Clustering

Sriram Sankararaman
(Adapted from slides by Junming Yin)

Outline

- **Introduction**
  - **Unsupervised learning**
  - **What is cluster analysis?**
  - Applications of clustering
- Dissimilarity (similarity) of samples
- Clustering algorithms
  - K-means
  - Gaussian mixture model (GMM)
  - Hierarchical clustering
  - Spectral clustering

Unsupervised Learning

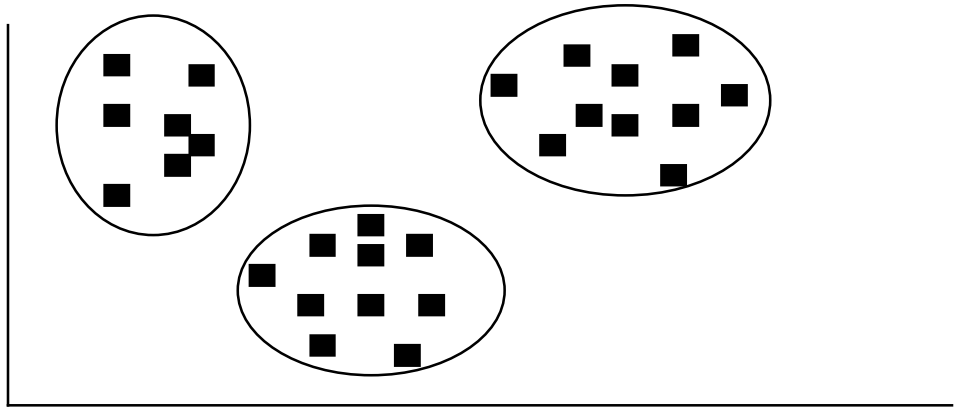
- Recall in the setting of classification and regression, the training data are represented as $(x_i, y_i)_{i=1\ldots n}$, the goal is to learn a function that predicts y given x. (supervised learning)
- In the unsupervised setting, we only have *unlabelled* data $(x_i)_{i=1\ldots n}$. Can we infer some properties of the distribution of X?

Why do Unsupervised Learning?

- Raw data is cheap but *labeling* them can be costly.
- The data lies in a high-dimensional space. We might find some low-dimensional *features* that might be sufficient to describe the samples (next lecture).
- In the early stages of an investigation, it may be valuable to perform *exploratory data analysis* and gain some insight into the nature or structure of data.
- Cluster analysis is one method for unsupervised learning.

What is Cluster Analysis?

- Cluster analysis aims to discover clusters or groups of samples such that samples within the same group are more similar to each other than they are to the samples of other groups.
  - A *dissimilarity (similarity)* function between samples.
  - A *loss function* to evaluate a groupings of samples into clusters.
  - An *algorithm* that optimizes this loss function.



Outline

- Introduction
  - Unsupervised learning
  - What is cluster analysis?
  - **Applications of clustering**
- Dissimilarity (similarity) of samples
- Clustering algorithms
  - K-means
  - Gaussian mixture model (GMM)
  - Hierarchical clustering
  - Spectral clustering

Image Segmentation

http://people.cs.uchicago.edu/~pff/segment/

Clustering Search Results

EisenLab
Commercial use of the ScanAlyze, **Cluster** and/or TreeView executable and/or ... **Cluster** and TreeView are an integrated pair of programs for analyzing and ...
rana.lbl.gov/EisenSoftware.htm - 11k - Cached - Similar pages

Book results for **cluster**
The Linux Enterprise Cluster : build a highly ... - by Karl Kopper - 466 pages
Messier's Nebulae and Star Clusters - by Kenneth Glyn Jones - 456 pages

Searches related to: **cluster**

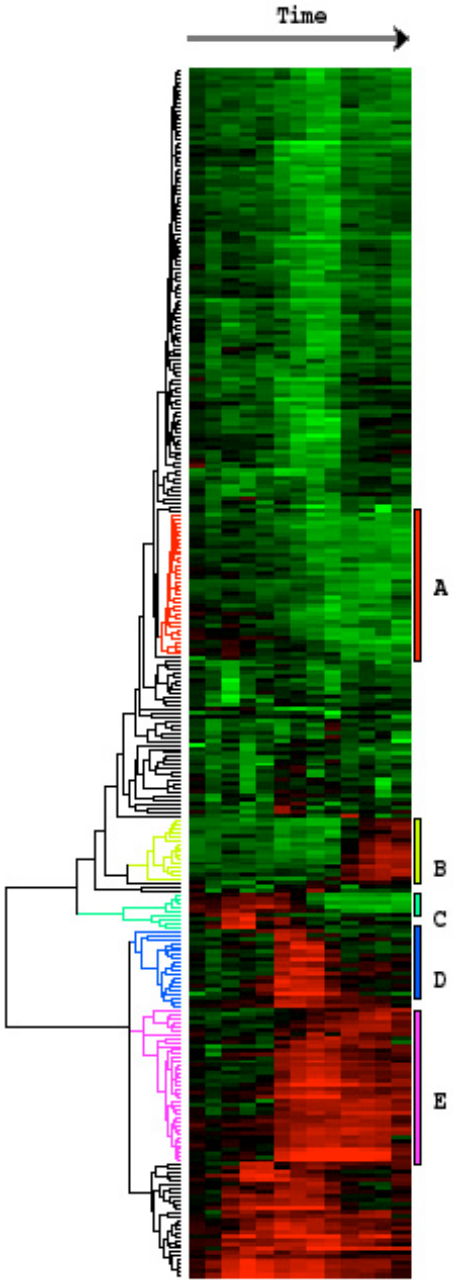
cluster **headache**	cluster **analysis**	**server** cluster	cluster **sampling**
cluster **windows 2003**	**sql** cluster	**oracle** cluster	**clusty**

Gooooooooooogle
1 2 3 4 5 6 7 8 9 10 Next

cluster Search

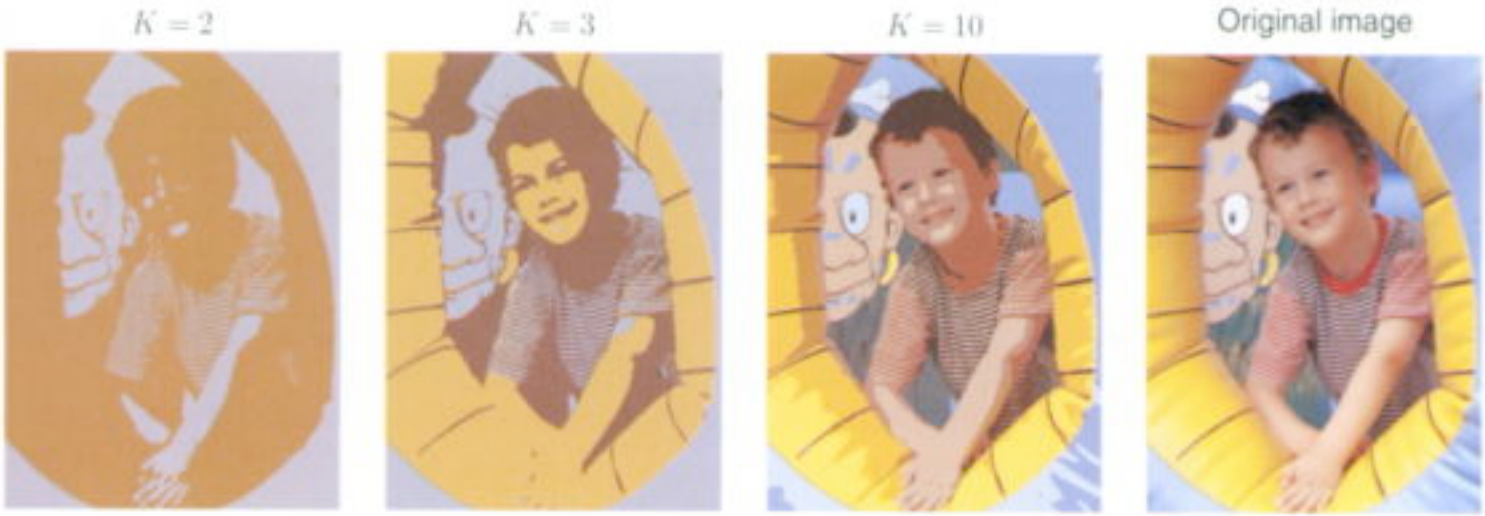
Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve |

Clustering gene expression data



Eisen et al, PNAS 1998

Vector quantization to compress images



Bishop, PRML

Outline

- Introduction
  - Unsupervised learning
  - What is cluster analysis?
  - Applications of clustering
- **Dissimilarity (similarity) of samples**
- Clustering algorithms
  - K-means
  - Gaussian mixture model (GMM)
  - Hierarchical clustering
  - Spectral clustering

Dissimilarity of samples

- The natural question now is: how should we measure the dissimilarity between samples?
  - The clustering results depend on the choice of dissimilarity.
  - Usually from *subject* matter consideration.
  - Need to consider the type of the features.
    - Quantitative, ordinal, categorical.
  - Possible to learn the dissimilarity from data for a particular application (later).

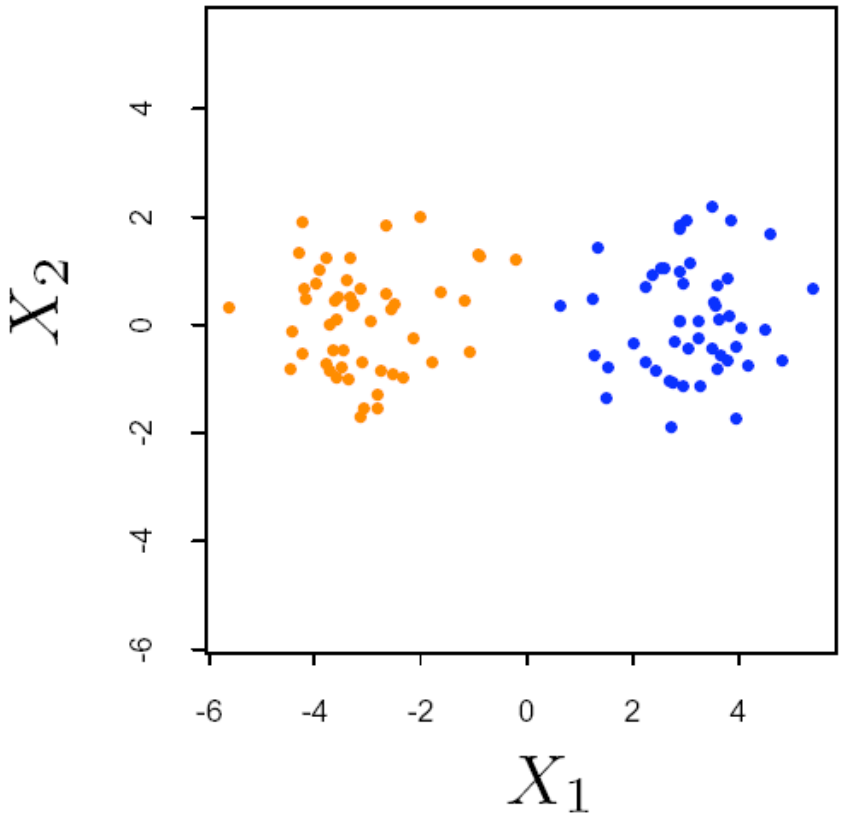
Dissimilarity Based on features

- Most of time, data x_i have measurements x_{ij} on features $j = 1, \ldots, p$.
- A common choice of dissimilarity function between samples is the Euclidean distance.

