function J is a quadratic function of u_k . $J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$
- Therefore it can be minimized by setting its derivative w.r.t. u_k to zero giving.

$$0 = 2 \sum_{n=1}^N r_{nk} \|x_n - \mu_k\| \quad \text{solved to give} \quad u_k = \frac{\sum r_{nk} x_n}{\sum_n r_{nk}}$$

M Step. How to update u_k centers?

- With r_{nk} assignments held fixed. We see that the objective function J is a quadratic function of u_k .

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$

- Therefore it can be minimized by setting its derivative w.r.t. u_k to zero giving.

$$0 = 2 \sum_{n=1}^N r_{nk} \|x_n - \mu_k\| \quad \text{solved to give} \quad u_k = \frac{\sum r_{nk} x_n}{\sum_n r_{nk}}$$

sum of all the vectors x_n
assigned to this cluster.

Mean of all vectors x_n assigned to this cluster

Number of points assigned to cluster k

M Step. How to update u_k centers?

- That's why this algorithm is the K-means algorithm, the M step updates the u_k means to the mean vector of the points assigned to its cluster.

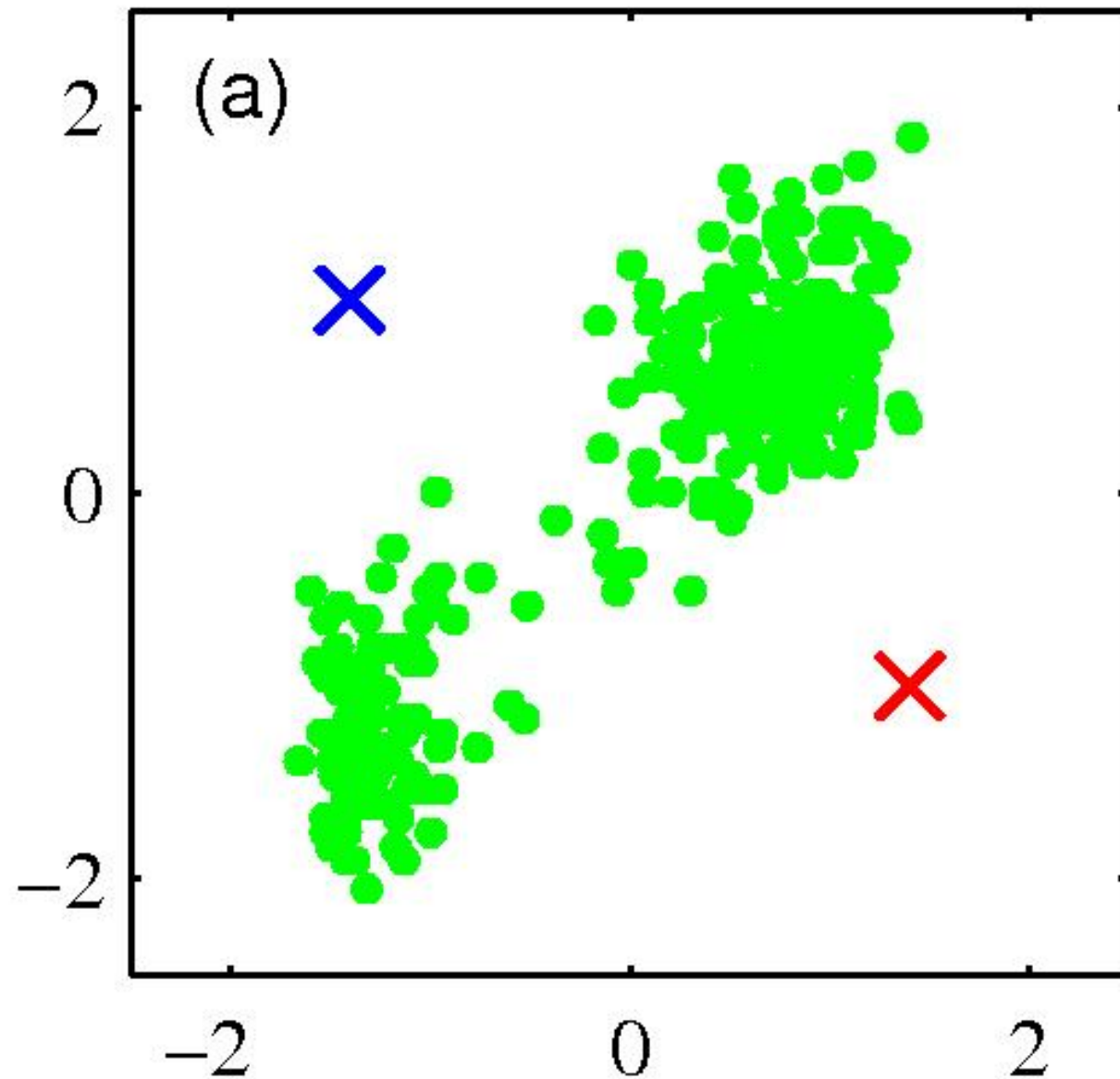
Convergence Properties of K-means

- These two phases are repeated until convergence, i.e. until no further change in the assignments occurs (or until some maximum number of iterations is exceeded).
- Each phase reduces the objective function J , therefore convergence is assured.
- However, the convergence may be on a local and not a global optimum of J .

An Example

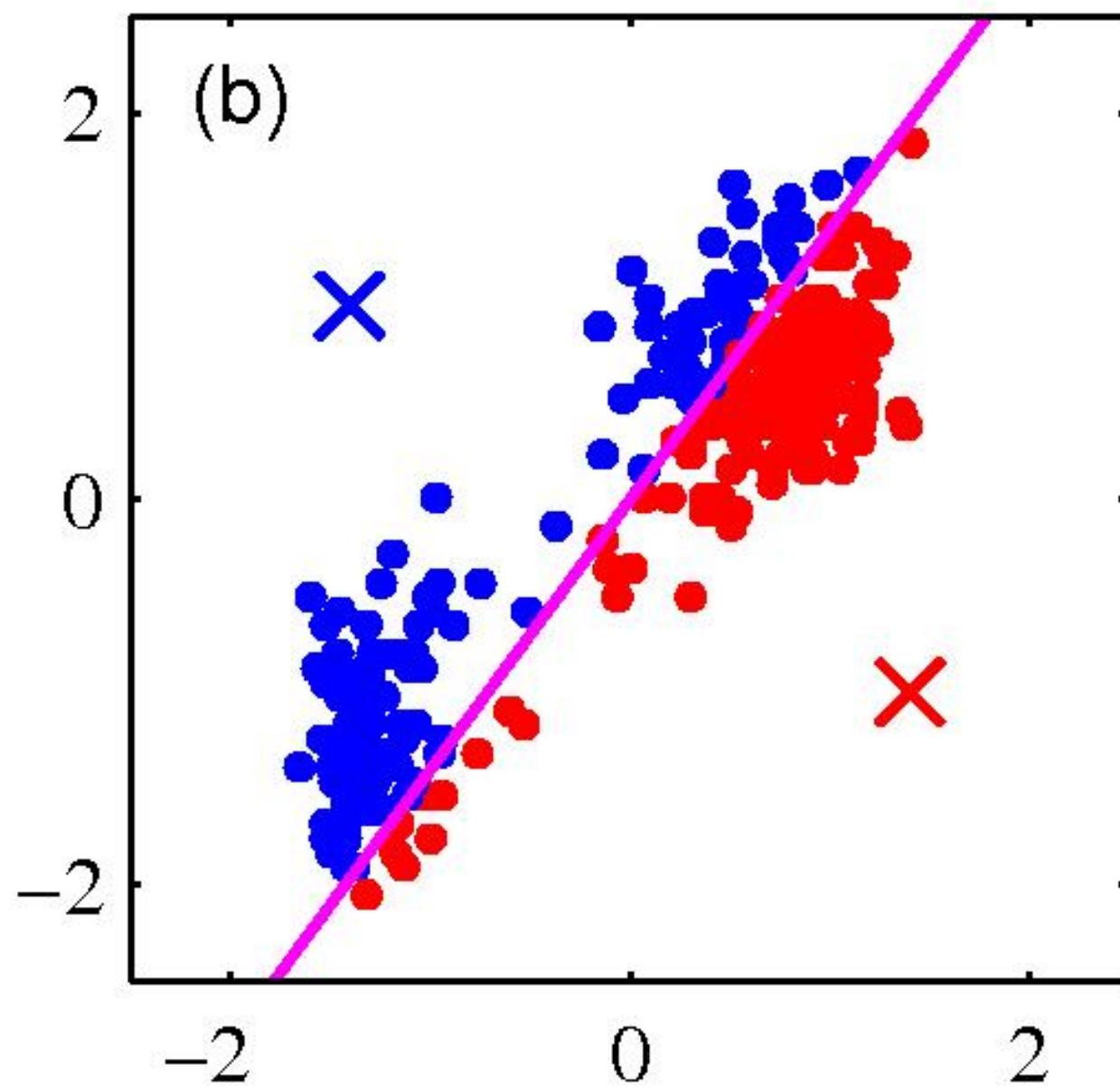
- First linearly re-scale data so they have zero mean and unit standard deviation.
- Choose value of K , e.g. $K = 2$
- Plot cost function after each E step (blue) and M step (red) of the K-means algorithm.
- This tells you when it has converged.

Initialize u_k centers randomly (or more sensibly)

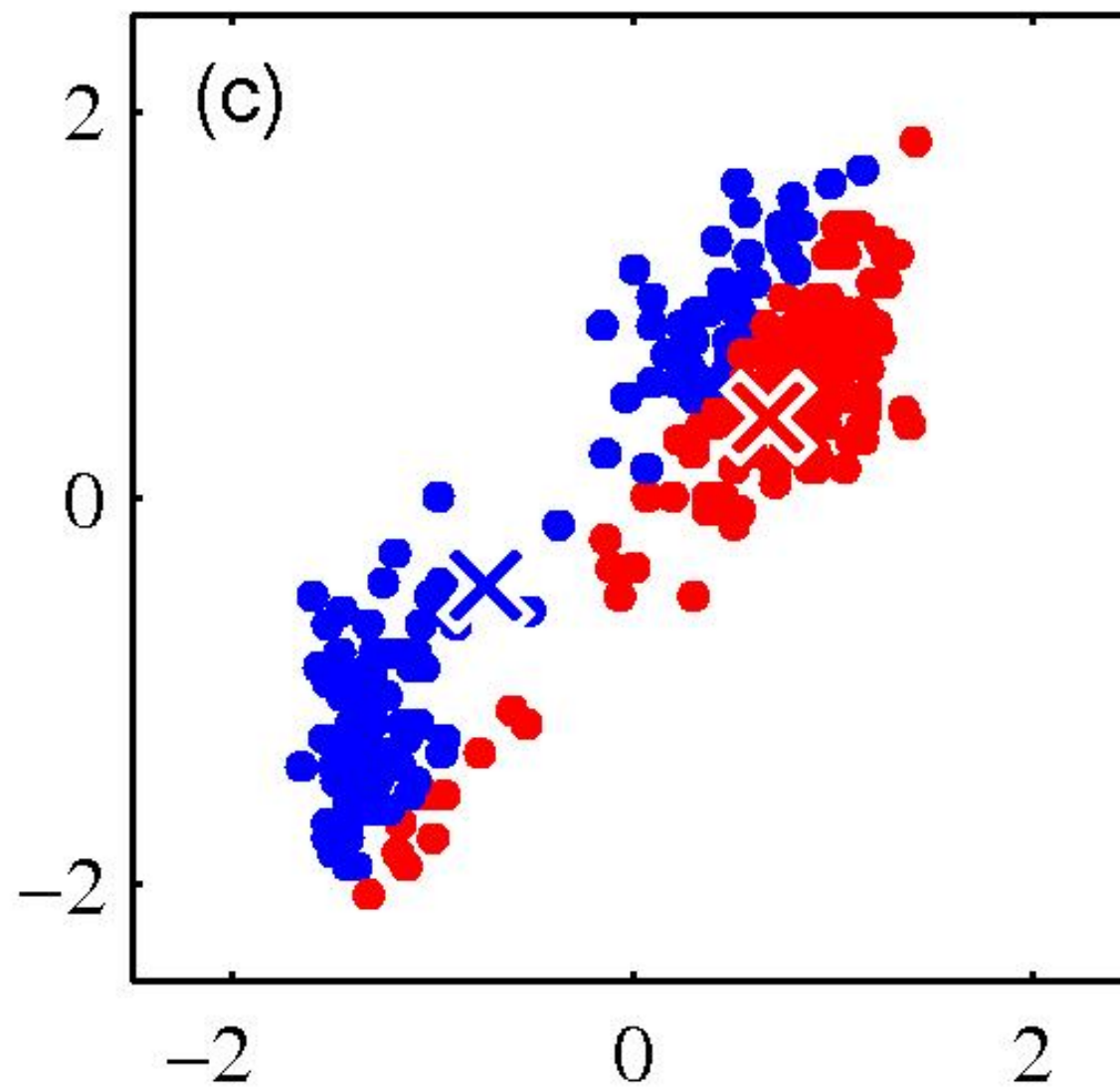


Old Faithful data set 272 eruptions (duration
,time to next eruption)