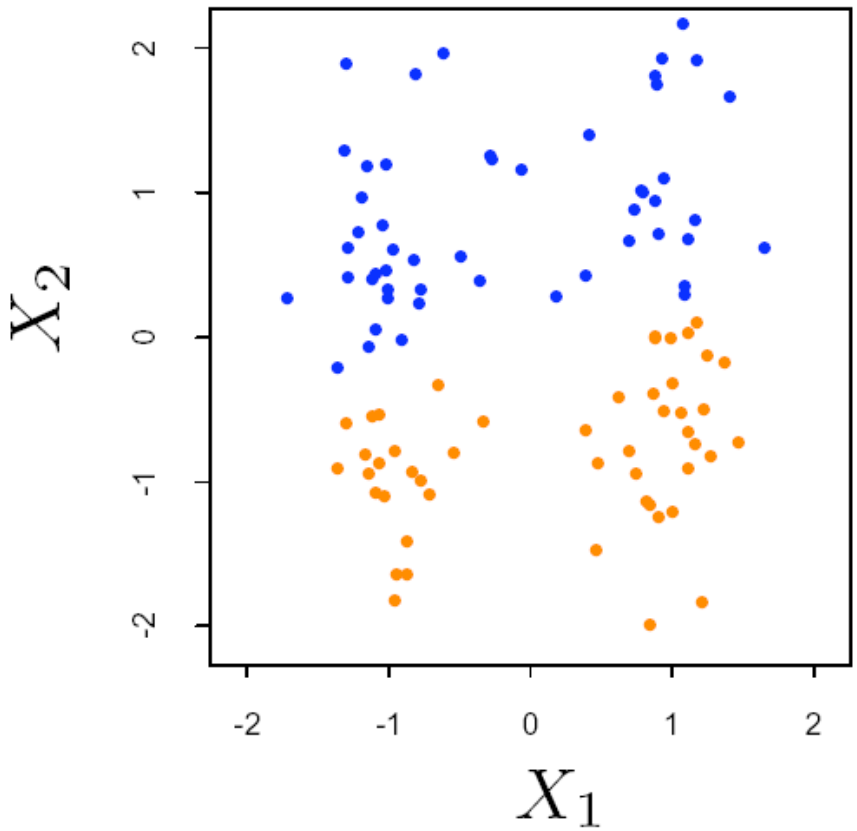
$$D(x_i, x_{i'}) = ||x_i - x_{i'}|| = \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2}$$

- Clusters defined by Euclidean distance is invariant to translations and rotations in feature space, but not invariant to scaling of features.
- One way to standardize the data: translate and scale the features so that all of features have zero mean and unit variance.
- BE CAREFUL! It is not always desirable.

Standardization not always helpful



Simulated data, 2-means without standardization



Simulated data, 2-means with standardization

Outline

- Introduction
  - Unsupervised learning
  - What is cluster analysis?
  - Applications of clustering
- Dissimilarity (similarity) of samples
- Clustering algorithms
  - **K-means**
  - Gaussian mixture model (GMM)
  - Hierarchical clustering
  - Spectral clustering

K-means: Idea

- Represent the data set in terms of K clusters, each of which is summarized by a prototype μ_k
- Each data is assigned to one of K clusters
  - Represented by responsibilities $r_{ik} \in \{0, 1\}$ such that $\sum_{k=1}^{K} r_{ik} = 1$ for all data indices *i*
- Example: 4 data points and 3 clusters

$$(r_{ik}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

K-means: Idea

- Loss function: the sum-of-squared distances from each data point to its assigned prototype (is equivalent to the within-cluster scatter).

data

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|x_i - \mu_k\|^2$$

responsibilities

prototypes

Minimizing the loss Function

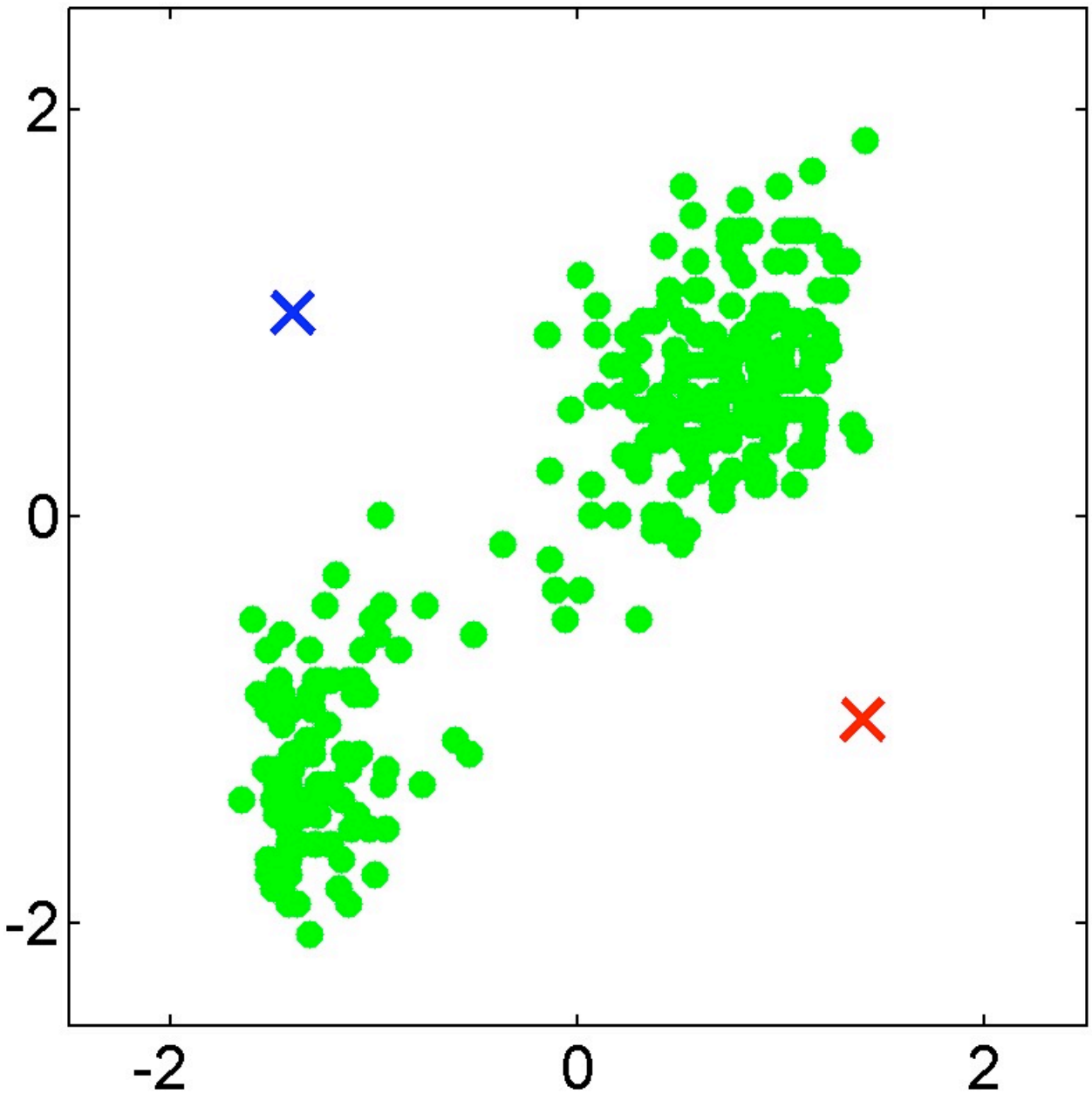
- Chicken and egg problem
  - If prototypes known, can assign responsibilities.
  - If responsibilities known, can compute optimal prototypes.
- We minimize the loss function by an iterative procedure.
- Other ways to minimize the loss function include a merge-split approach.