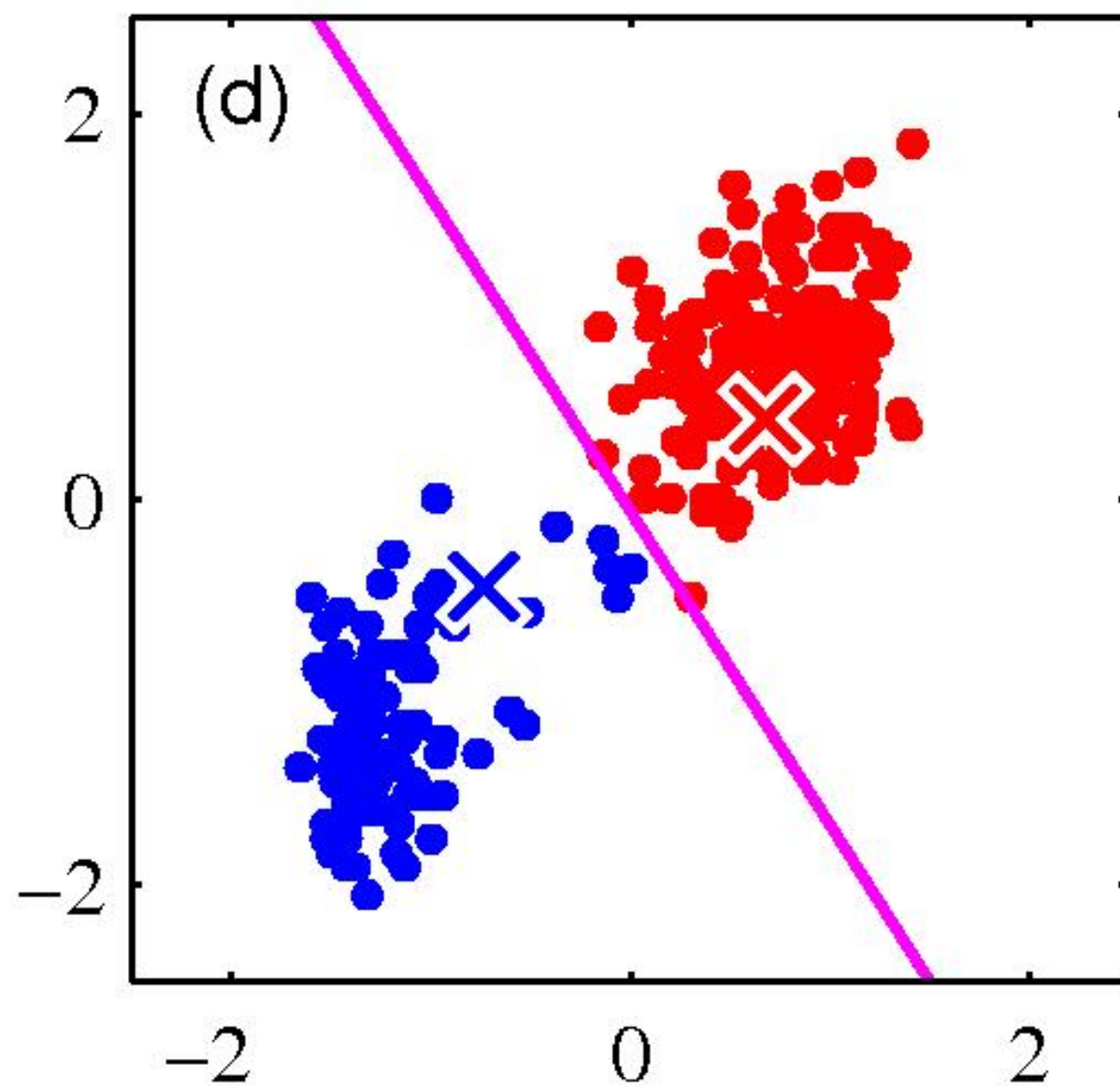
E Step



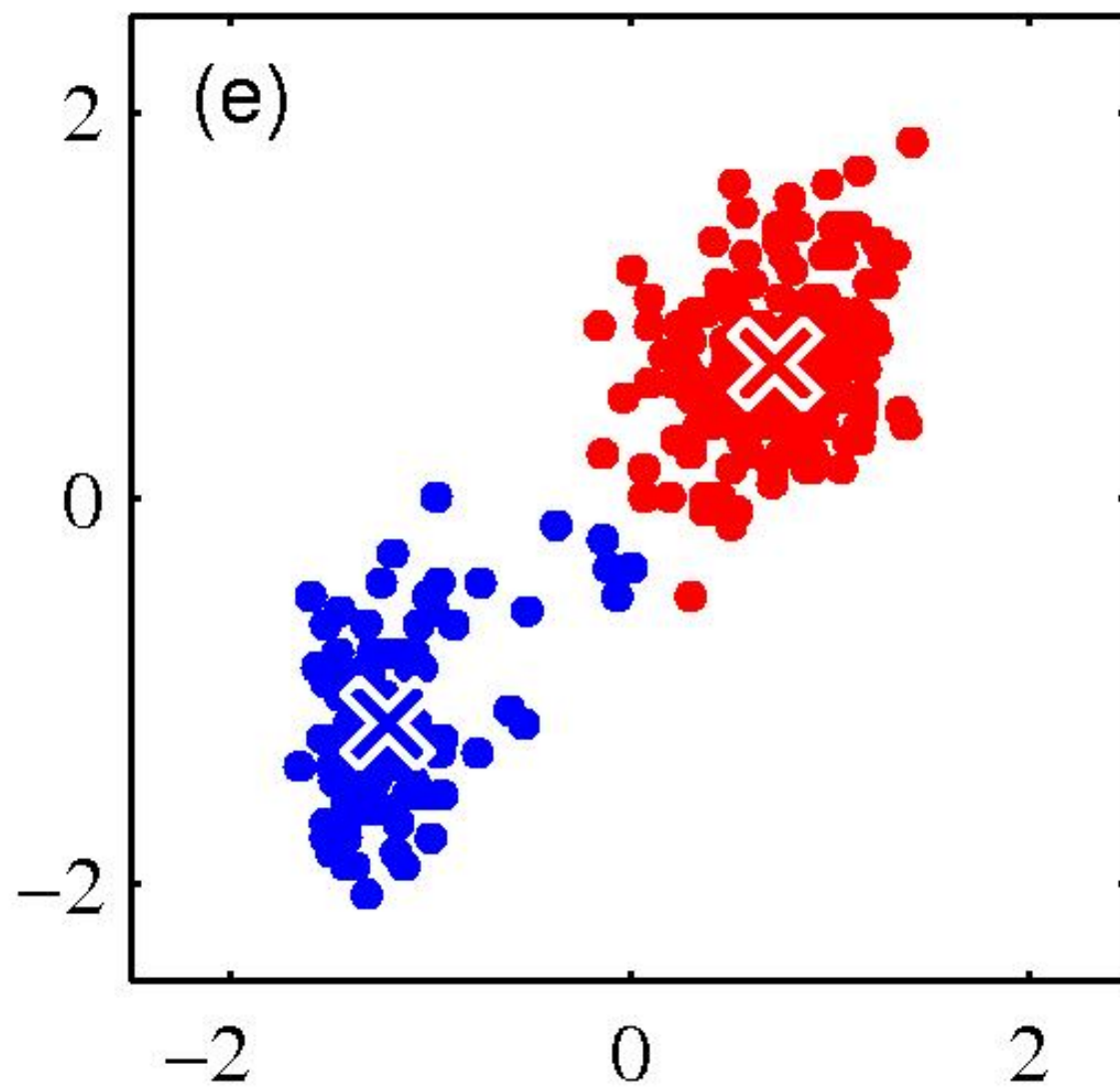
M Step



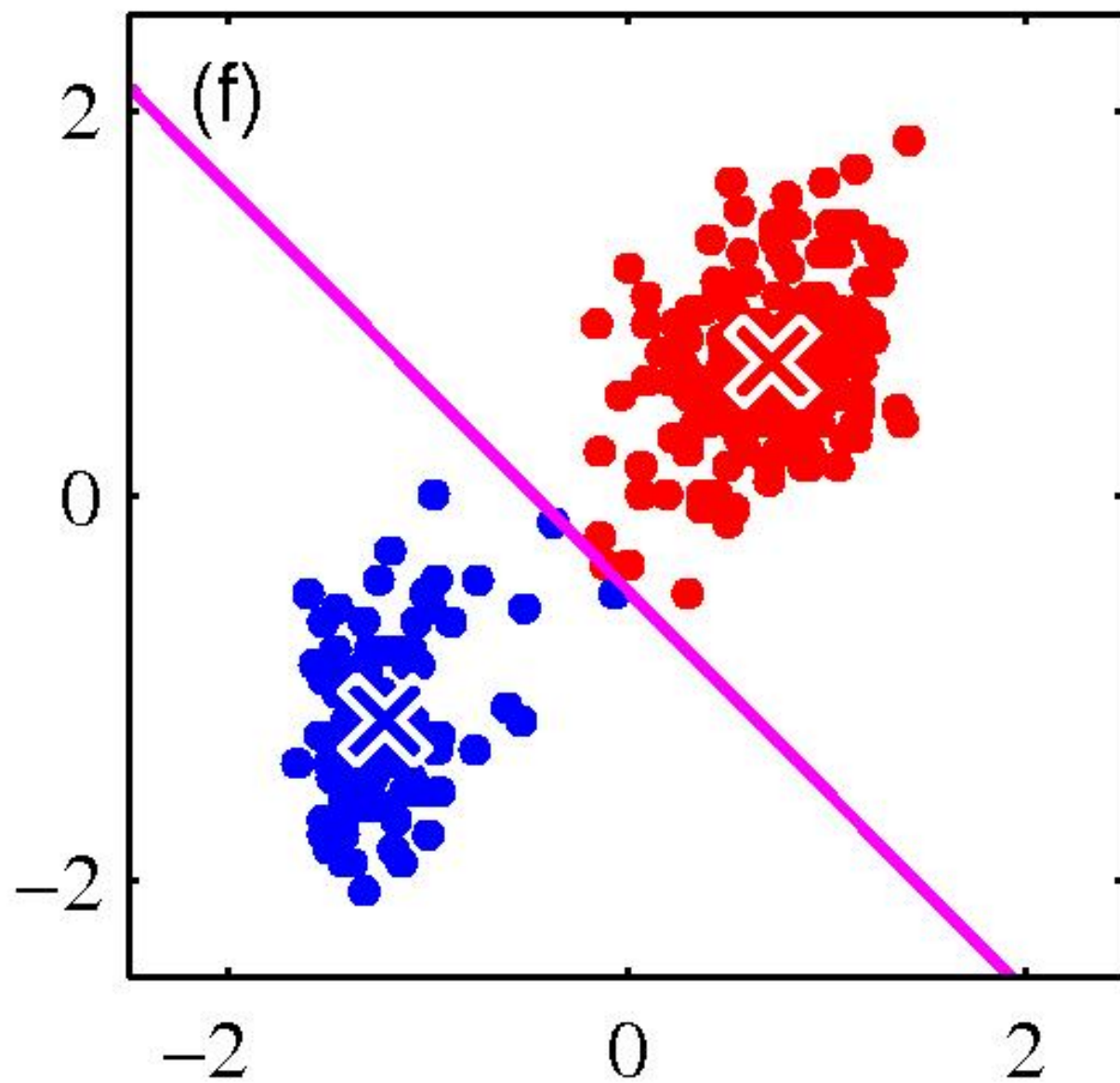
E Step



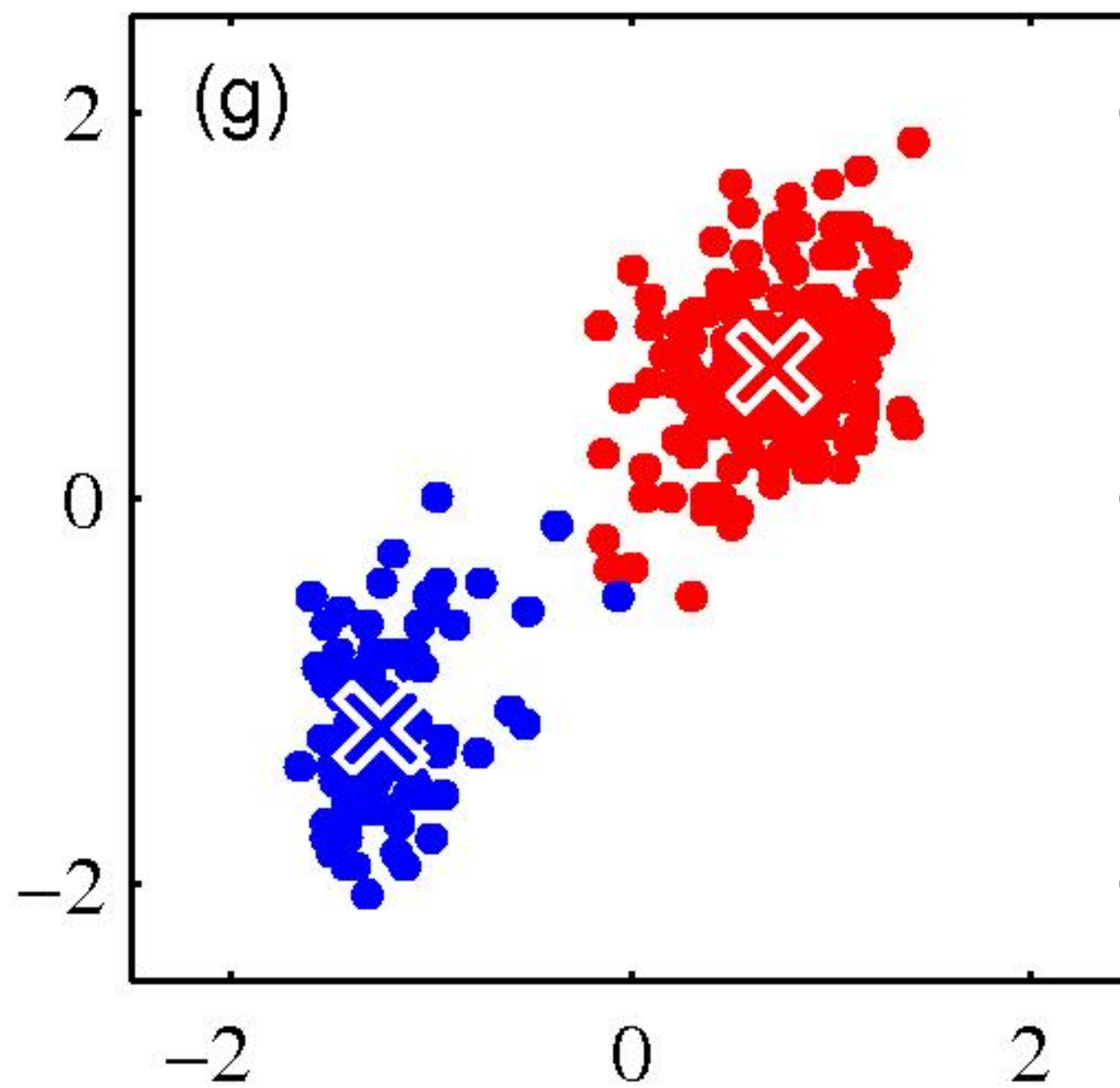
M Step



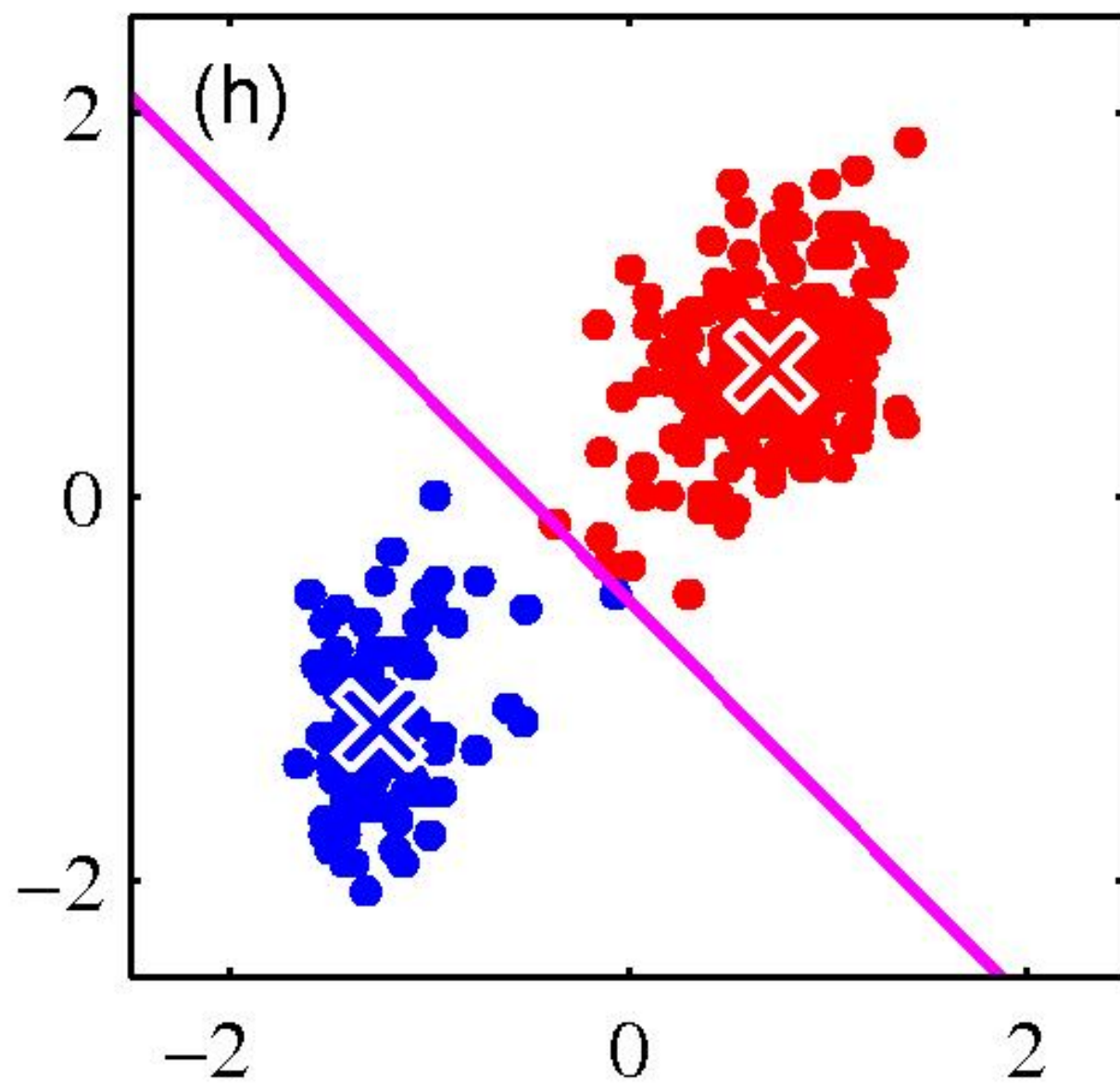
E Step



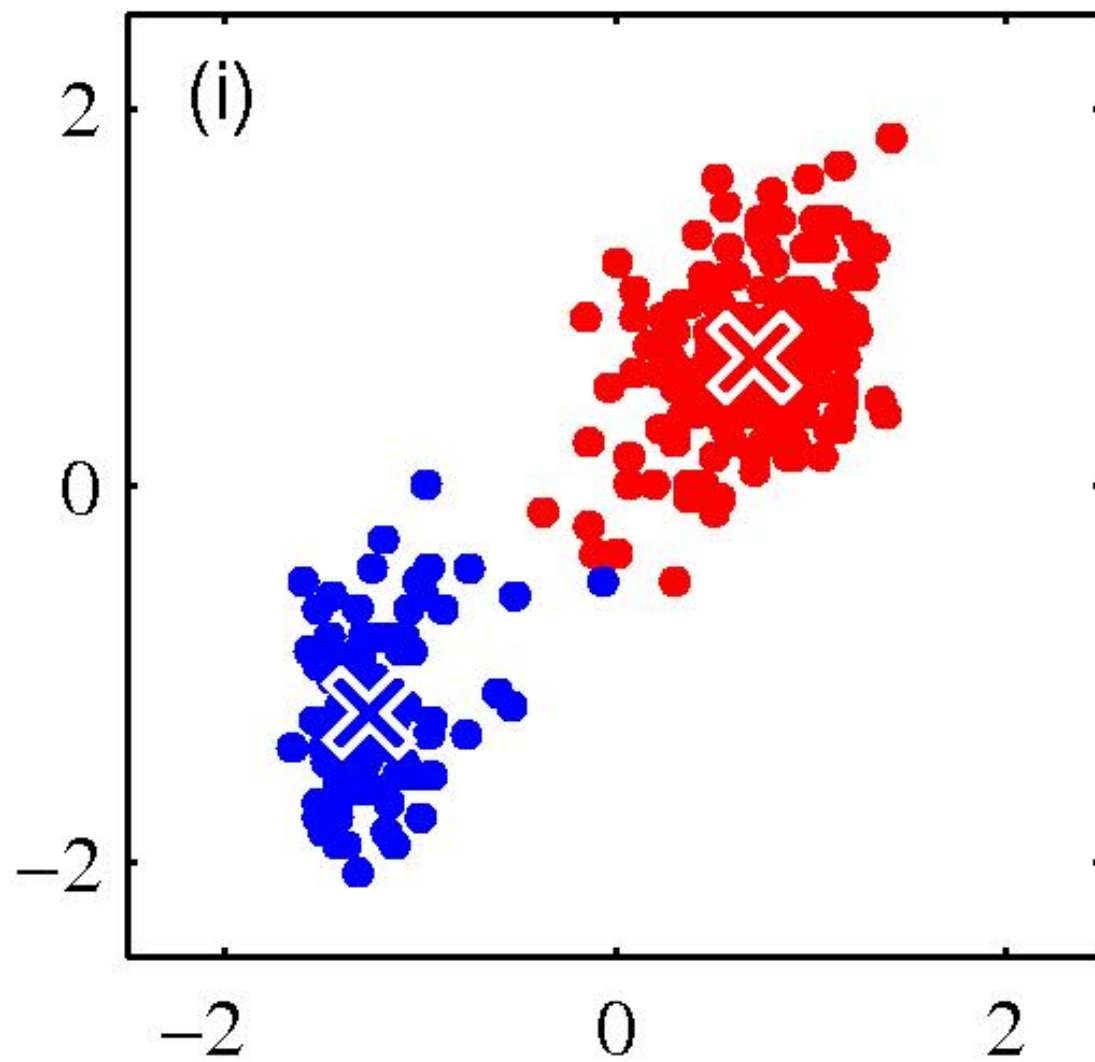
M Step

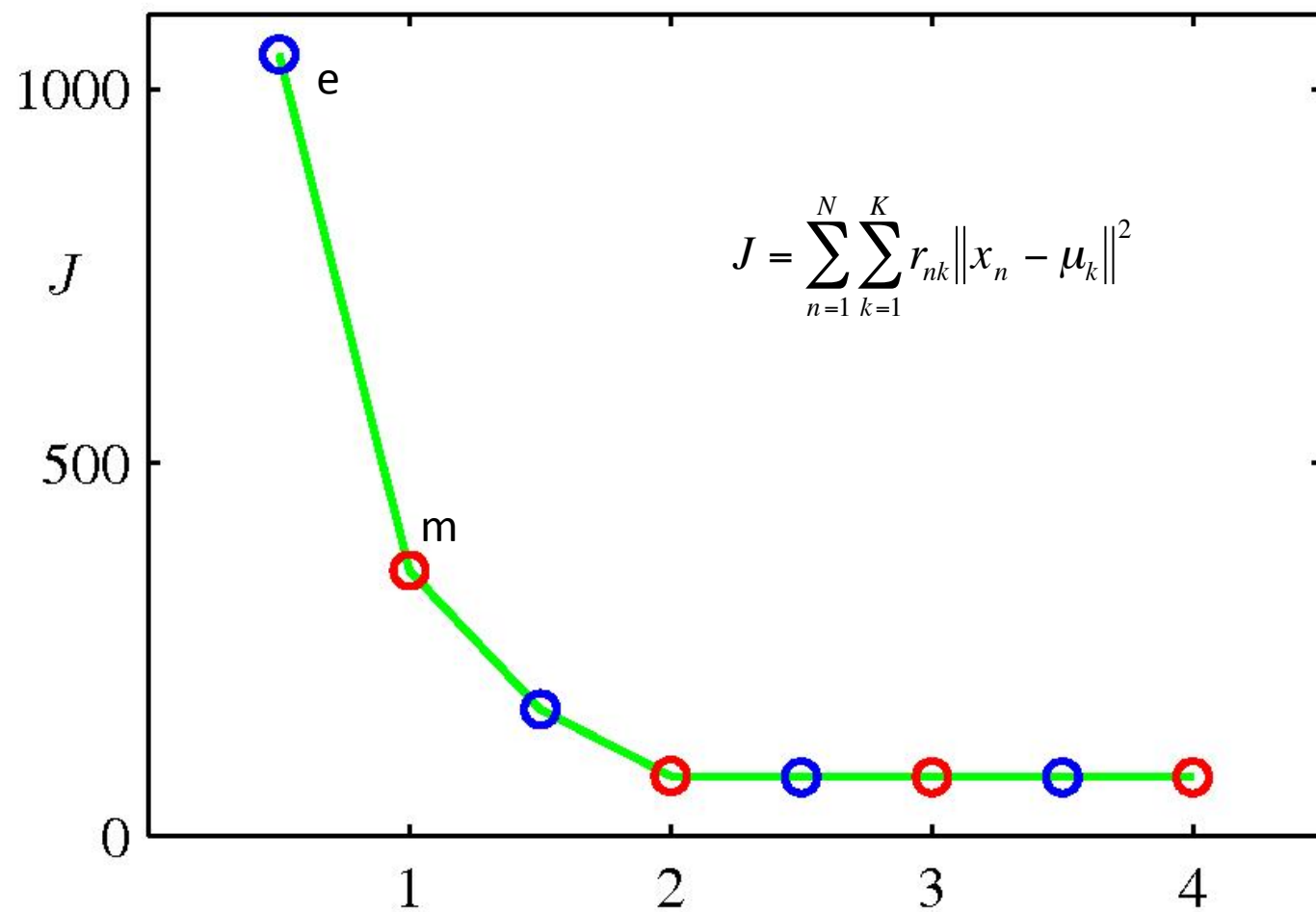


E Step



M Step (Converged)





Problems

- Its slow because each E step requires computation of Euclidean distance between every prototype vector and every data point.
- This is a batch version (whole data set is used to update u_k).
- The crucial thing is that K-means is based on the use of squared Euclidean distance as a measure of dissimilarity between a data point and its prototype vector.
 - Inappropriate if variables represent categorical labels.
 - Can make determination of cluster means non-robust to outliers.

A Generalised Dissimilarity Measure