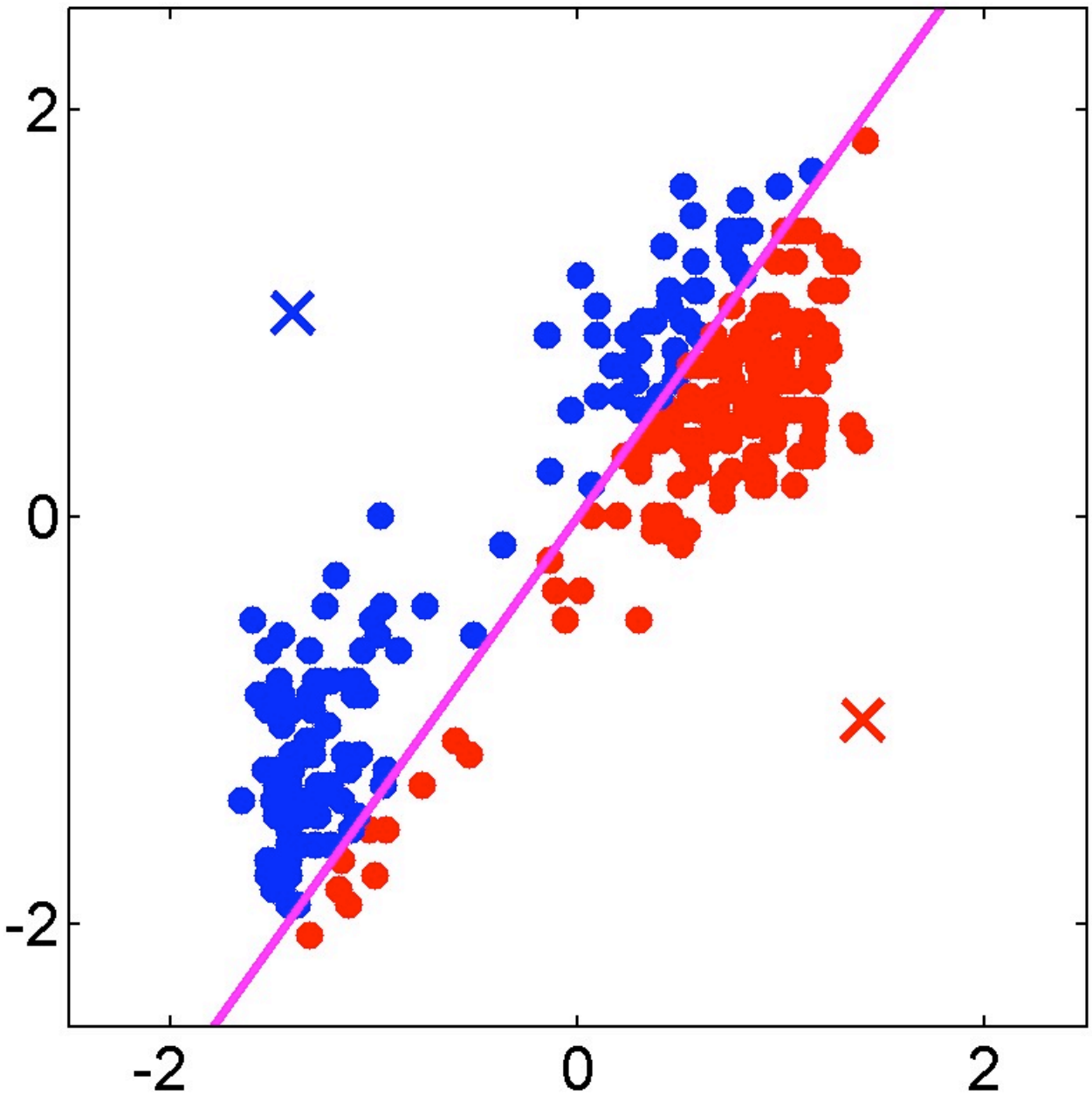
Minimizing the loss Function

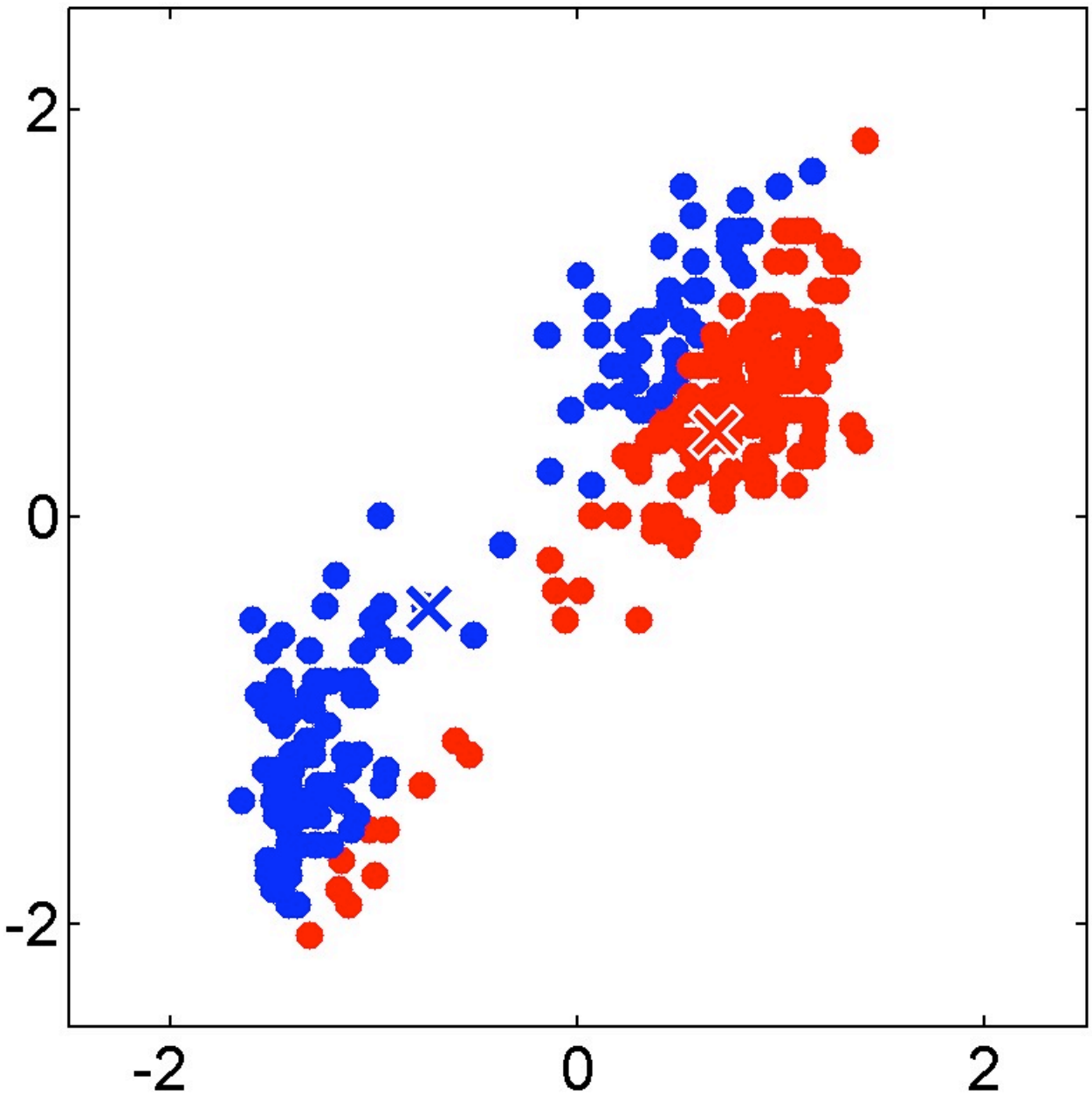
- E-step: Fix values for μ_k and minimize J w.r.t. r_{ik}
  - Assign each data point to its nearest prototype
- M-step: Fix values for r_{ik} and minimize J w.r.t μ_k
  - This gives

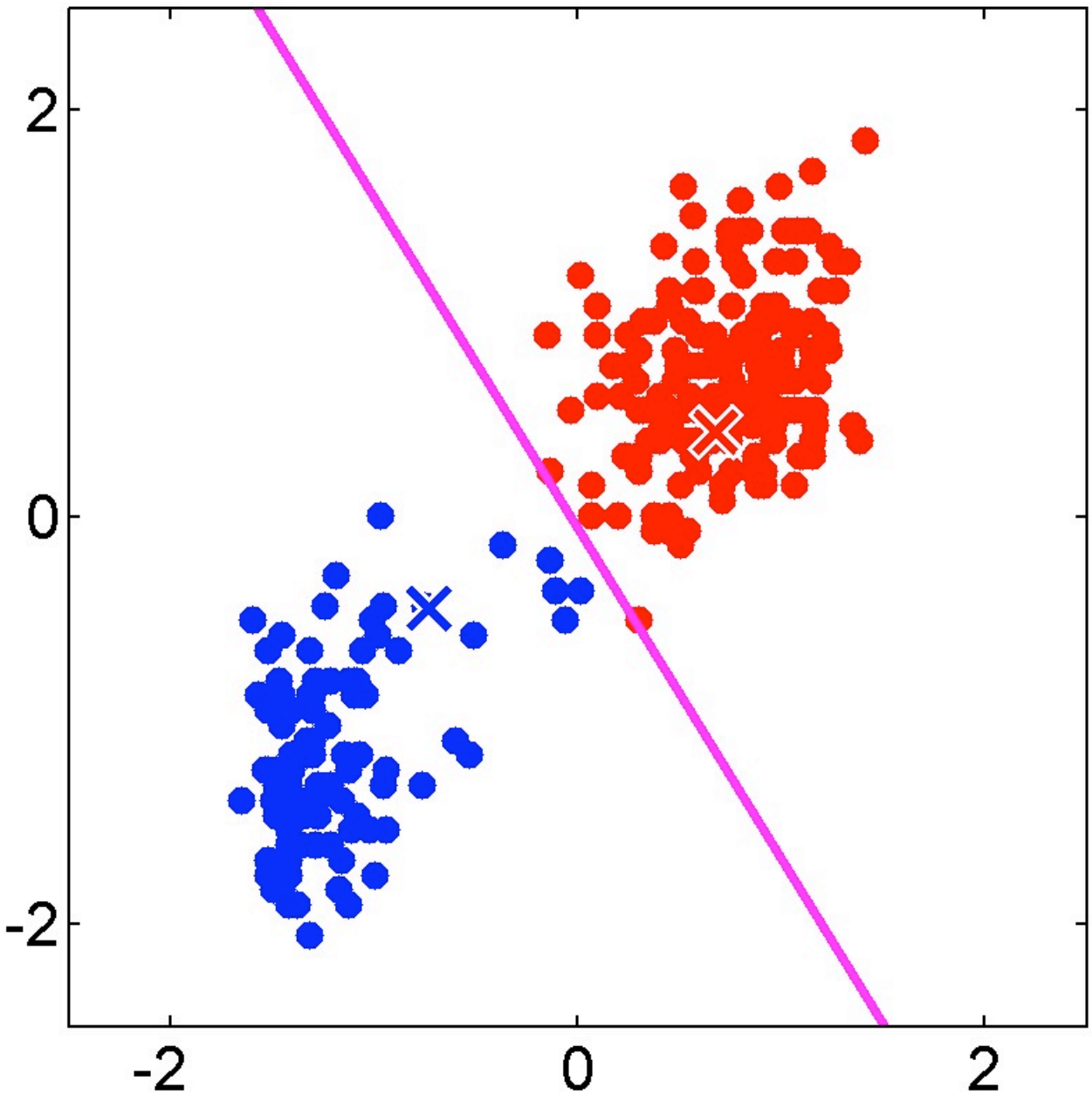
$$\mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}$$

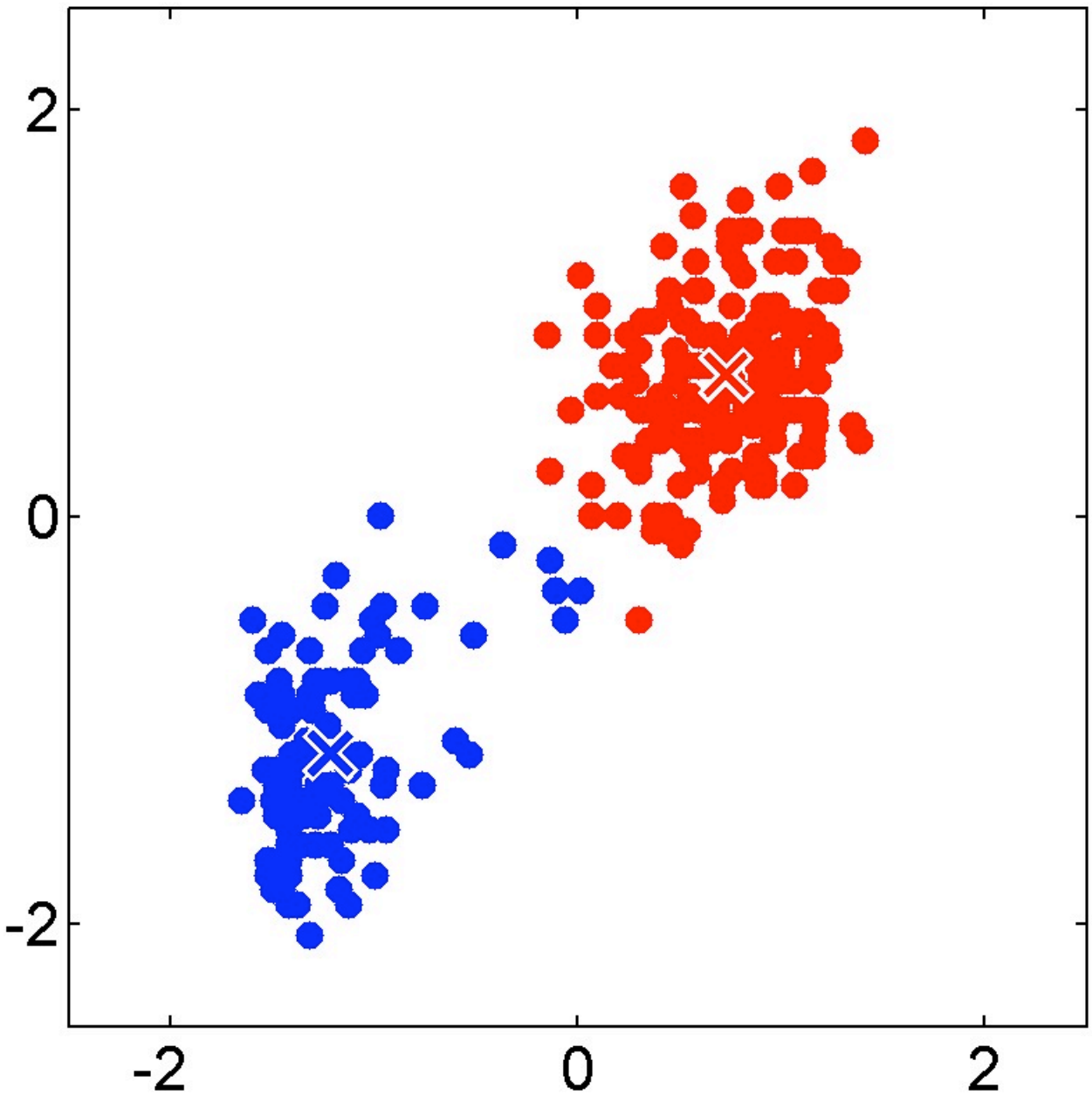
  - Each prototype set to the mean of points in that cluster.
- Convergence guaranteed since there are a finite number of possible settings for the responsibilities.
- It can only find the local minima, we should start the algorithm with many different initial settings.