- $V(\mathbf{x}, \mathbf{x}')$ between two vectors \mathbf{x} and \mathbf{x}' has the following distortion measure...

$$\tilde{J} = \sum_{n=1}^N \sum_{k=1}^K r_{nk} V(x_n, \mu_k)$$

Examples of K-means

- Image segmentation (spatial proximity ignored)

$K = 2$



$K = 3$



$K = 10$



Original image

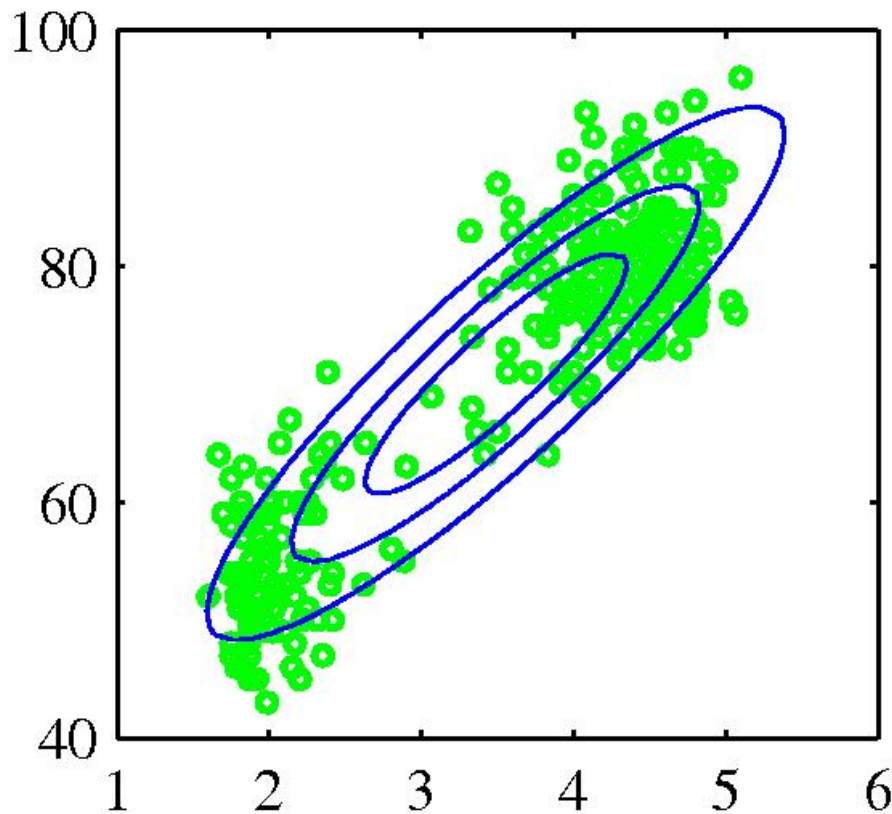


$x_i \{R,G,B\}$ value $[0,1]$

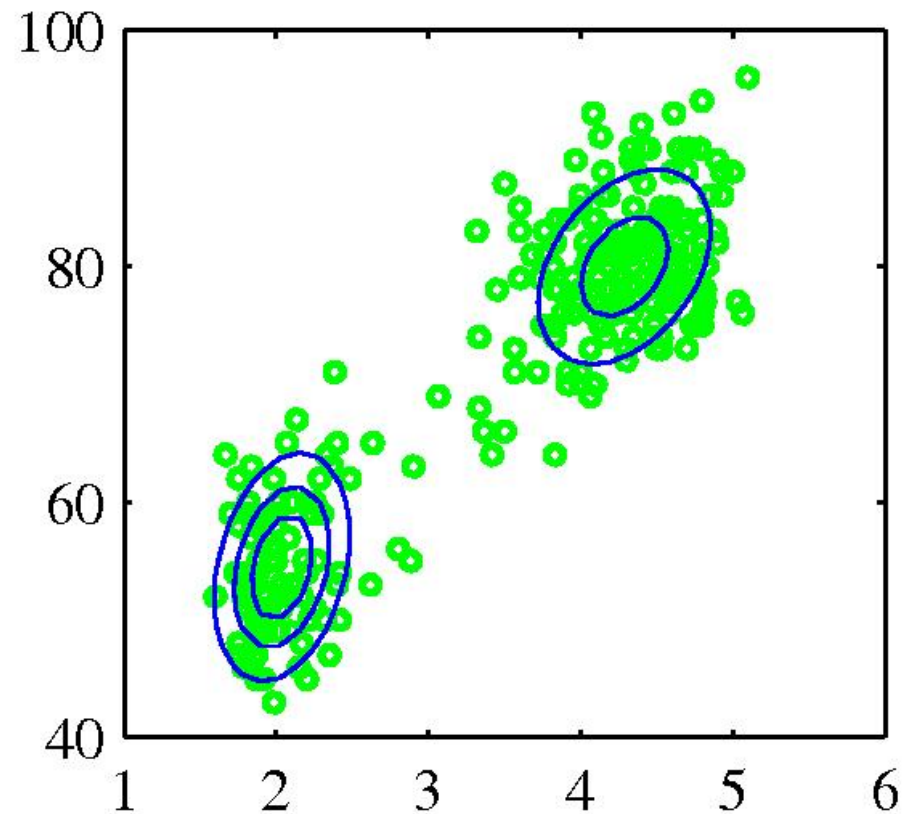
Each data point is ONLY assigned to
one cluster

- Even though some data points are much closer to a center u_k than others.
- Perhaps a probabilistic formulation would be better for capturing this uncertainty?

An Alternative: Mixture of Gaussian Model



Fitting a single Gaussian is unable to capture this structure

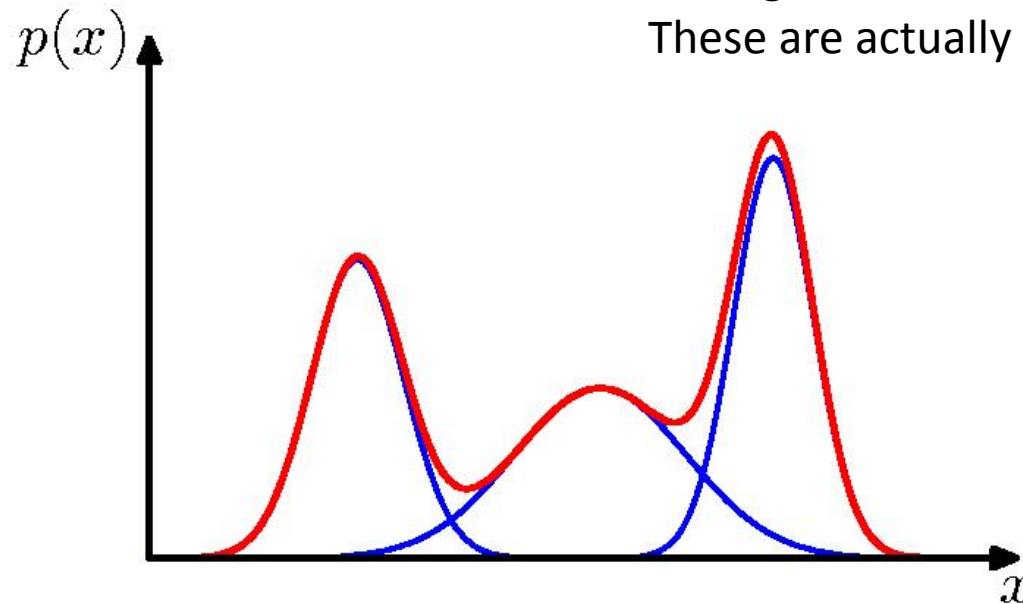


Linear superposition of two Gaussians gives better characterization of data.

Superimpose K Gaussian densities

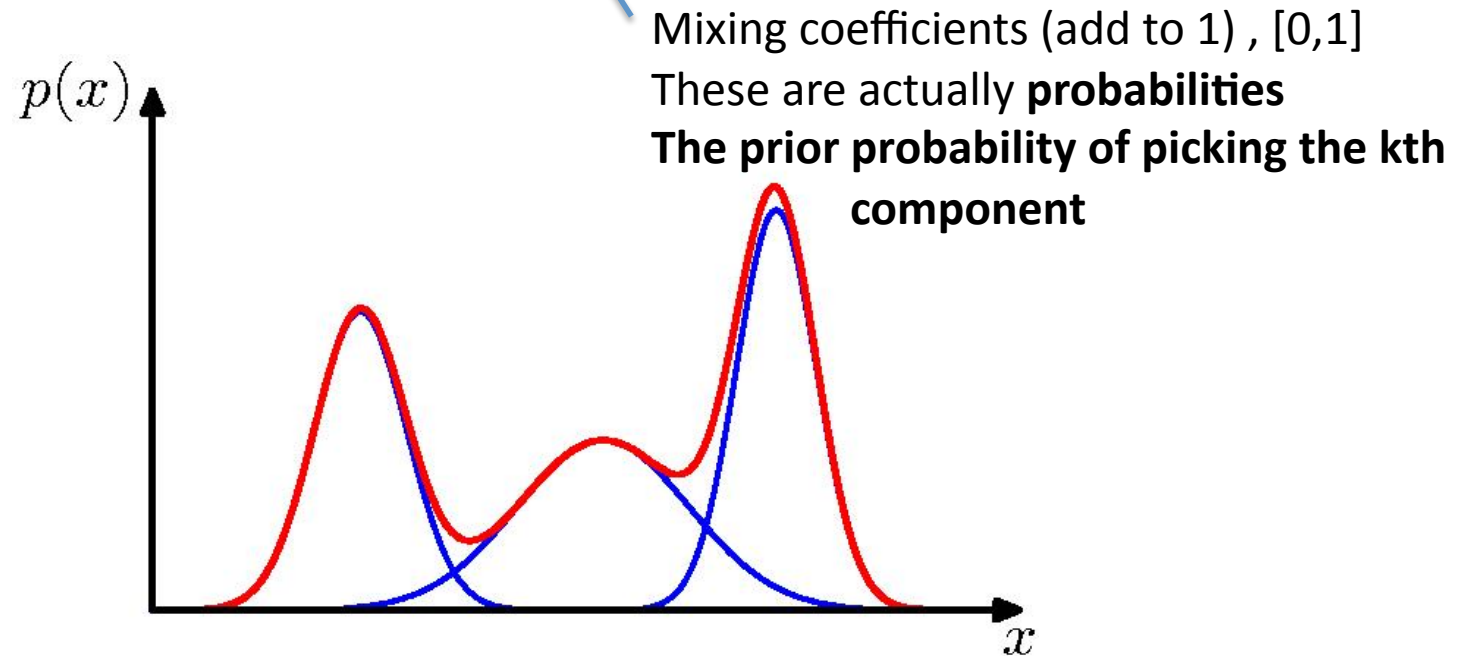
$$p(x) = \sum_{k=1}^K \pi_k N(x | u_k, \Sigma_k)$$