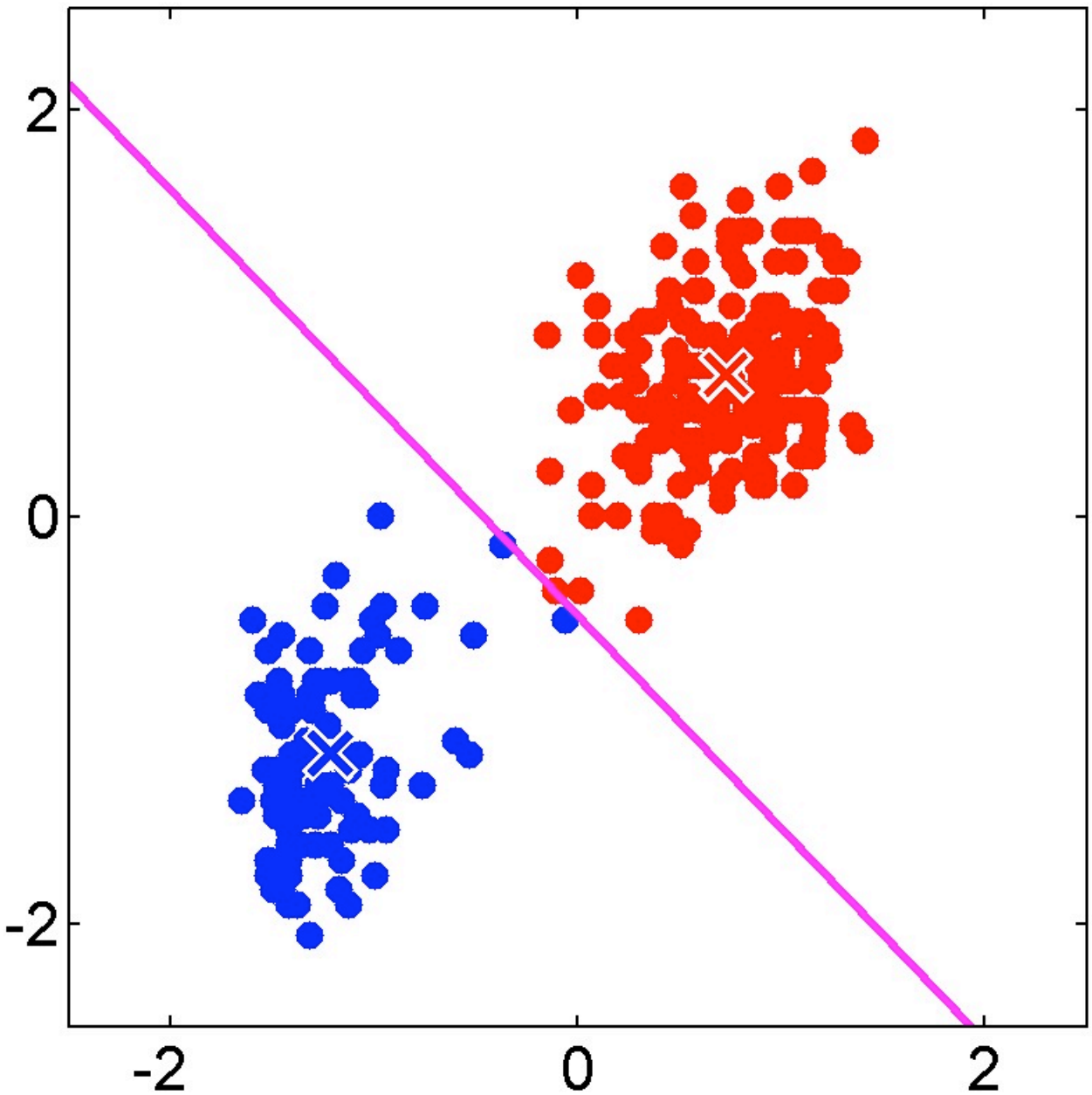

2
0
-2
-2
0
2

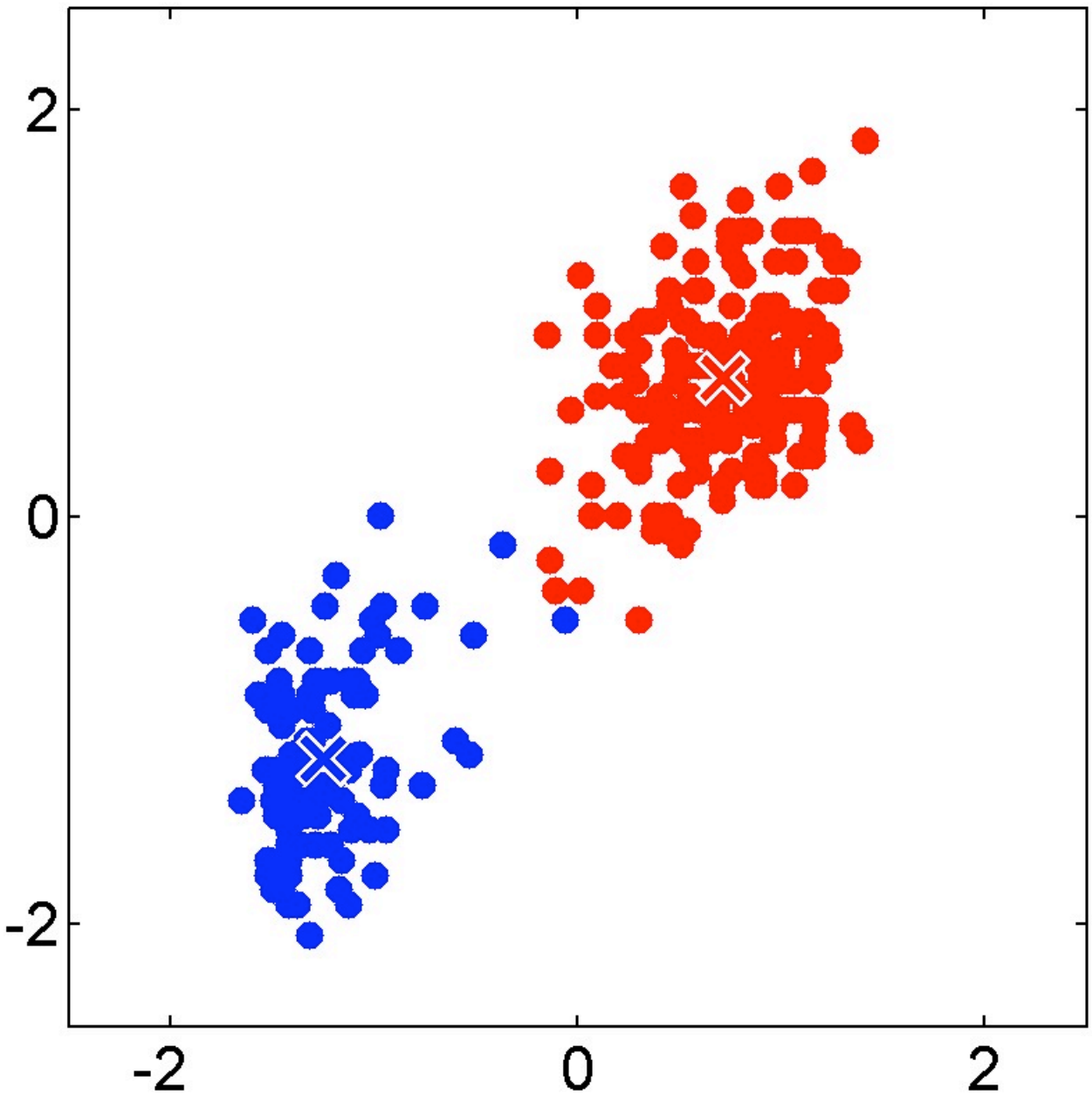

2
0
-2
-2
0
2

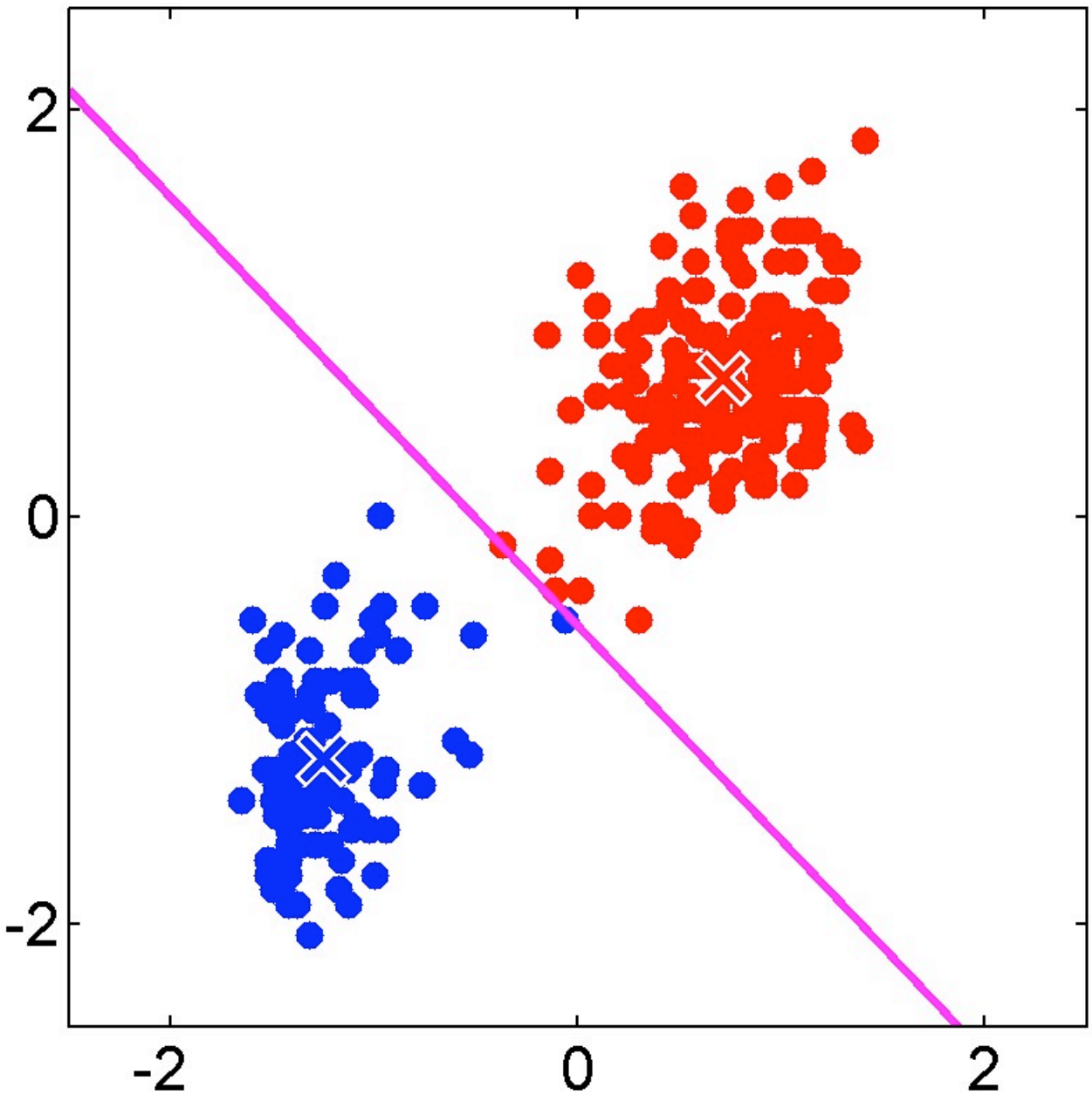

2
0
-2
-2
0
2

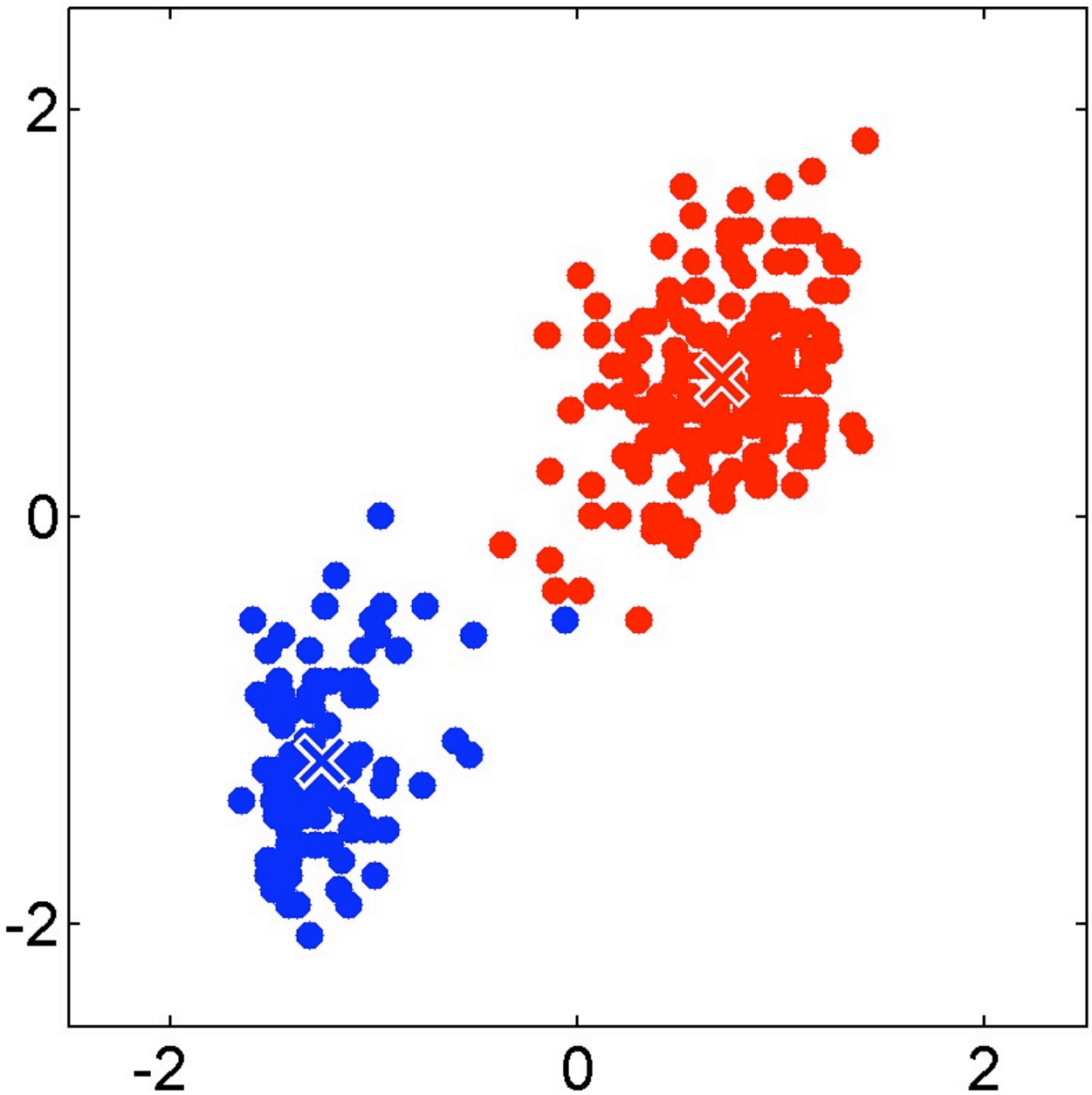

2
0
-2
-2
0
2


2
0
-2
-2
0
2

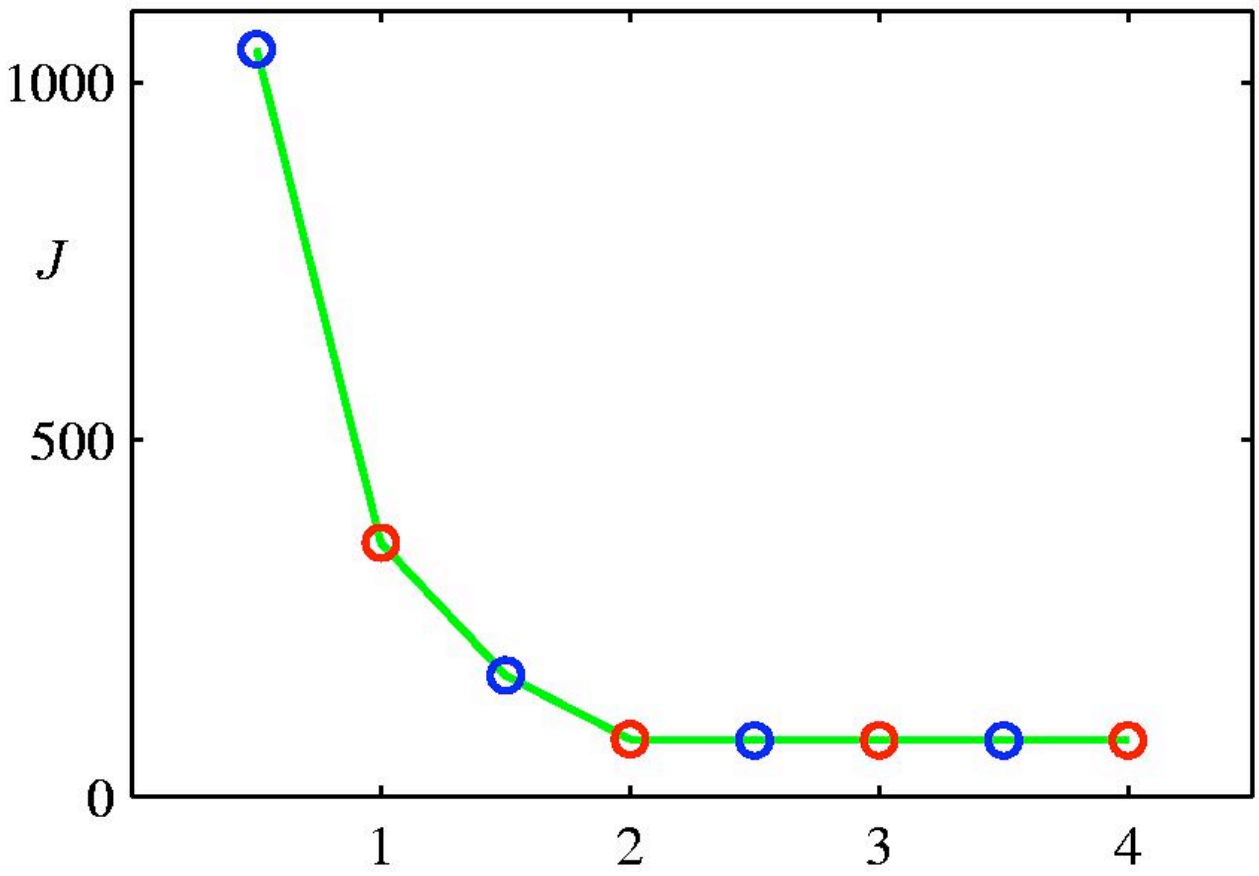

2
0
-2
-2
0
2


2
0
-2
-2
0
2


2
0
-2
-2
0
2


2
0
-2
-2
0
2

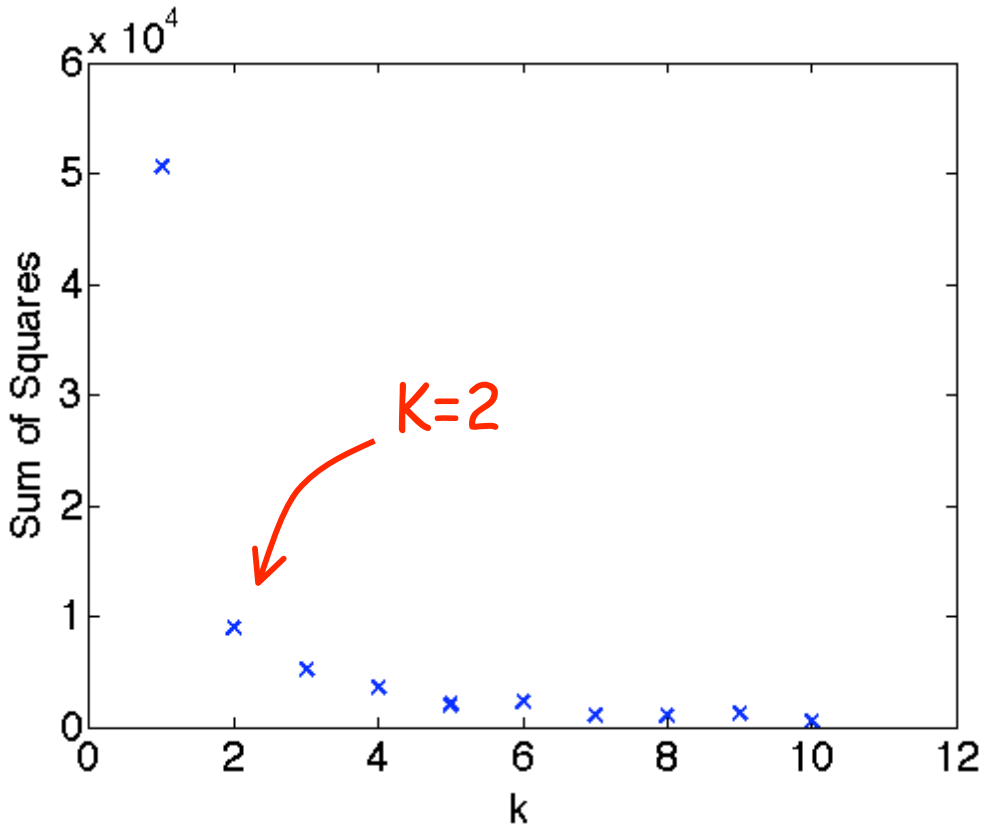
The Cost Function after each E and M step



How to Choose *K*?

- In some cases it is known apriori from problem domain.
- Generally, it has to be estimated from data and usually selected by some heuristics in practice.
  - Recall the choice of parameter *K* in nearest-neighbor.
- The loss function *J* generally decrease with increasing *K*
- Idea: Assume that K^* is the right number
  - We assume that for $K<K^*$ each estimated cluster contains a subset of true underlying groups
  - For $K>K^*$ some natural groups must be split
  - Thus we assume that for $K<K^*$ the cost function falls substantially, afterwards not a lot more

How to Choose *K*?



- The Gap statistic provides a more principled way of setting K.

Initializing K-means

- K-means converge to a local optimum.
- Clusters produced will depend on the initialization.
- Some heuristics
  - Randomly pick *K* points as prototypes.