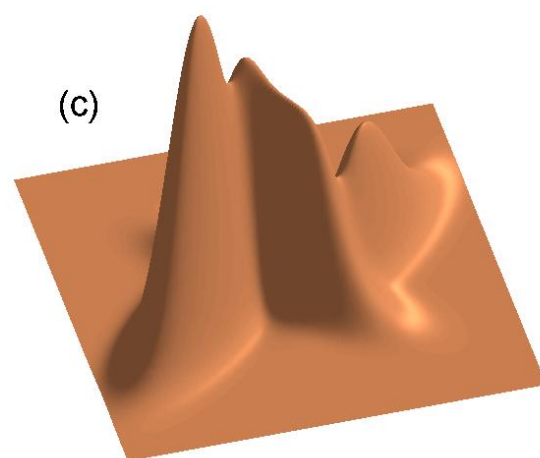
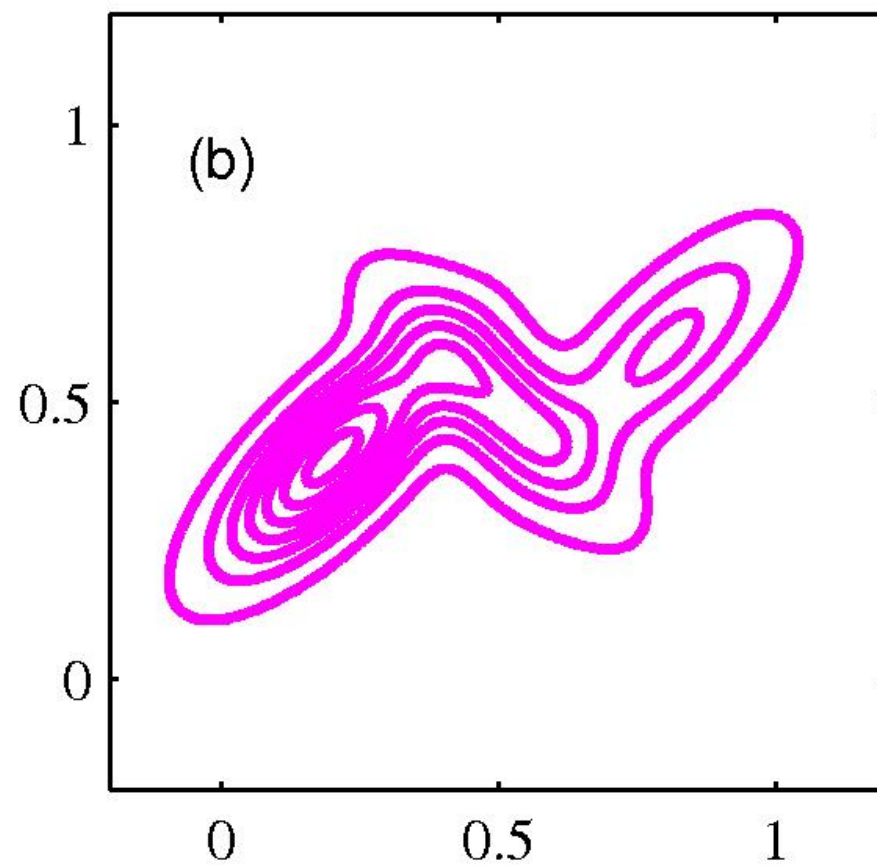
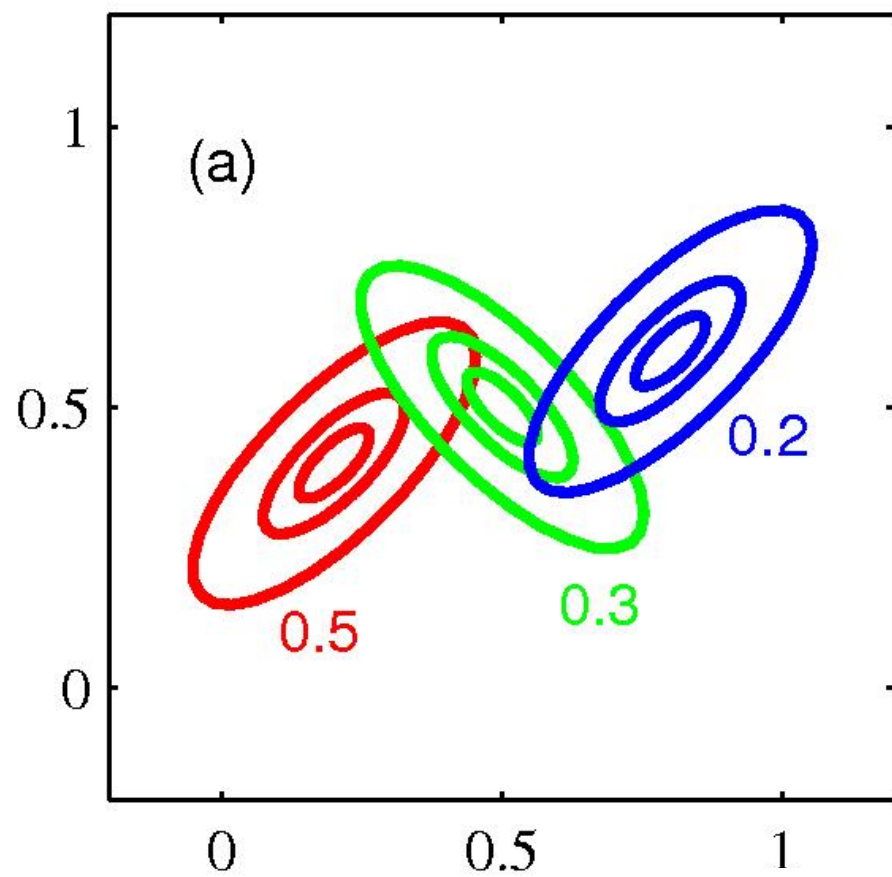
Mixing coefficients (add to 1) , [0,1]
These are actually **probabilities**



Superimpose K Gaussian densities

$$p(x) = \sum_{k=1}^K p(k) p(x | k)$$





The Posteriors $p(k|x)$ are called the responsibilities

Bayes theorem is used
to get posterior from
Prior x Likelihood

$$\gamma_k(x) \equiv p(k|x)$$

$$\gamma_k(x) \equiv \frac{p(k)p(x|k)}{\sum_l p(l)p(x|l)}$$

$$\gamma_k(x) \equiv \frac{\pi_k N(x|u_k, \Sigma_k)}{\sum_l \pi_l N(x|u_l, \Sigma_l)}$$

We can formulate Gaussian mixtures in terms of discrete **latent** variables.

- Introduce the RV \mathbf{z} having 1-of- representation (i.e. a 1 and the rest zeros), e.g.
- 0001, 0010, 0100, 1000, for $k = 4$
- $P(z_k=1) = \pi_k$ (mixing probability)

$$p(x) = \sum_z p(z) p(x | z) = \sum_{k=1}^K \pi_k N(x | u_k, \Sigma_k)$$

For every data point x_n there is a corresponding latent variable z_n .

$$P(z_k=1 | x)$$

Bayes theorem is used
to get posterior from
Prior x Likelihood

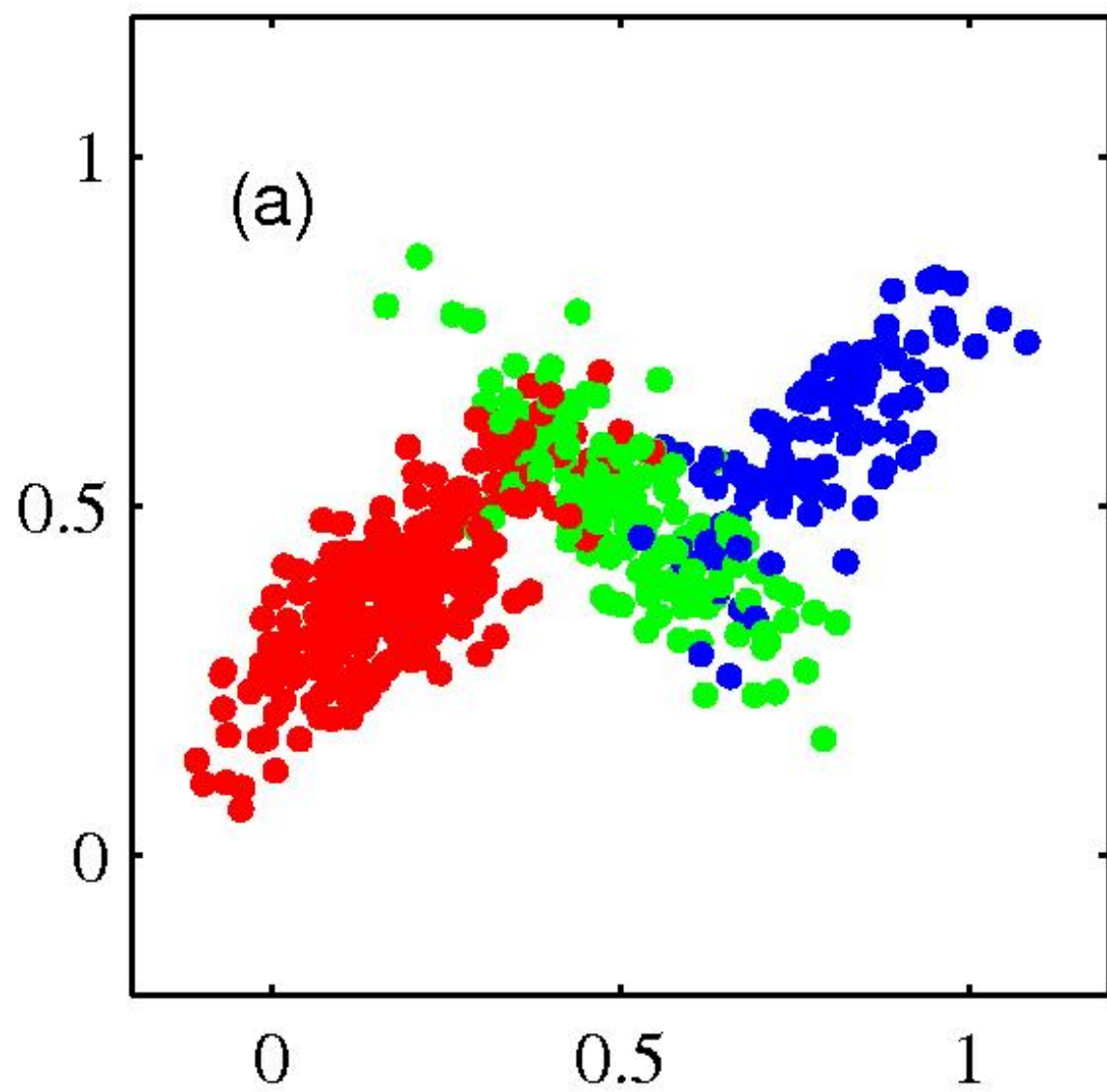
$$\gamma_k(z_k) \equiv p(z_k = 1 | x)$$

$$p(z_k = 1 | x) \equiv \frac{p(z_k = 1)p(x | z_k = 1)}{\sum_l p(z_l = 1)p(x | z_l = 1)}$$

$$p(z_k = 1 | x) \equiv \frac{\pi_k N(x | u_k, \Sigma_k)}{\sum_l \pi_l N(x | u_l, \Sigma_l)}$$

'Ancestral Sampling' from Joint Distribution $p(\mathbf{x}, \mathbf{z})$

- According to the Gaussian mixture model
- First generate a value for \mathbf{z} from the marginal distribution $p(\mathbf{z})$, e.g. (0.1,0.2,0.3,0.5) for $k = 4$.
- Then generate a value for $p(\mathbf{x} | \mathbf{z})$ from the conditional distribution (for that Gaussian).
- Plot points at the corresponding values of \mathbf{x} and colour them according to the value of \mathbf{z} .



Samples from Marginal distribution $p(x)$

- As before, but ignore the values of z of the points.

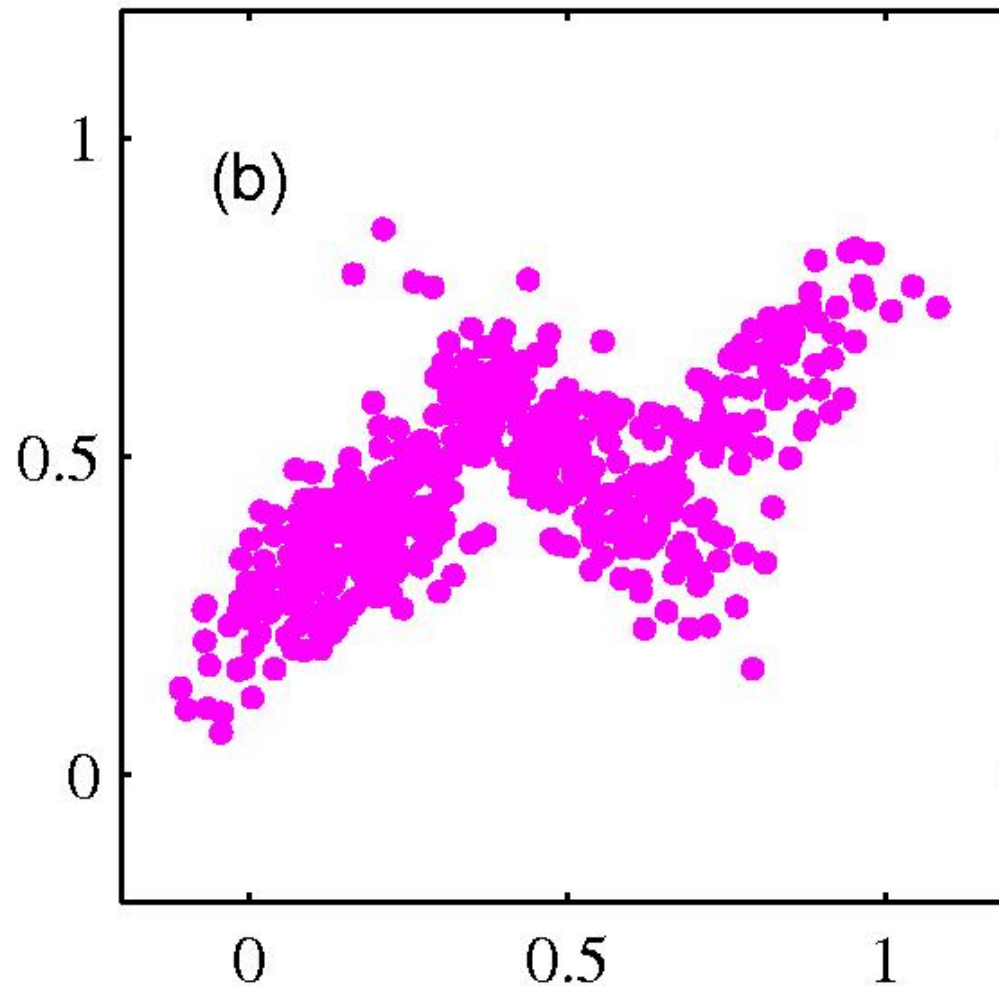
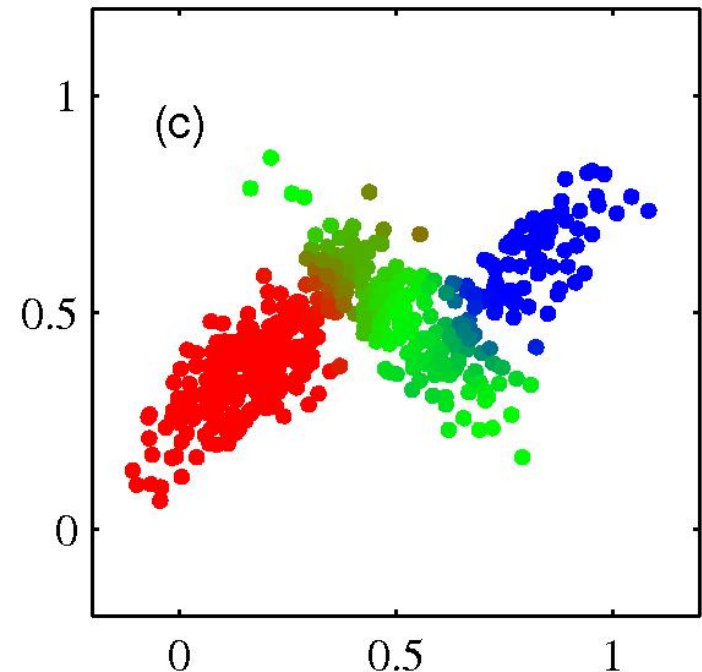


Illustration of responsibilities $p(z|x)$

- Evaluate the responsibilities $p(z|x)$ for every data point generated.
- For $k = 3$, use RGB encoding to represent $p(z_1|x)$, $p(z_2|x)$, $p(z_3|x)$ in colour.



How to fit a mixture of Gaussians?

- Find the MoG that maximizes the likelihood, i.e. the probability of the data given the model $p(\mathbf{x} | M)$.
- Use the Expectation Maximization algorithm (EM algorithm)

EM Algorithm

- Choose initial values for means, covariances, and mixing coefficients of MoG model.
- Repeat the E and M steps
- In the E (expectation) step: use current values for parameters to evaluate the posterior probabilities or responsibilities of each Gaussian (see prev. equation)
- In the M (maximization) step: re-estimate the means, covariances, and mixing coefficients.

M Step

- First update the means.

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n \quad N_k = \sum_{n=1}^N \gamma(z_{nk}) \quad N_k = \text{Effective number of points assigned to cluster } k$$

- Then use these means to update covariances

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T$$

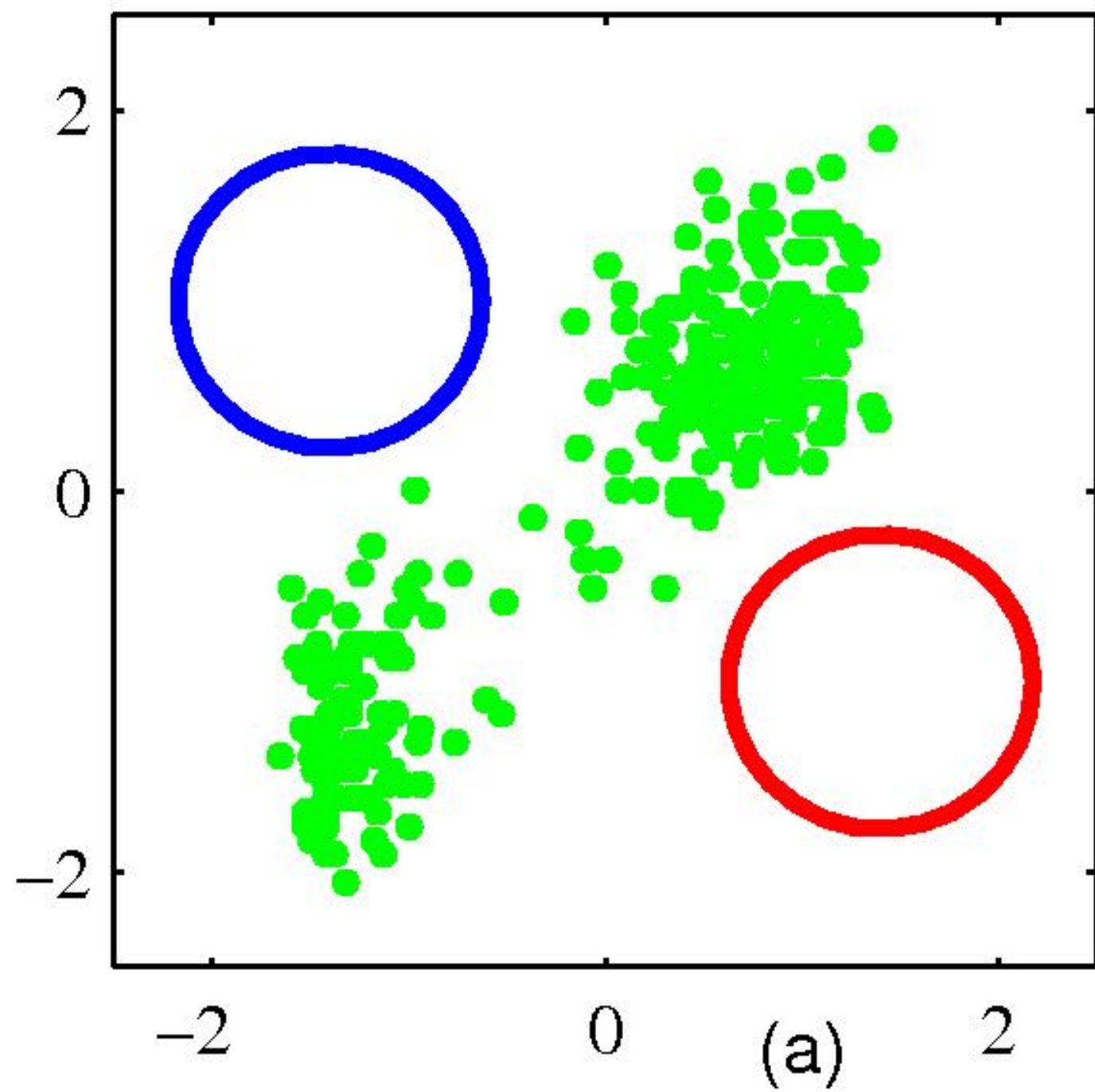
- Then the mixing coefficients (given from the average responsibilities

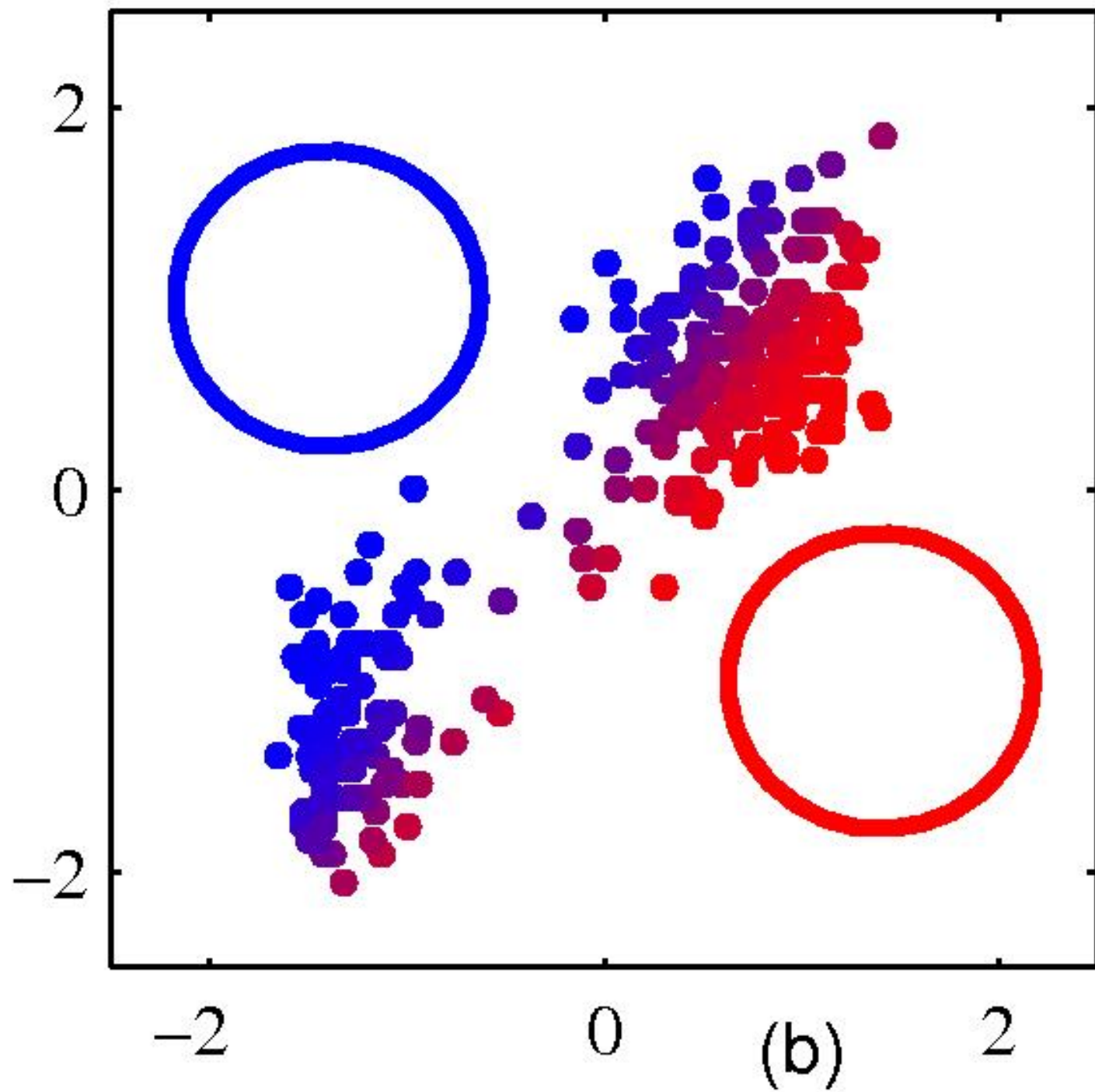
$$\pi_k = \frac{N_k}{N}$$

Algorithm

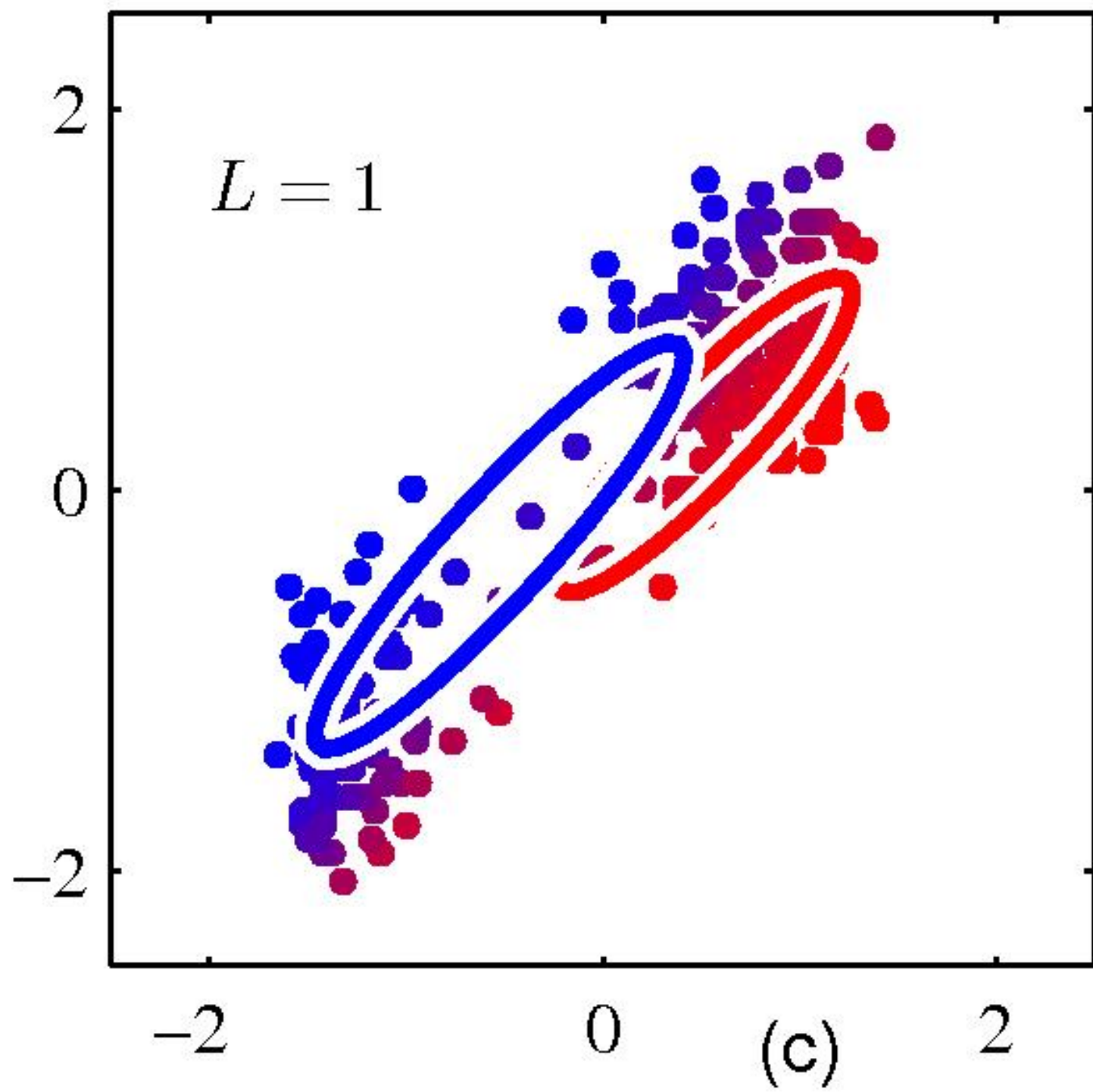
- Each update is (as with K-means) guaranteed to increase the log likelihood function.
- This is plotted through the run to evaluate convergence.

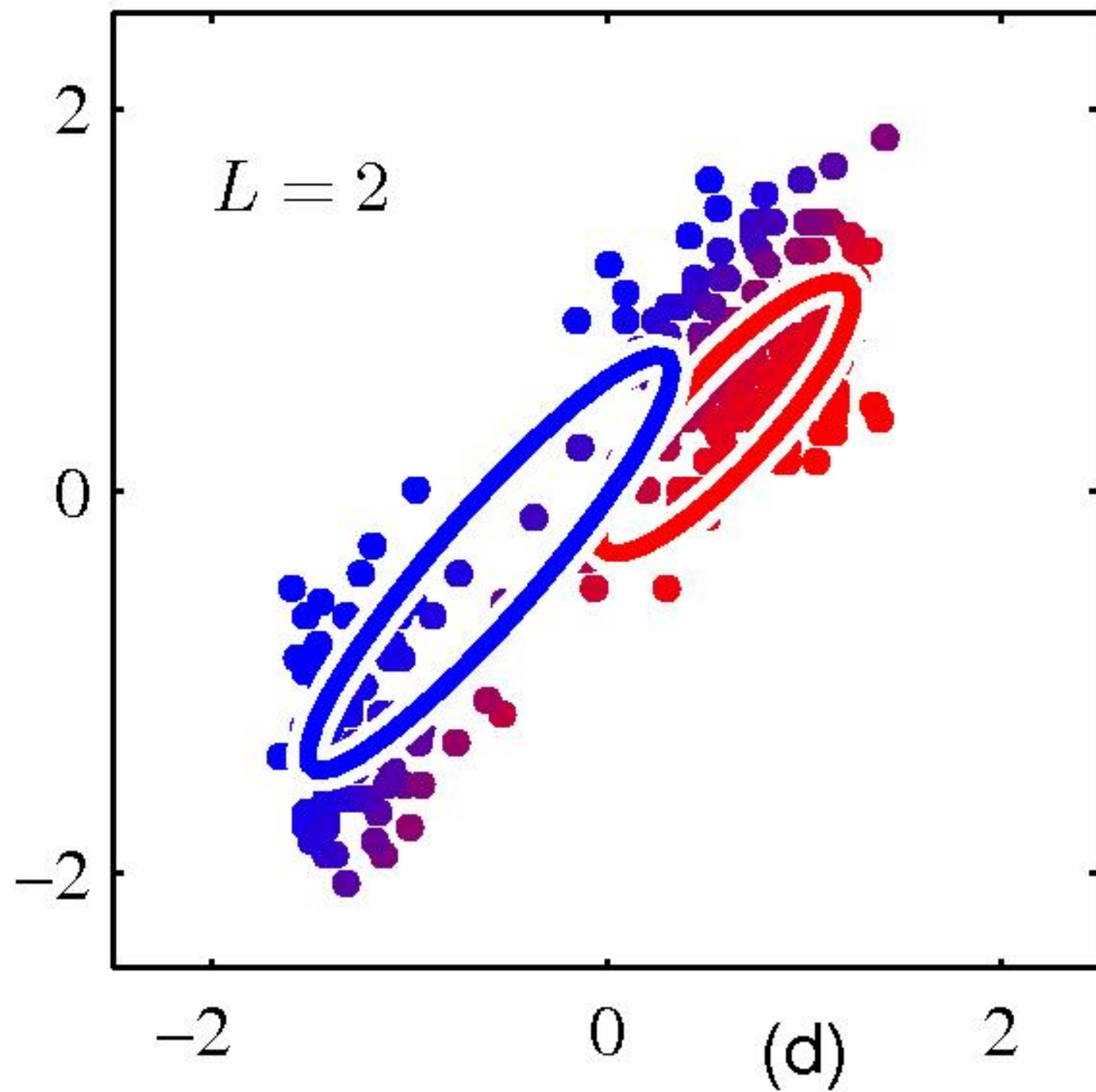
$$\ln p(X | u, \Sigma, \pi) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k N(x_n | u_k, \Sigma_k) \right\}$$