  - A greedy strategy. Pick prototype $i+1$ so that it is farthest from prototypes $\{1, \ldots, i\}$

Limitations of K-means

- Hard assignments of data points to clusters
  - Small shift of a data point can flip it to a different cluster
  - Solution: replace hard clustering of K-means with soft probabilistic assignments (GMM)
- Assumes spherical clusters and equal probabilities for each cluster.
  - Solution: GMM
- Clusters arbitrary with different values of *K*
  - As K is increased, cluster memberships change in an arbitrary way, the clusters are not necessarily nested
  - Solution: hierarchical clustering
- Sensitive to outliers.
  - Solution: use a different loss function.
- Works poorly on non-convex clusters.
  - Solution: spectral clustering

Outline

- Introduction
  - Unsupervised learning
  - What is cluster analysis?
  - Applications of clustering
- Dissimilarity (similarity) of samples
- Clustering algorithms
  - K-means
  - **Gaussian mixture model (GMM)**
  - Hierarchical clustering
  - Spectral clustering

The Gaussian Distribution

- Multivariate Gaussian

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

mean covariance

- Maximum likelihood estimation

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

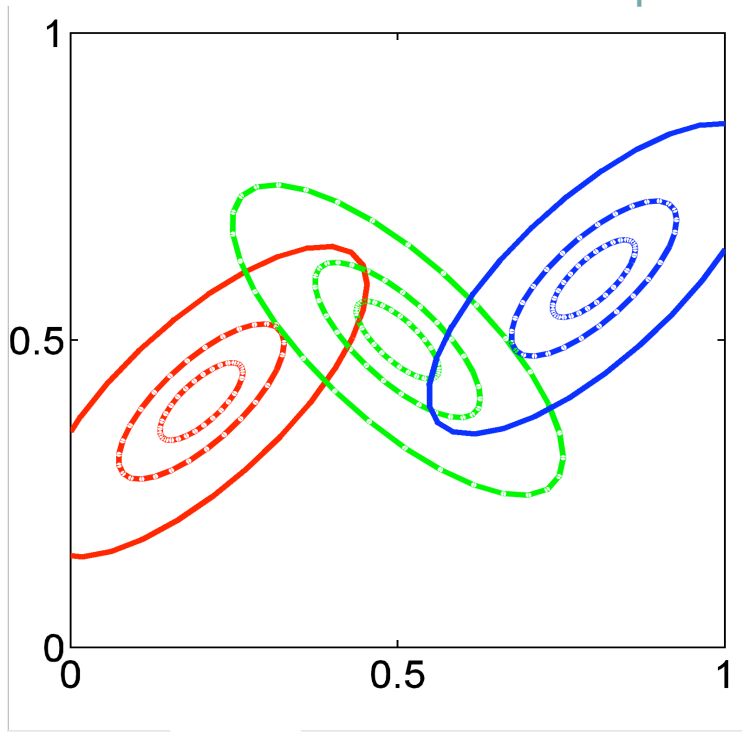
$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(x_i-\hat{\mu})(x_i-\hat{\mu})^T$$