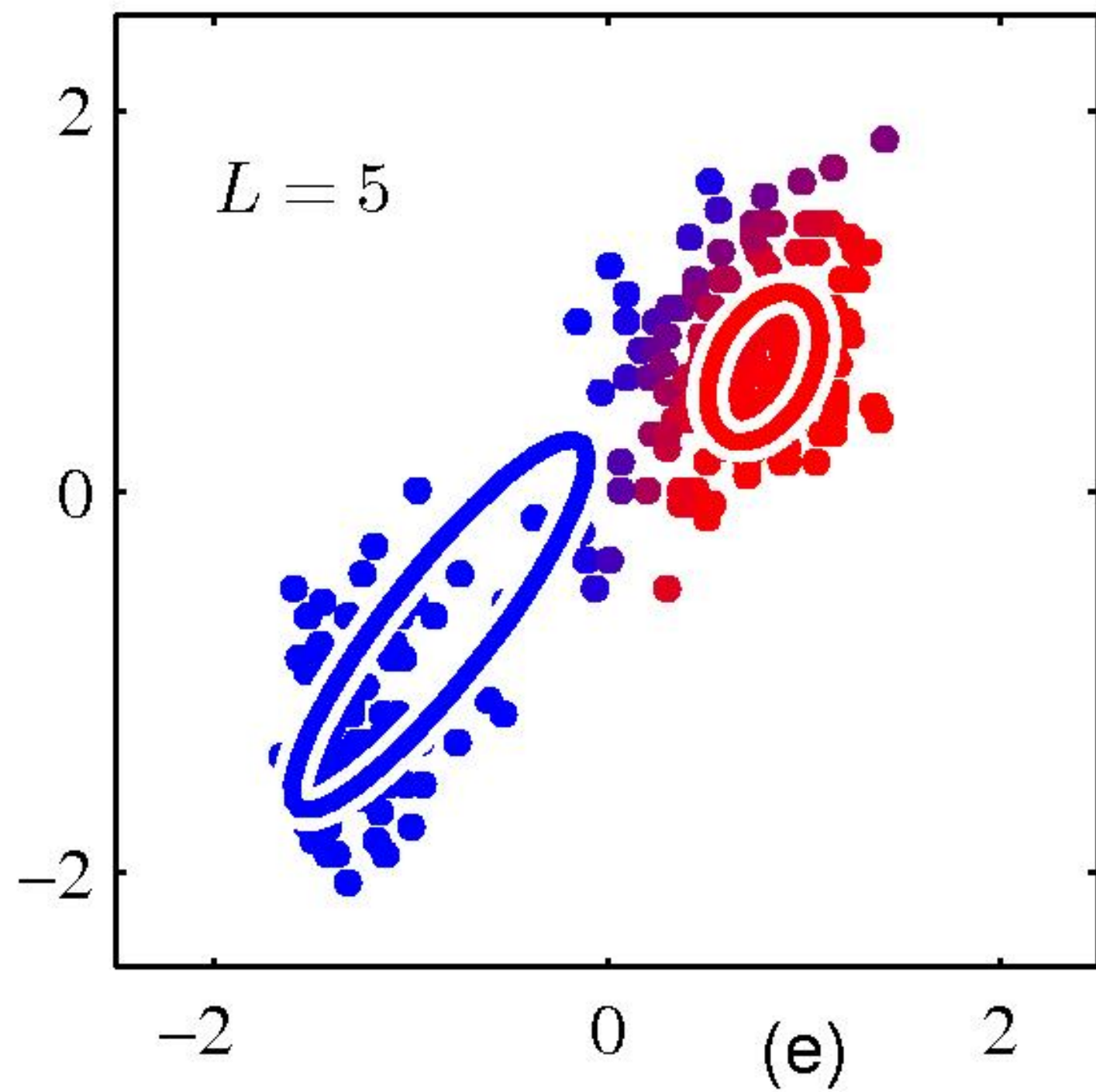


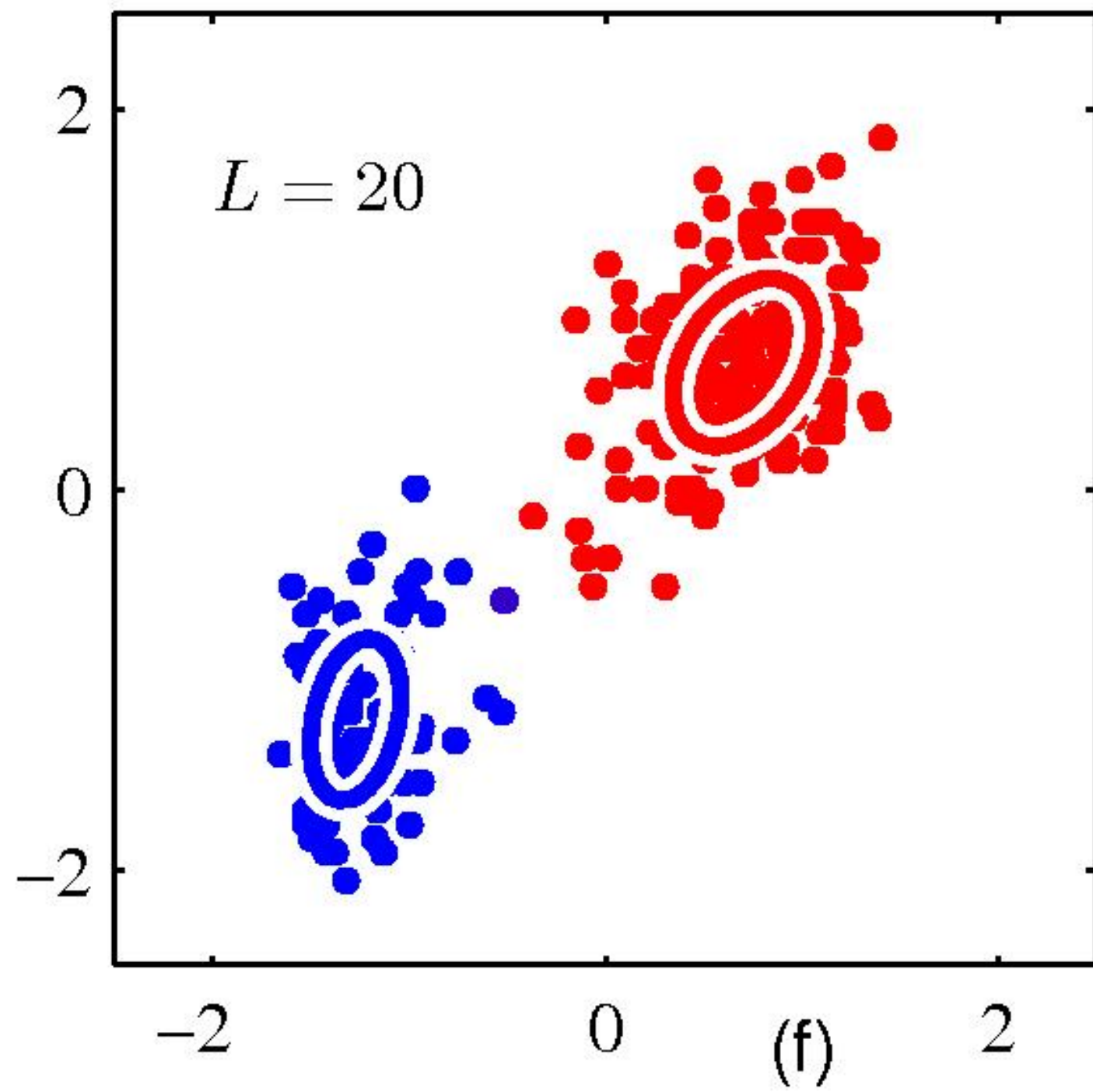


Initial E step
(assignment of
responsibility)









Relation to K-means

- EM takes longer to converge, so its normal to run K-means first to find centers of Gaussians.
- Set mixing coefficients to fractions of data points assigned to clusters by K-means.
- Set covariances to sample covarainces in clusters found by k-means.
- K-means forms hard assignment, EM makes soft assignment.

You need to **know and implement** K-means, but just remember EM exists.

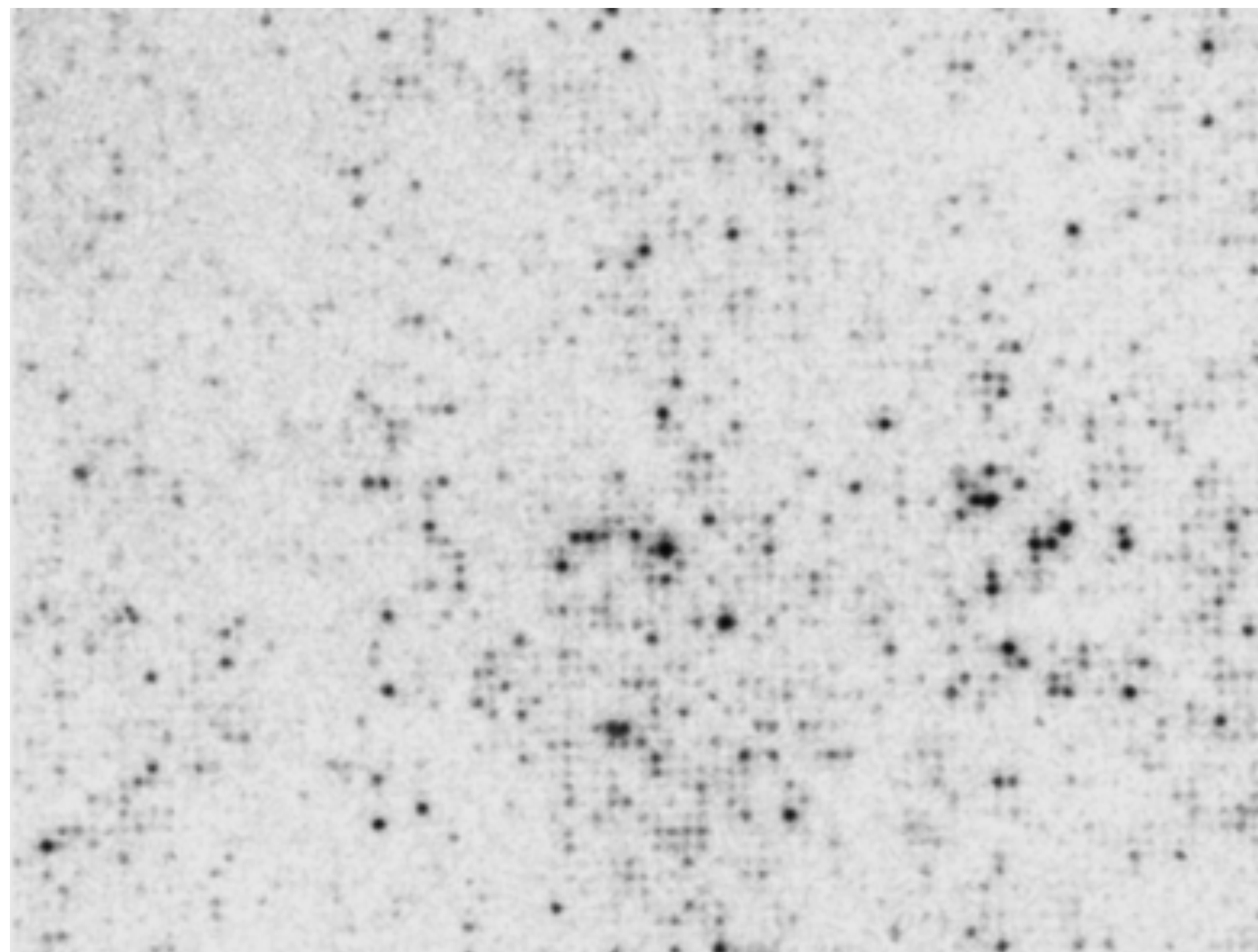
- Now lets apply this to computational genomics.
- Orengo, Jones, Thornton, Protein Bioinformatics BIOS 2003
 - Chapter 14 Biological background
 - Chapter 15 Computation

Clustering is useful in biology because...

- There is lots of data
- Want to group together 'things which are similar'
 - Easier to visualize
 - Easier to interpret the data and understand relationships between data

Microarray Analysis

- Microarrays measure the activity (expression level) of the gene under varying conditions/time points
- Expression level is estimated by measuring the amount of mRNA for that particular gene
 - A gene is active if it is being transcribed
 - More mRNA usually indicates more gene activity

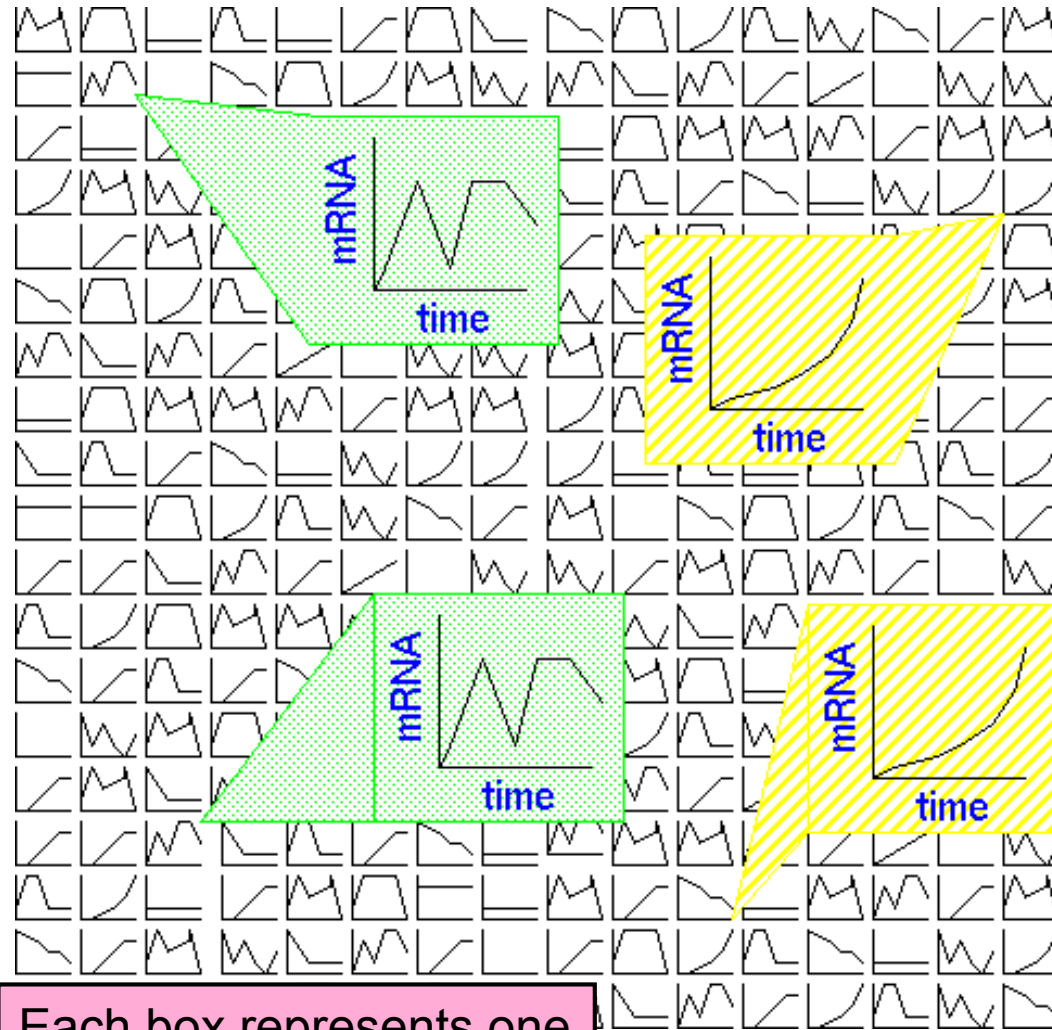


Using Microarrays (cont'd)

- **Green:** expressed only from control
- **Red:** expresses only from experimental cell
- **Yellow:** equally expressed in both samples
- **Black:** NOT expressed in either control or experimental cells



Using Microarrays



Track the sample over a period of time to see gene expression over time

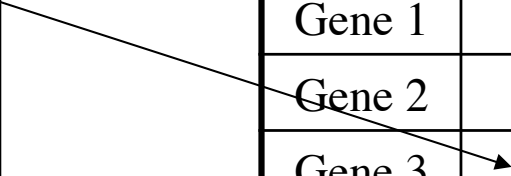
Track two different samples under the same conditions to see the difference in gene expressions

Each box represents one gene's expression over time

Microarray Data

- Microarray data are usually transformed into an **intensity matrix** (below)
- The intensity matrix allows biologists to make correlations between different genes (even if they are dissimilar) and to understand how genes functions might be related
- Clustering comes into play

Intensity (expression level)
of gene at measured time



Time:	Time X	Time Y	Time Z
Gene 1	10	8	10
Gene 2	10	0	9
Gene 3	4	8.6	3
Gene 4	7	8	3
Gene 5	1	2	3

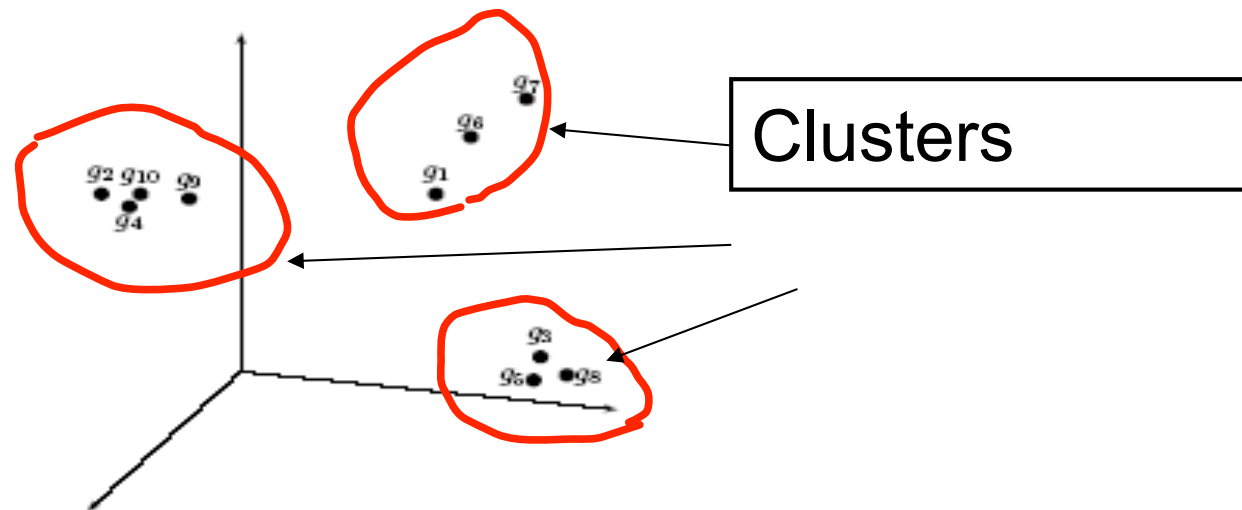
Clustering of Microarray Data (cont'd)