Gaussian Mixture

- Linear combination of Gaussians

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \quad \text{where} \quad \sum_{k=1}^{K} \pi_k = 1, \quad 0 \leqslant \pi_k \leqslant 1$$

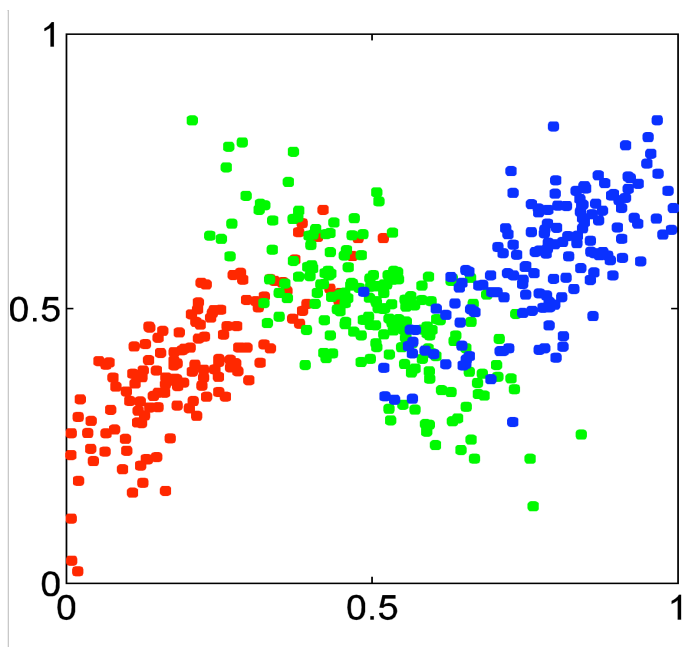
parameters to be estimated



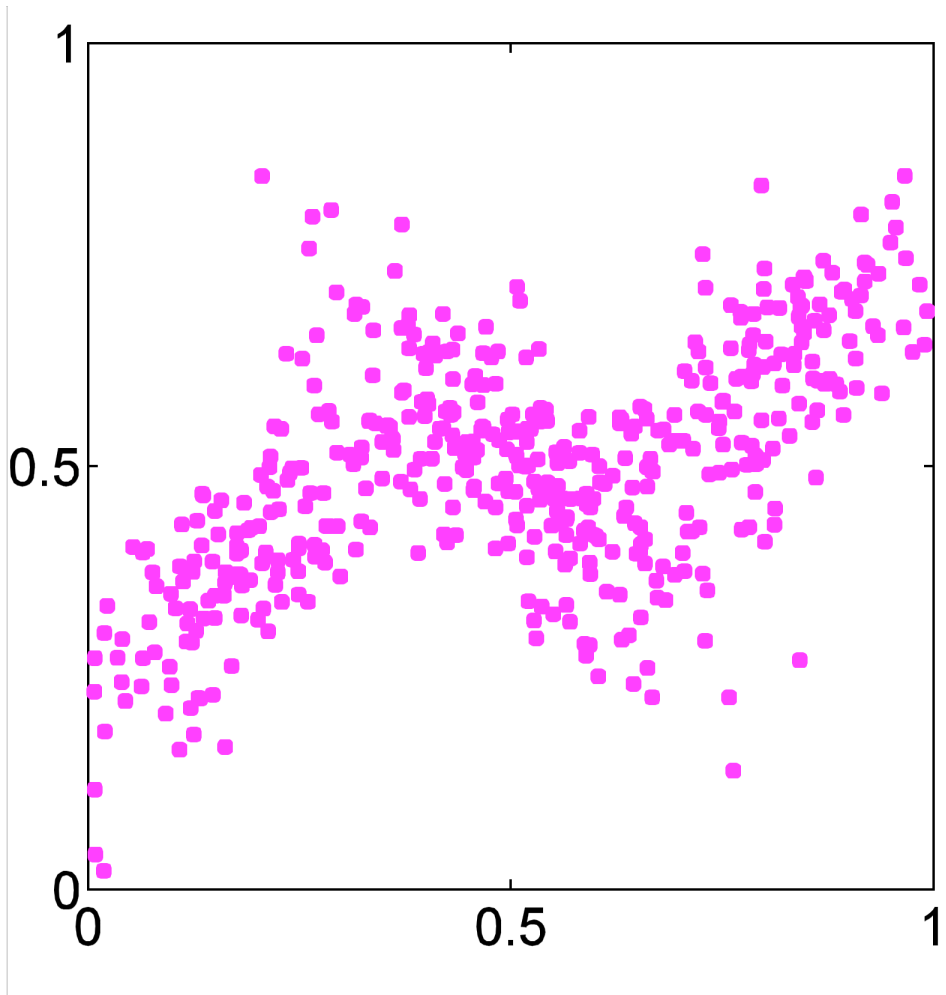
Gaussian Mixture

- To generate a data point:
  - first pick one of the components with probability π_k
  - then draw a sample x_i from that component distribution
- Each data point is generated by one of K components, a latent variable $z_i = (z_{i1}, \ldots, z_{iK})$ is associated with each x_i

$$\sum_{k=1}^{K} z_{ik} = 1 \text{ and } p(z_{ik} = 1) = \pi_k$$



Synthetic Data Set Without Colours



Gaussian Mixture

- Loss function: The negative log likelihood of the data.
  - Equivalently, maximize the log likelihood.

$$\ln p(x|\pi, \mu, \Sigma) = \sum_{i=1}^{n} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)\}$$

- Without knowing values of latent variables, we have to maximize the incomplete log likelihood.
  - Sum over components appears inside the logarithm, no closed-form solution.

Fitting the Gaussian Mixture

- Given the complete data set $(x, z) = (x_i, z_i)_{i=1,\ldots,n}$
  - Maximize the complete log likelihood.

$$\ln p(x, z|\pi, \mu, \Sigma) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \{\ln \pi_k + \ln \mathcal{N}(x_i|\mu_k, \Sigma_k)\}$$

  - Trivial closed-form solution: fit each component to the corresponding set of data points.
  - Observe that if all the π_k and Σ_k are equal, then the complete log likelihood is exactly the loss function used in K-means.
- Need a procedure that would let us optimize the incomplete log likelihood by working with the (easier) complete log likelihood instead.

The Expectation-Maximization (EM) Algorithm

- E-step: for given parameter values we can compute the expected values of the latent variables (responsibilities of data points)

$$
\begin{aligned}
r_{ik} \equiv E(z_{ik}) &= p(z_{ik} = 1 | x_i, \pi, \mu, \Sigma) \\
&= \frac{p(z_{ik} = 1) p(x_i | z_{ik} = 1, \pi, \mu, \Sigma)}{\sum_{k=1}^{K} p(z_{ik} = 1) p(x_i | z_{ik} = 1, \pi, \mu, \Sigma)} \\
&= \frac{\pi_k \mathcal{N}(x_i | u_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i | u_k, \Sigma_k)}
\end{aligned}
$$

Bayes rule

- Note that $r_{ik} \in [0, 1]$ instead of $\{0, 1\}$ but we still have $\sum_{k=1}^{K} r_{ik} = 1 \text{ for all } i$

The EM Algorithm

- M-step: maximize the expected complete log likelihood

$$E[\ln p(x, z|\pi, \mu, \Sigma)] = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \{\ln \pi_k + \ln \mathcal{N}(x_i|\mu_k, \Sigma_k)\}$$

- Parameter update:

$$\pi_k = \frac{\sum_i r_{ik}}{n} \qquad \mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}$$

$$\Sigma_k = \frac{\sum_i r_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i r_{ik}}$$