Time	1 hr	2 hr	3 hr
g_1	10.0	8.0	10.0
g_2	10.0	0.0	9.0
g_3	4.0	8.5	3.0
g_4	9.5	0.5	8.5
g_5	4.5	8.5	2.5
g_6	10.5	9.0	12.0
g_7	5.0	8.5	11.0
g_8	2.7	8.7	2.0
g_9	9.7	2.0	9.0
g_{10}	10.2	1.0	9.2

(a) Intensity matrix, I

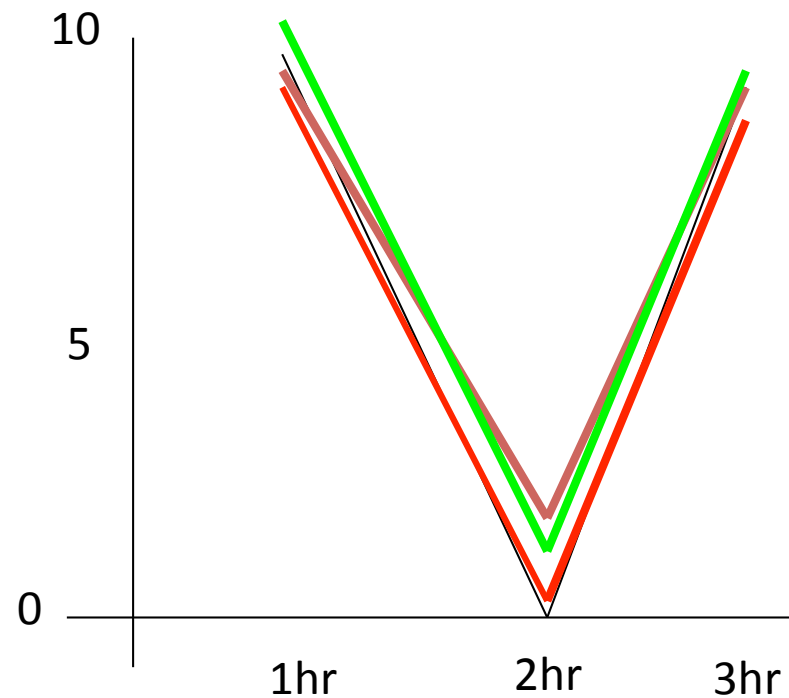
	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0

(b) Distance matrix, d



(c) Expression patterns as points in three-dimensional space.

Lets look at g2, g4, g9, g10



	1hr	2hr	3hr
g2	10	0	9.0
g4	9.5	0.5	8.5
g9	9.7	2.0	9.0
g10	10.2	1.0	9.2

Why cluster?

- Important for analysis of gene expression data; many other applications in bioinformatics (eg clustering protein sequence space to group together similar sequences)
- Clustering - organizing a set of unlabelled **patterns** (vectors) into groups based on similarity.
- Expect that patterns within a cluster are more similar to each other than to patterns in another cluster.

Why cluster

- Can identify groups of genes that may be up- or down- regulated in a given disease condition.
- Supervised learning methods (eg **support vector machines**) are also used in analysis of gene expression data.

Measures of dissimilarity (distance)

- Consider the levels of expression of 2 genes over p time points:

- These can be represented as \mathbf{x} and \mathbf{y} (these p dimensional vectors are called **expression patterns** in Orengo et al.)

$$- \mathbf{x} = [x_1, x_2, \dots, x_p]$$

$$- \mathbf{y} = [y_1, y_2, \dots, y_p]$$

$$d_E(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad d_M(x, y) = \sum_{i=1}^n |x_i - y_i|.$$

- Can use Euclidean or Manhattan distance to find out how close the patterns are.

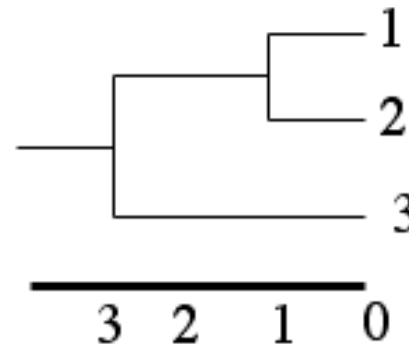
- Apply standardization.

$$x \mapsto \frac{x - \bar{x}}{\hat{\sigma}_x}$$

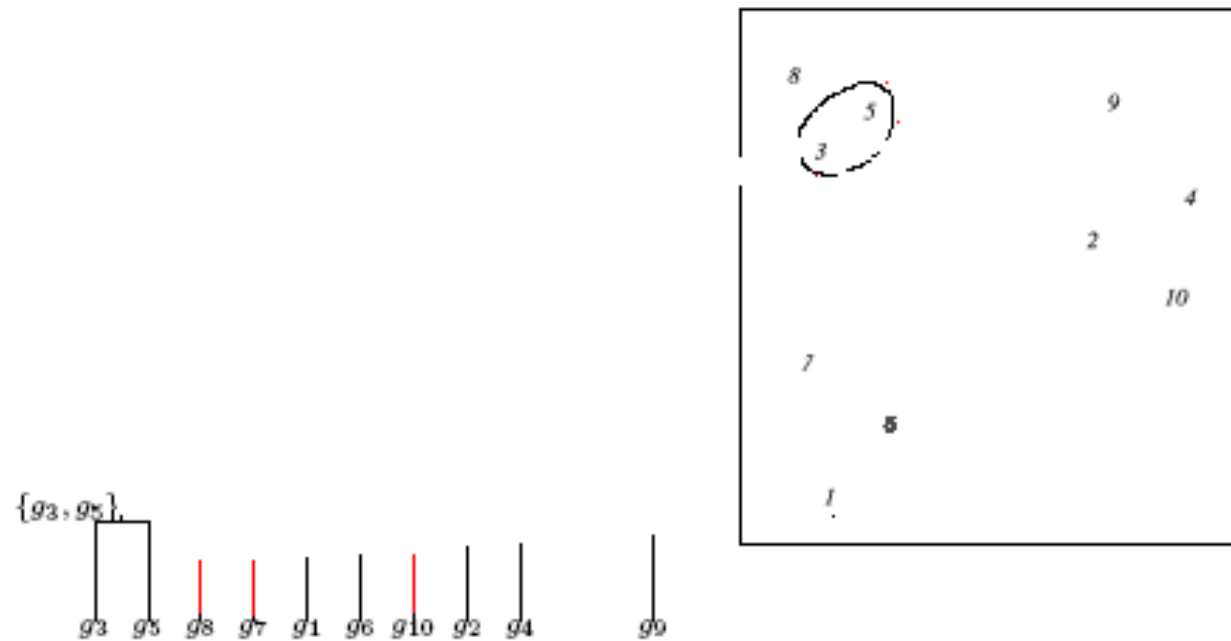
- This distance measure is used in the K-means algorithm for example...

Hierarchical clustering

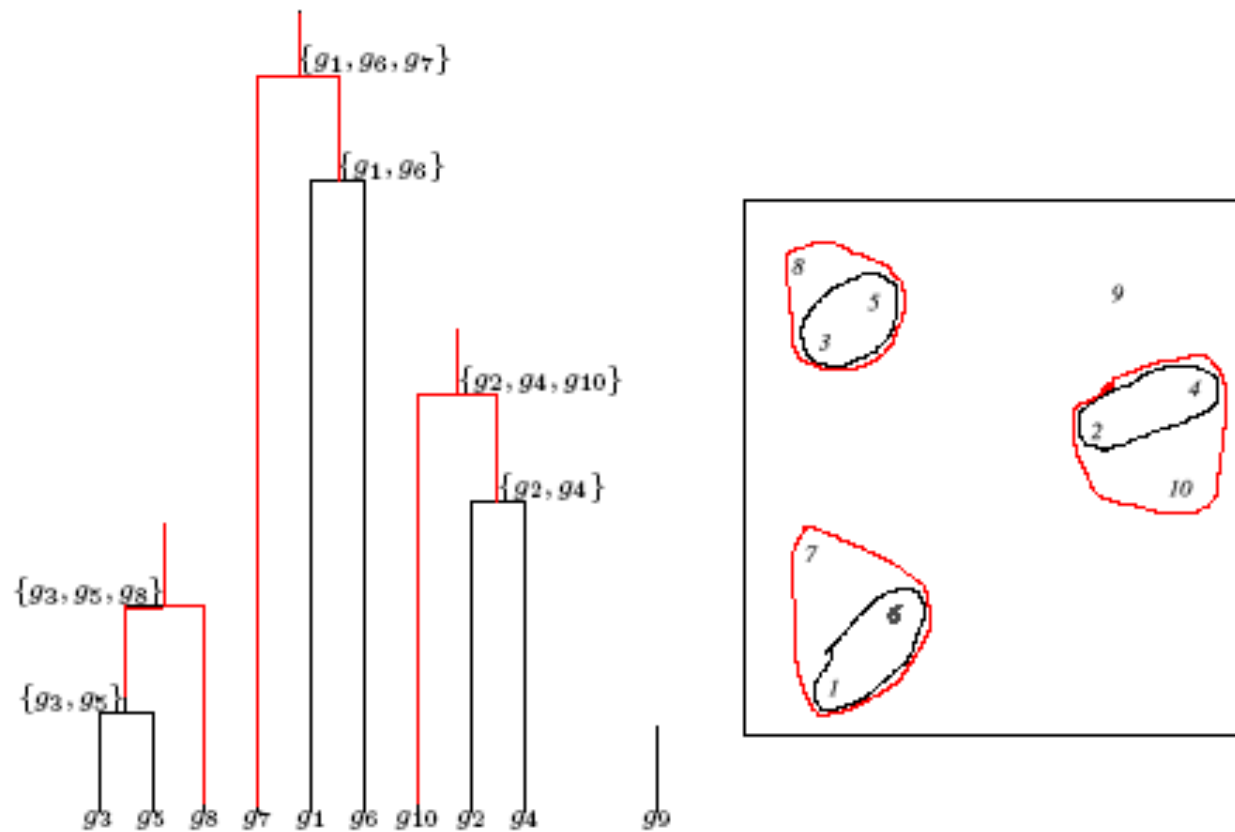
- K-means required i. no of clusters K, initial assignment of data to clusters, a distance measure between data points.
- Hierarchical clustering requires only **a measure of similarity between GROUPS of data points!**
- Produces a **tree** or dendrogram
- The distance between two groups is represented by the position where the branches join.