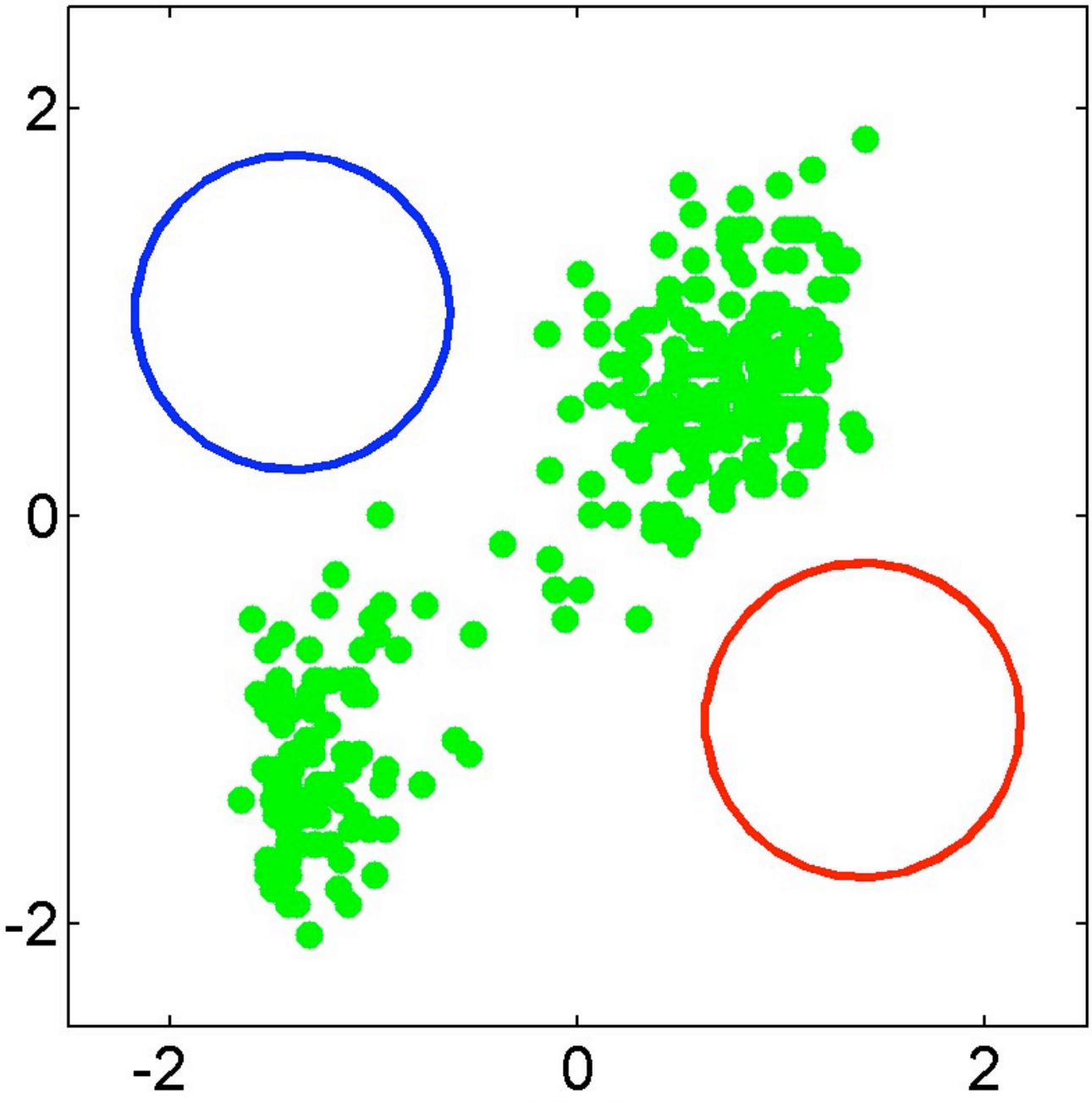
The EM Algorithm

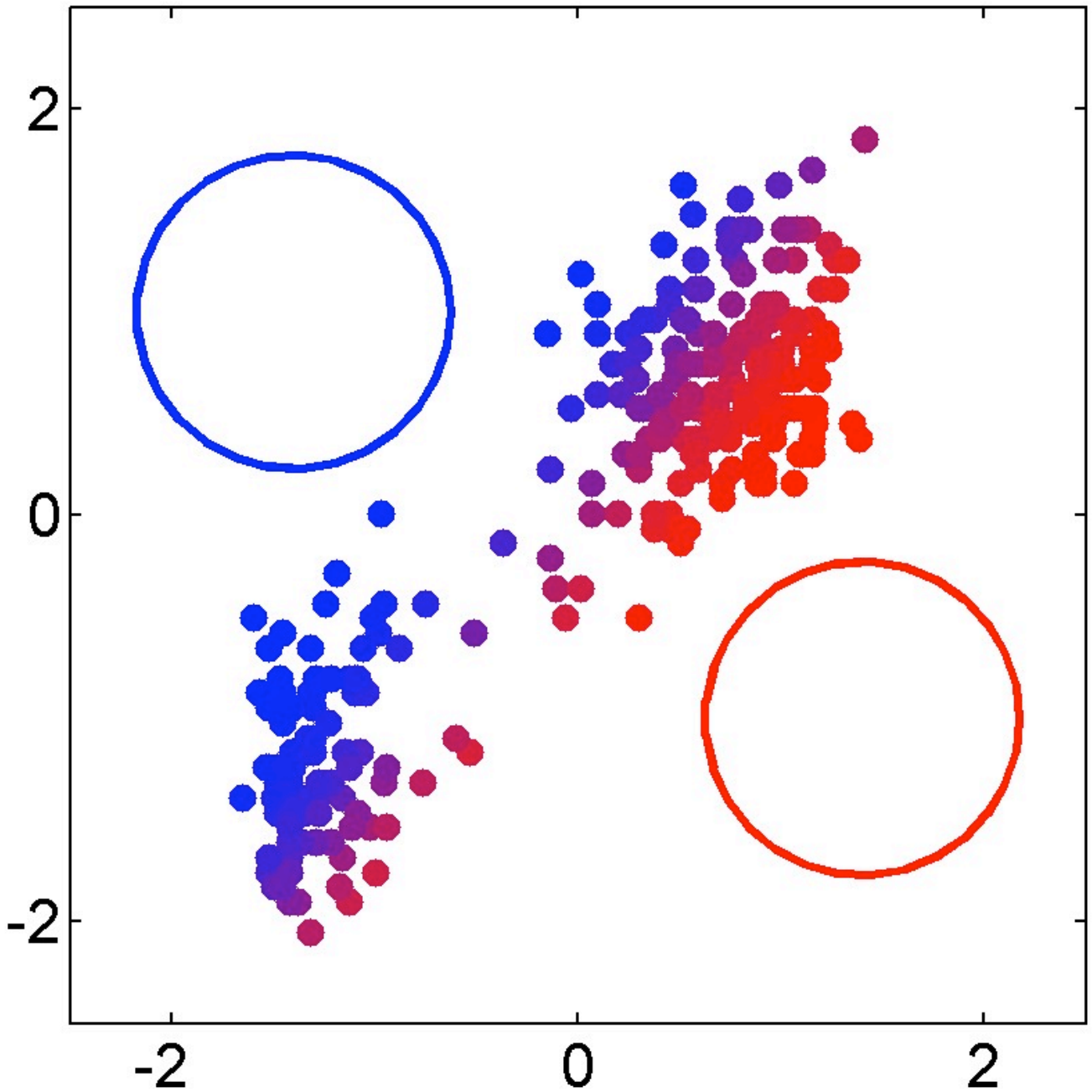
- Iterate E-step and M-step until the log likelihood of data does not increase any more.
  - Converge to local optima.
  - Need to restart algorithm with different initial guess of parameters (as in K-means).

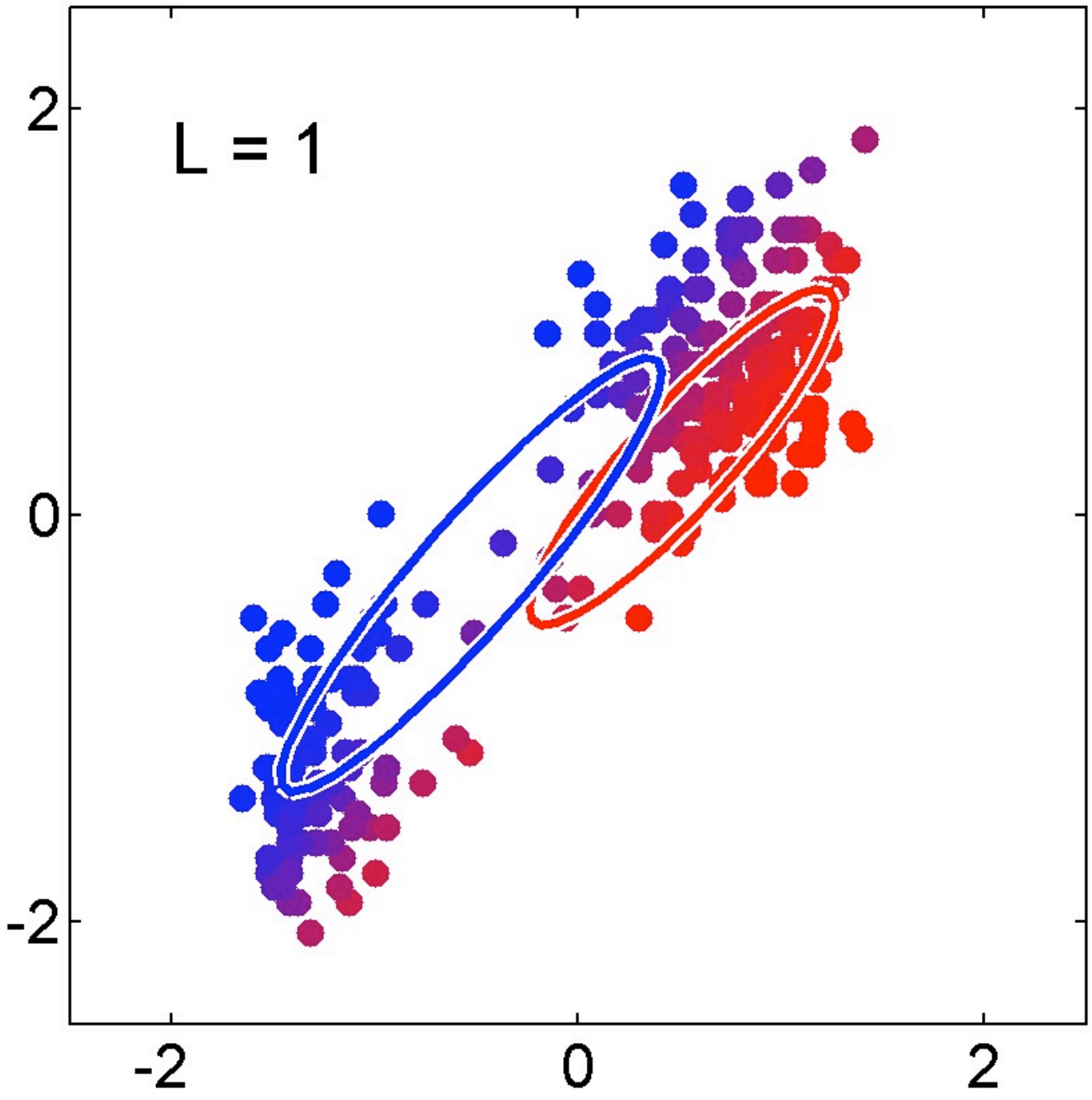
- Relation to K-means
  - Consider GMM with common covariance.