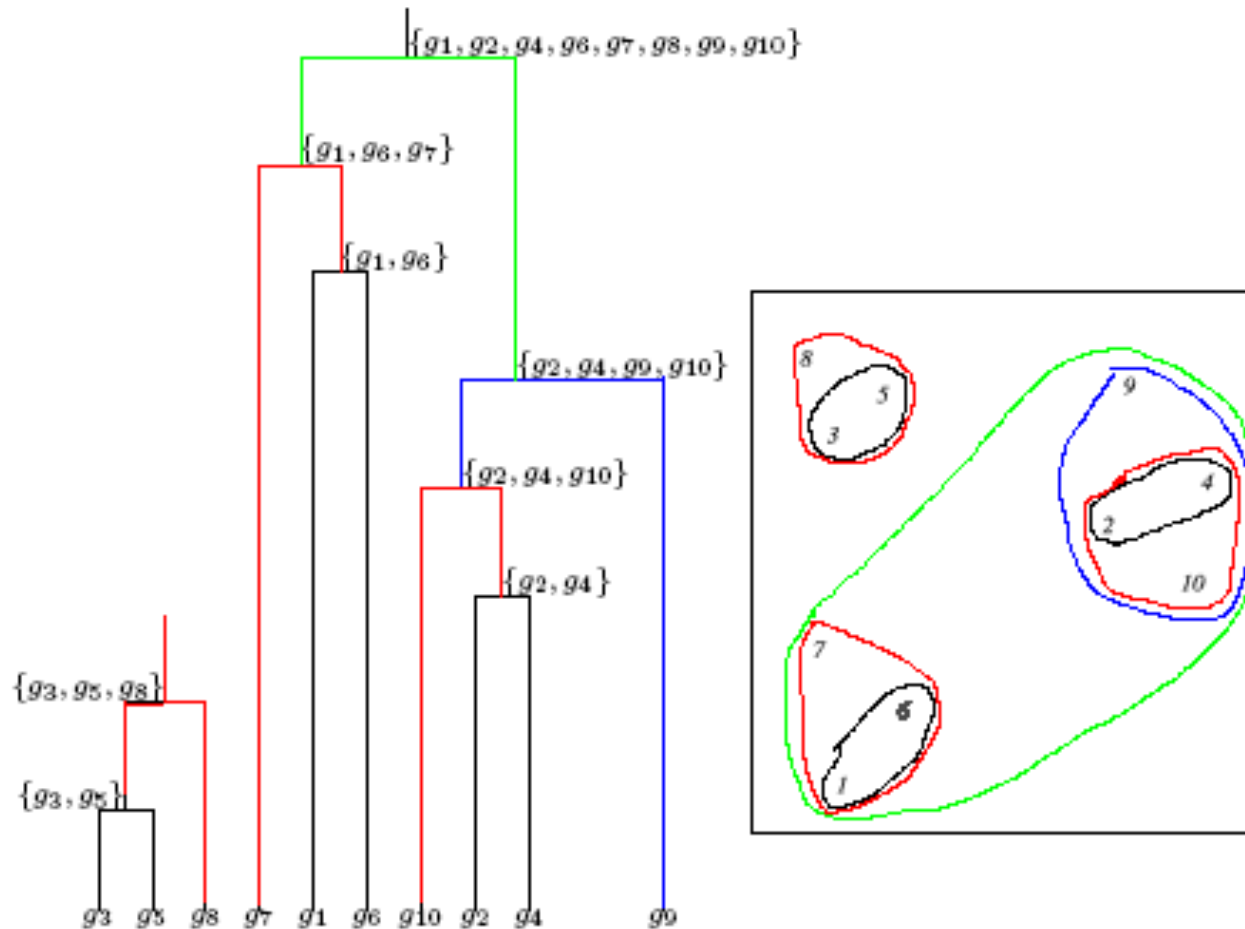
Hierarchical Clustering: Example



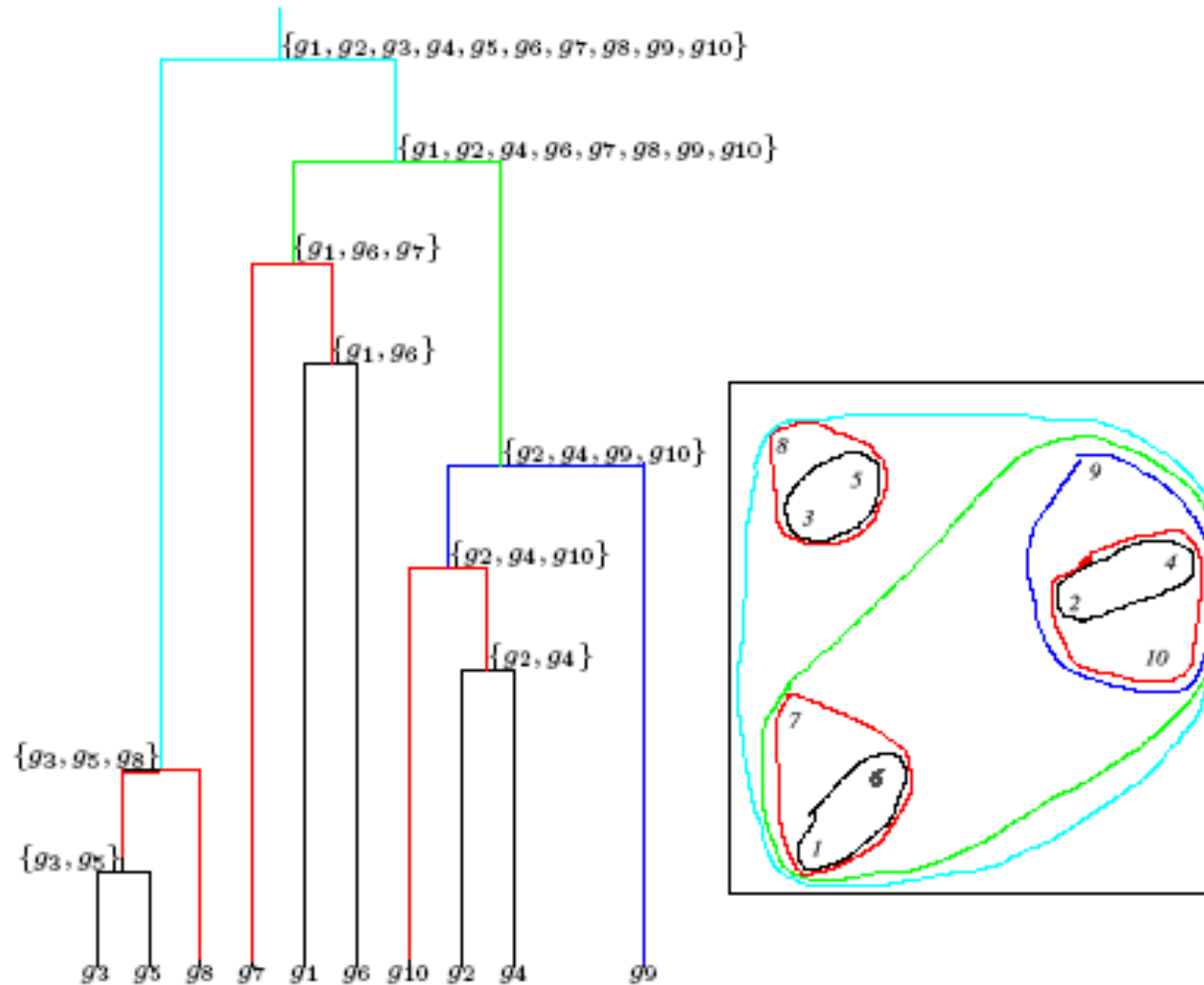
Hierarchical Clustering: Example



Hierarchical Clustering: Example

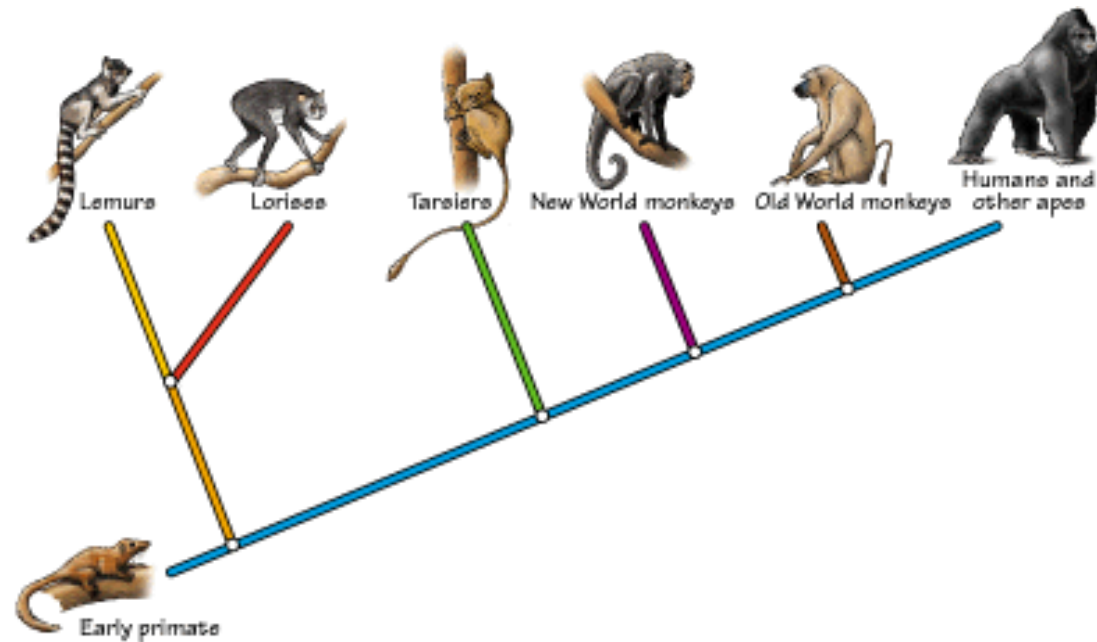


Hierarchical Clustering: Example



Hierarchical Clustering

- Hierarchical Clustering is often used to reveal evolutionary history



Hierarchical Clustering Algorithm

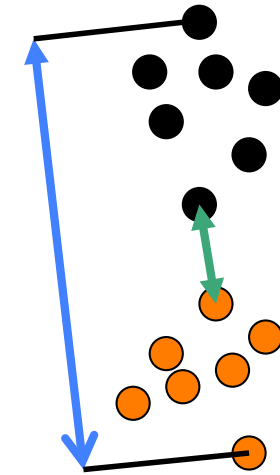
Hierarchical Clustering: The algorithm takes a $n \times n$ distance matrix d of pairwise distances between points as an input.

Form n clusters each with one element

- Construct a graph T by assigning one vertex to each cluster
- **while** there is more than one cluster
- Find the two closest clusters C_1 and C_2
- Merge C_1 and C_2 into new cluster C with $|C_1| + |C_2|$ elements
- **Compute distance from C to all other clusters**
- Add a new vertex C to T and connect to vertices C_1 and C_2
- Remove rows and columns of d corresponding to C_1 and C_2
- Add a row and column to d corresponding to the new cluster C
- return T

Different ways to define distances between clusters may lead to different clustering models

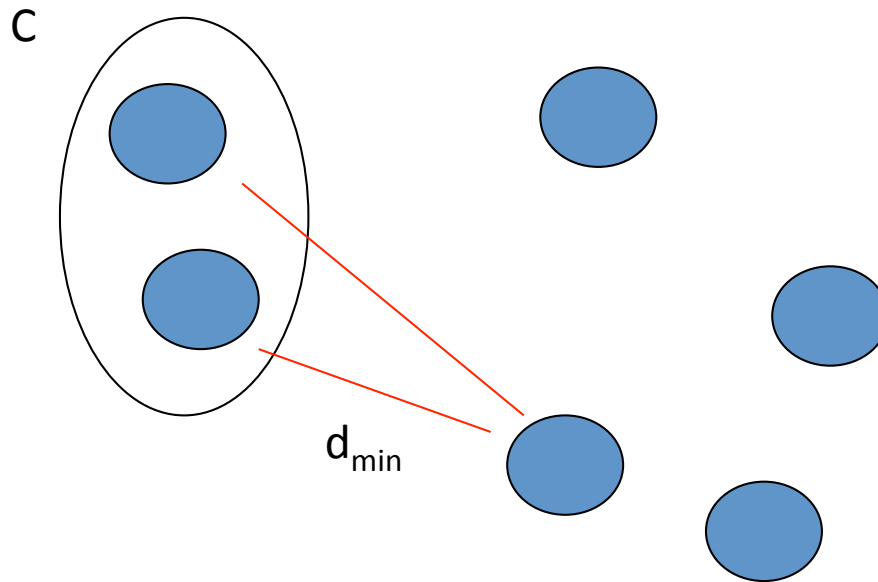
Single Linkage



- Consider 2 clusters A and B.
- Minimum distance between all pairs of patterns (one pattern from each cluster) - used by single linkage algorithm

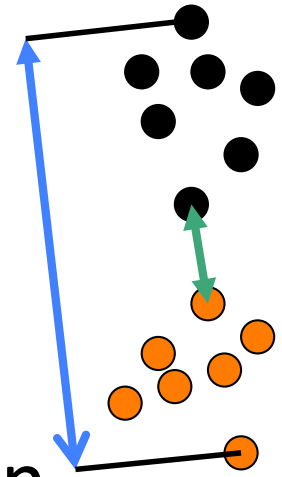
$$d_{AB} = \min_{i \in A; j \in B} d_{ij}$$

Single linkage - minimum distance



Decision to join a cluster is based on the minimum distance to the cluster

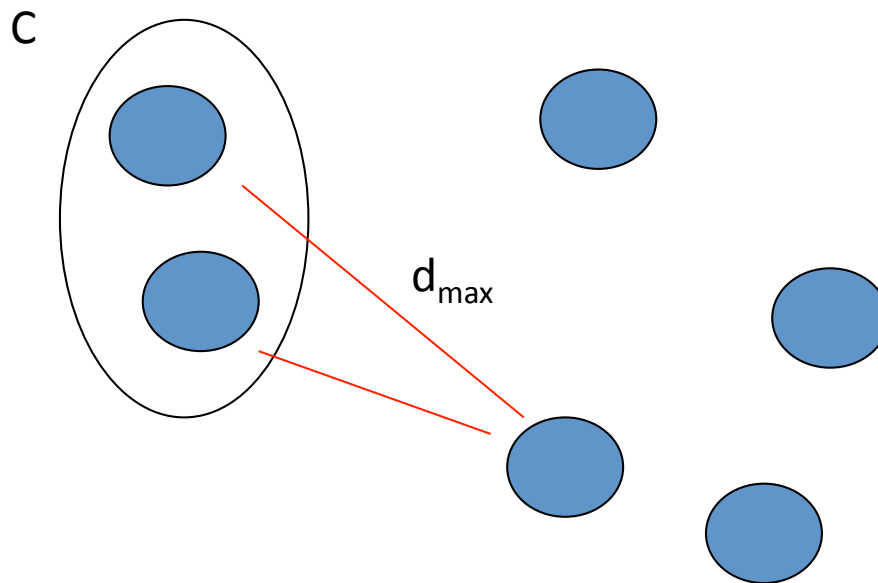
Complete Linkage



- Maximum distance between patterns in the two clusters (used by complete linkage algorithm).

$$d_{AB} = \max_{i \in A; j \in B} d_{ij}$$

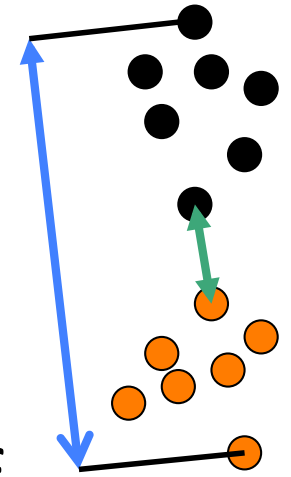
Complete linkage - maximum distance



Decision to join a cluster is based on the maximum distance to the cluster

Average Linkage

- Average of distances between all pairs of patterns with one pattern from each cluster - used by group average algorithm.



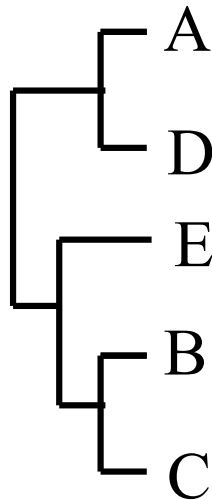
$$d_{AB} = \frac{1}{n_A n_B} \sum_{i \in A} \sum_{j \in B} d_{ij}$$

where n_A and n_B are the number of patterns in clusters A and B.

Hierarchical Clustering Summary

Advantages

- Easy to implement
- Very Visual



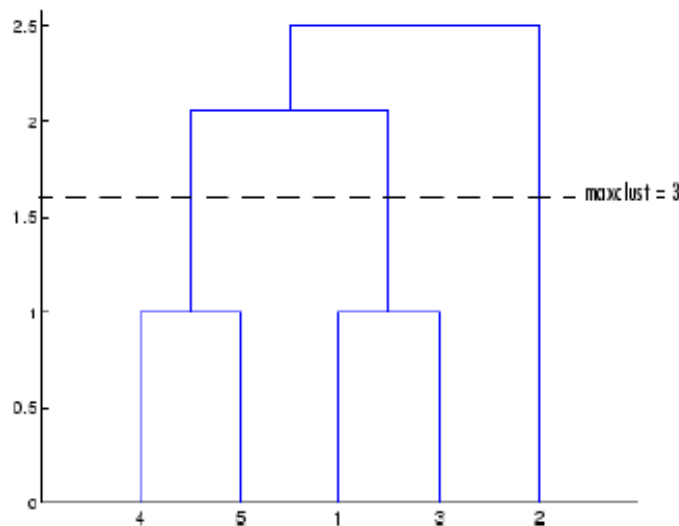
Disadvantages

- Unrelated Genes Are Eventually Joined
- Hard To Define Clusters: Where to Cut the Tree
- Manual Interpretation Often Required
- Cluster depends critically on distance measure.
- Imposes a tree structure on the data even though it might be the wrong model.

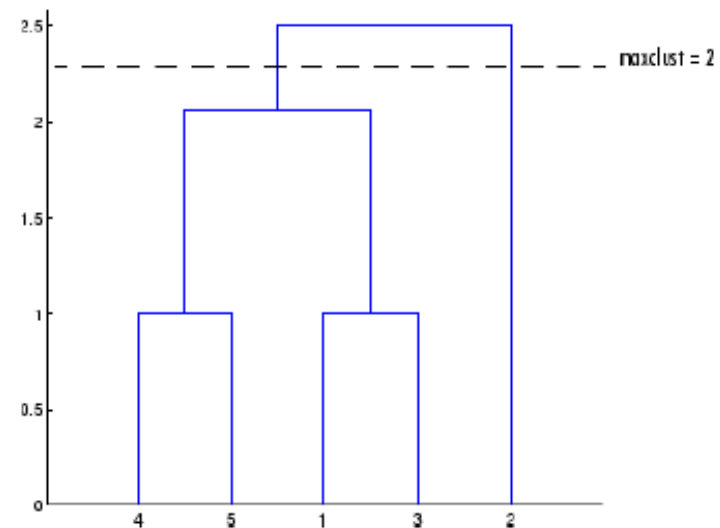
Hierarchical Clustering:

Where to cut the tree in order to determine (optimum) number of clusters and clustering models

Based on user's experience (arbitrary selection)



3-cluster model



2-cluster model

Hierarchical Clustering in MATLAB

To perform hierarchical cluster analysis on a data set using the Statistics Toolbox functions, follow this procedure:

- **Step - 1 Find the similarity or dissimilarity between every pair of objects in the data set:**
- In this step, you calculate the *distance* between objects using the **pdist** function. The **pdist** function supports many different ways to compute this measurement.
- **Step - 2 Group the objects into a binary, hierarchical cluster tree:**
- In this step, you link pairs of objects that are in close proximity using the **linkage** function, which is the main function to implement hierarchical clustering method. The **linkage** function uses the distance information generated in step 1 to determine the proximity of objects to each other. As objects are paired into binary clusters, the newly formed clusters are grouped into larger clusters until a hierarchical tree is formed.
- **Step - 3 Determine where to cut the hierarchical tree into clusters:**
- In this step, you use the cluster function to prune branches off the bottom of the hierarchical tree, and assign all the objects below each cut to a single cluster. This creates a partition of the data. The cluster function can create these clusters by detecting natural groupings in the hierarchical tree or by cutting off the hierarchical tree at an arbitrary point.
- The MATLAB's Statistics Toolbox includes a convenience function, **clusterdata**, which performs all these steps. No need to execute the **pdist**, **linkage**, or **cluster** functions separately. **For further details in the functions and implementation of the method, see the user manual.**

Example:

Given a data set X

$Y = \text{pdist}(X)$

$Z = \text{linkage}(Y)$

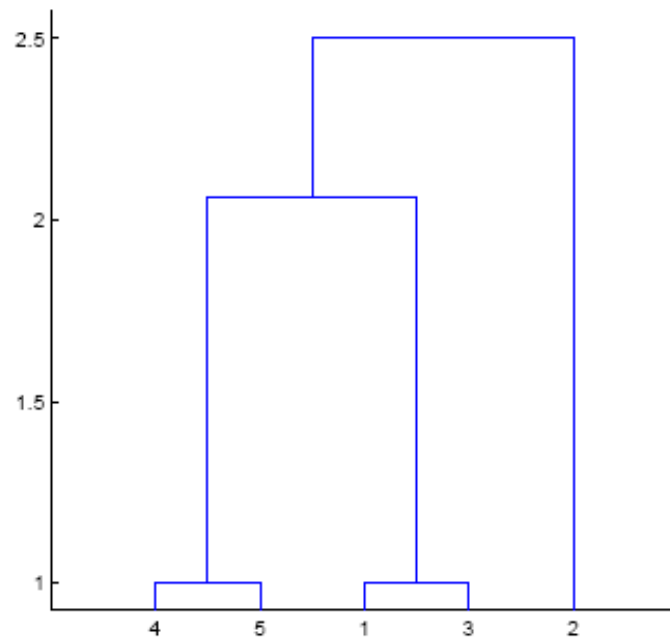
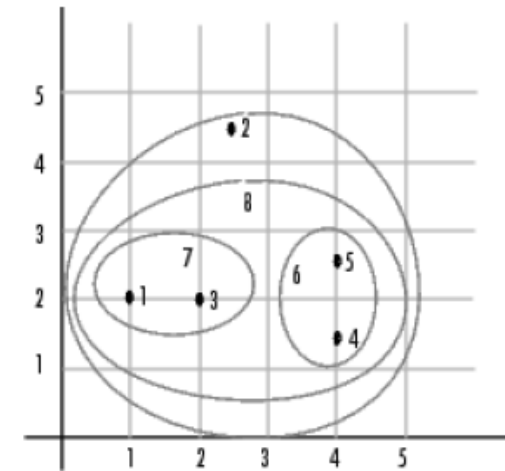
$Z = \text{linkage}(Y)$

$Z =$

4.0000	5.0000	1.0000
1.0000	3.0000	1.0000
6.0000	7.0000	2.0616
2.0000	8.0000	2.5000

Plotting the Cluster Tree

`dendrogram(Z)`



Using Matlab Bioinformatics Toolbox

- I strongly encourage you to PLAY with this stuff in Matlab or some other language...
- <http://www.mathworks.co.uk/help/toolbox/bioinfo/ug/a1060813239b1.html>