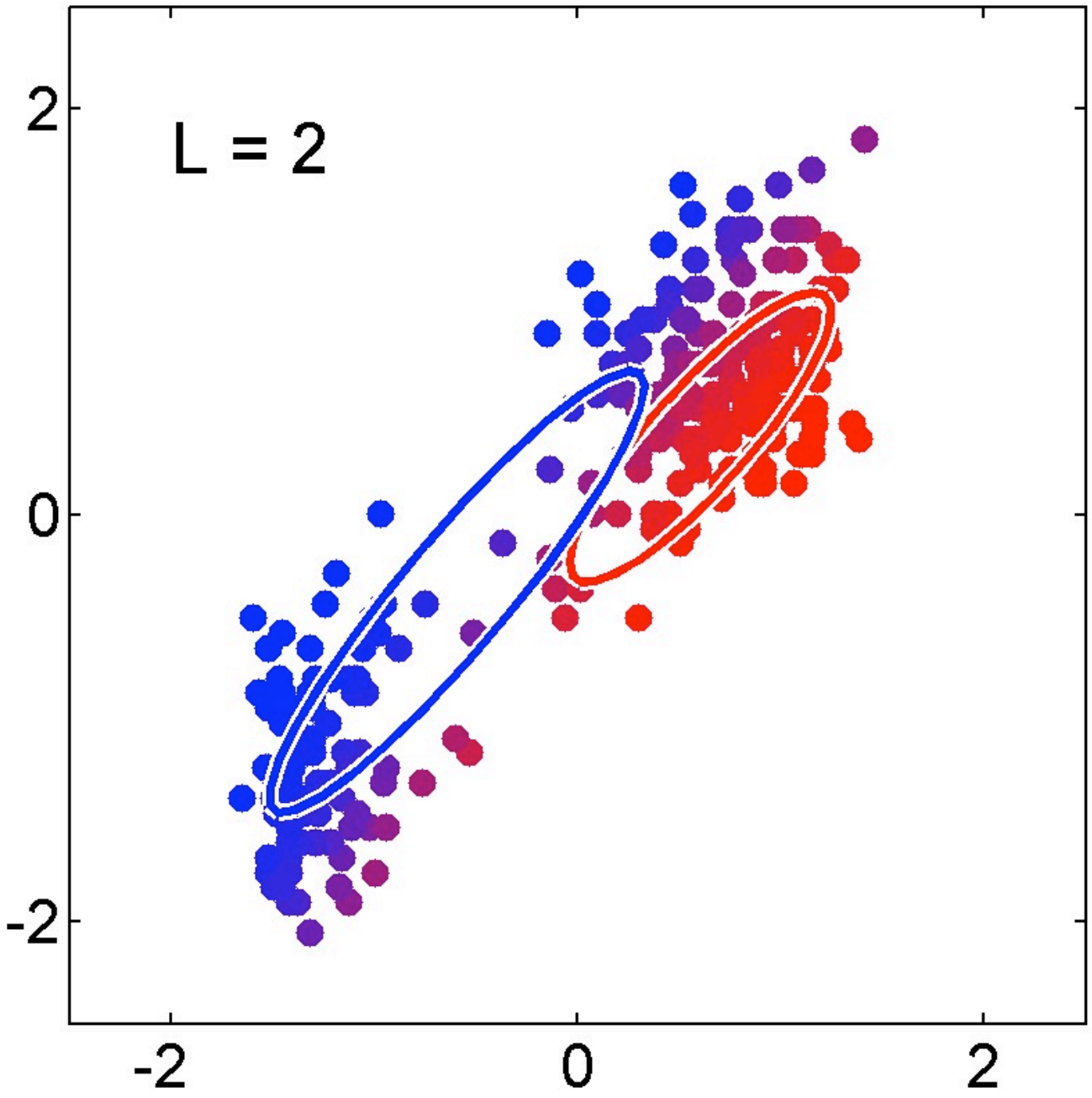
$$\Sigma_k = \delta^2 I$$

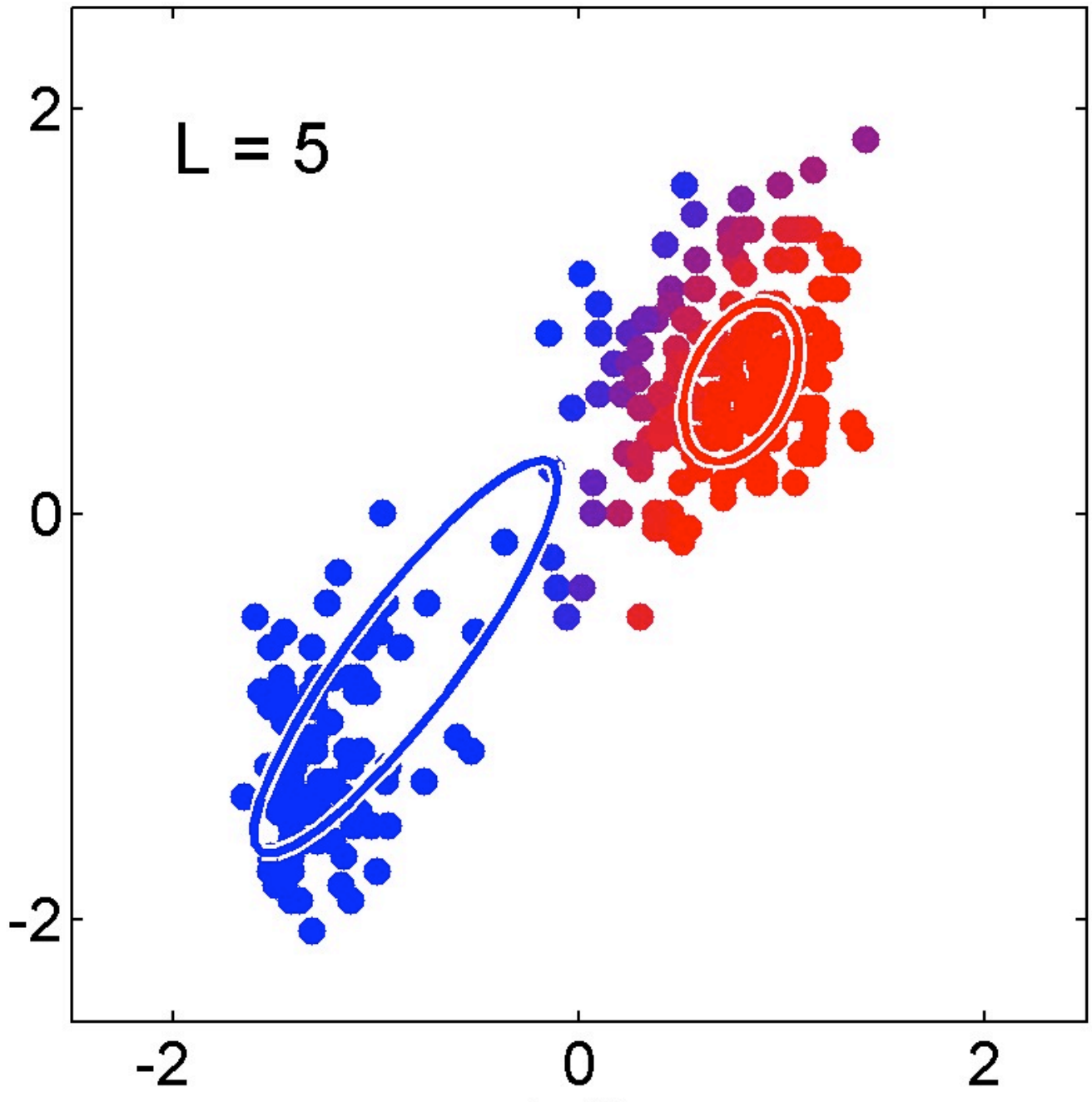
  - As $\delta^2 \to 0, r_{ik} \to 0 \text{ or } 1$, two methods coincide.

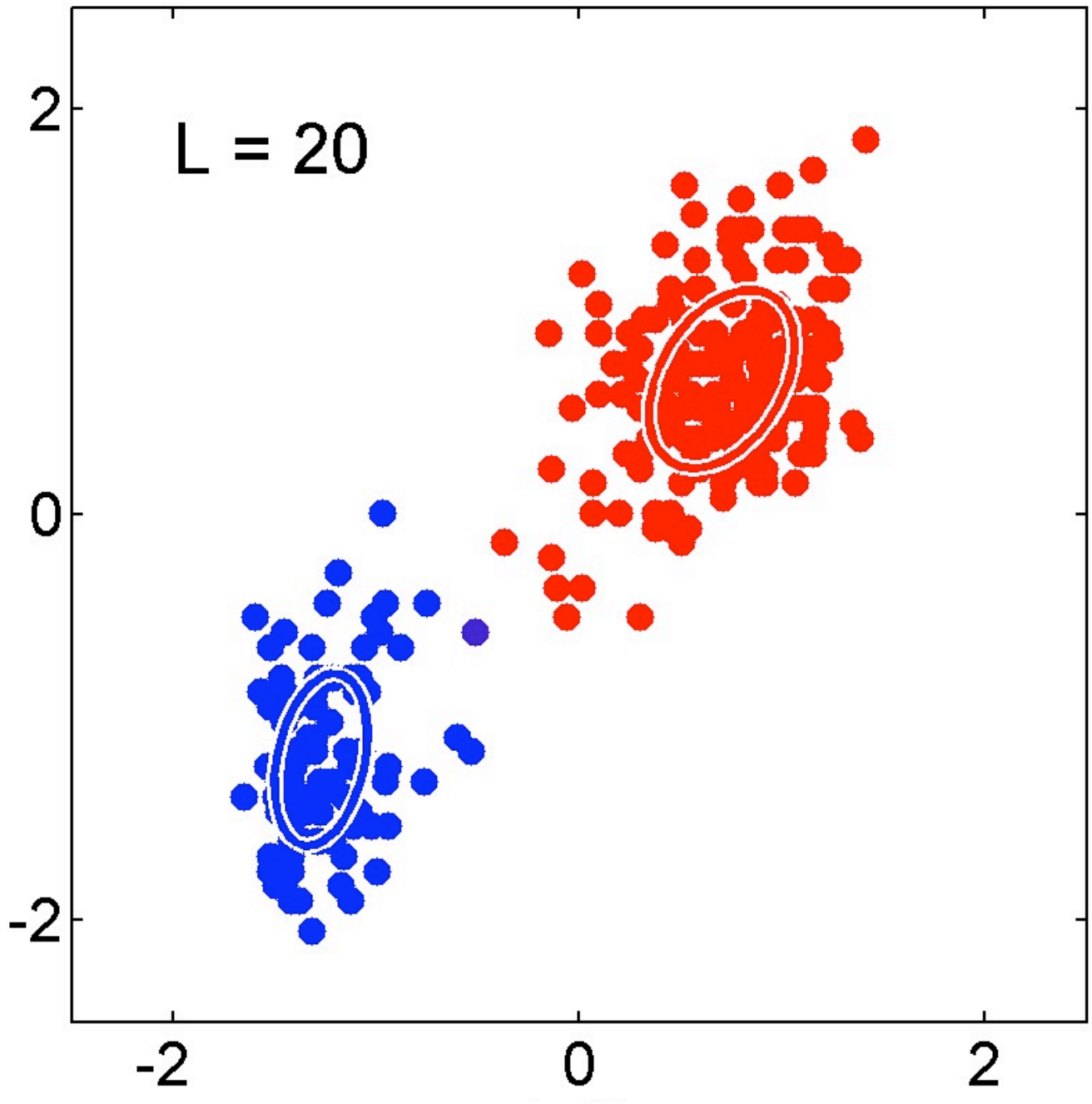

2
0
-2
-2
0
2


2
0
-2
-2
0
2


L = 1
2
0
-2
-2
0
2


L = 2
2
0
-2
-2
0
2


L = 5
2
0
-2
-2
0
2


L = 20
2
0
-2
-2
0
2

K-means vs GMM

- Loss function:
  - Minimize sum of squared Euclidean distance.
- Can be optimized by an EM algorithm.
  - E-step: assign points to clusters.
  - M-step: optimize clusters.
  - Performs hard assignment during E-step.
- Assumes spherical clusters with equal probability of a cluster.

- Loss function
  - Minimize the negative log-likelihood.
- EM algorithm
  - E-step: Compute posterior probability of membership.
  - M-step: Optimize parameters.
  - Perform soft assignment during E-step.
- Can be used for non-spherical clusters. Can generate clusters with different probabilities.

K-medoids

- K-means not robust.
  - Squared Euclidean distance gives greater weight to more distant points.
- Only the dissimilarity matrix may be given and not the attributes.
- Attributes may not be quantitative.

K-medoids

- Restrict the prototypes to one of the data points assigned to the cluster.
- E-step: Fix the prototypes and minimize J w.r.t. r_{ik}
  - Assigns each data point to its nearest prototype
- M-step: Fix values for r_{ik} and minimize J w.r.t the prototypes.

K-medoids: Example

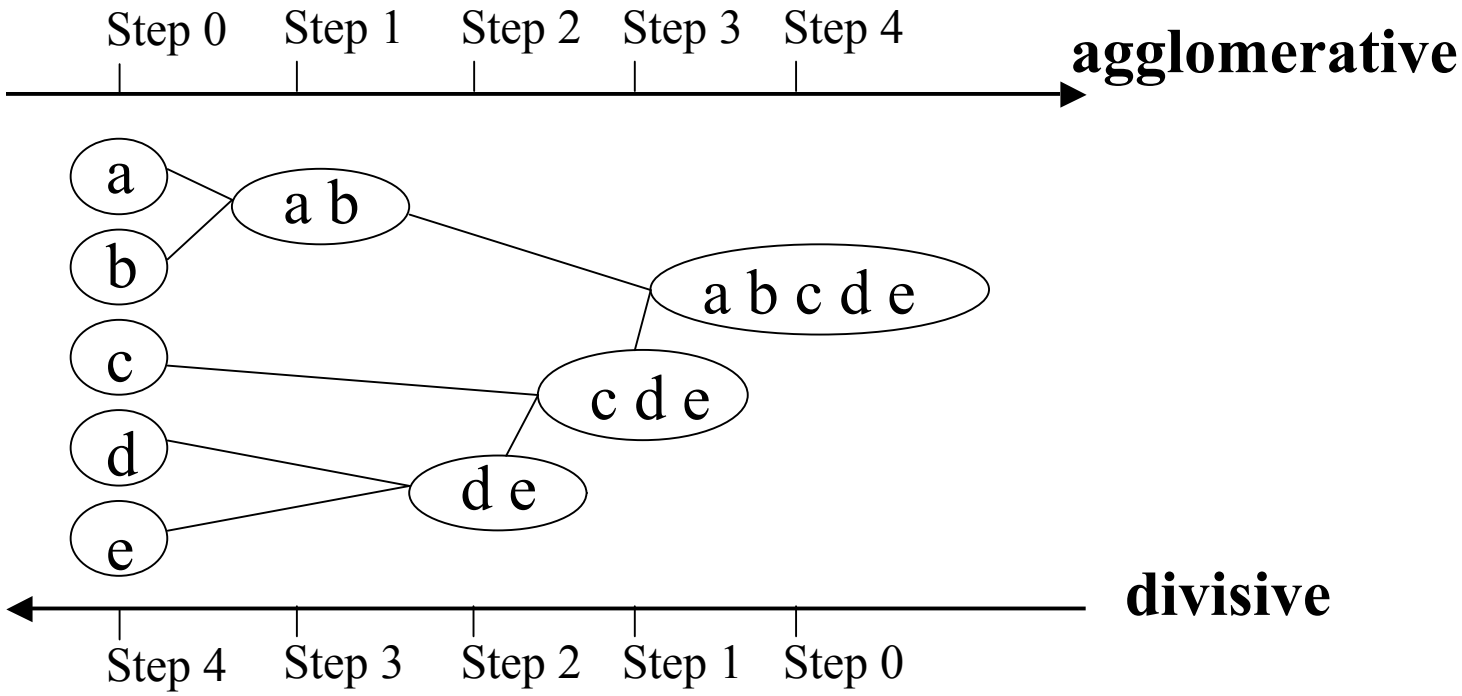
- Use L1 distance instead of squared Euclidean distance.
- Prototype is the median of points in a cluster.
- Recall the connection between linear and L1 regression.

Outline

- Introduction
  - Unsupervised learning
  - What is cluster analysis?
  - Applications of clustering
- Dissimilarity (similarity) of samples
- Clustering algorithms
  - K-means
  - Gaussian mixture model (GMM)
  - **Hierarchical clustering**
  - Spectral clustering

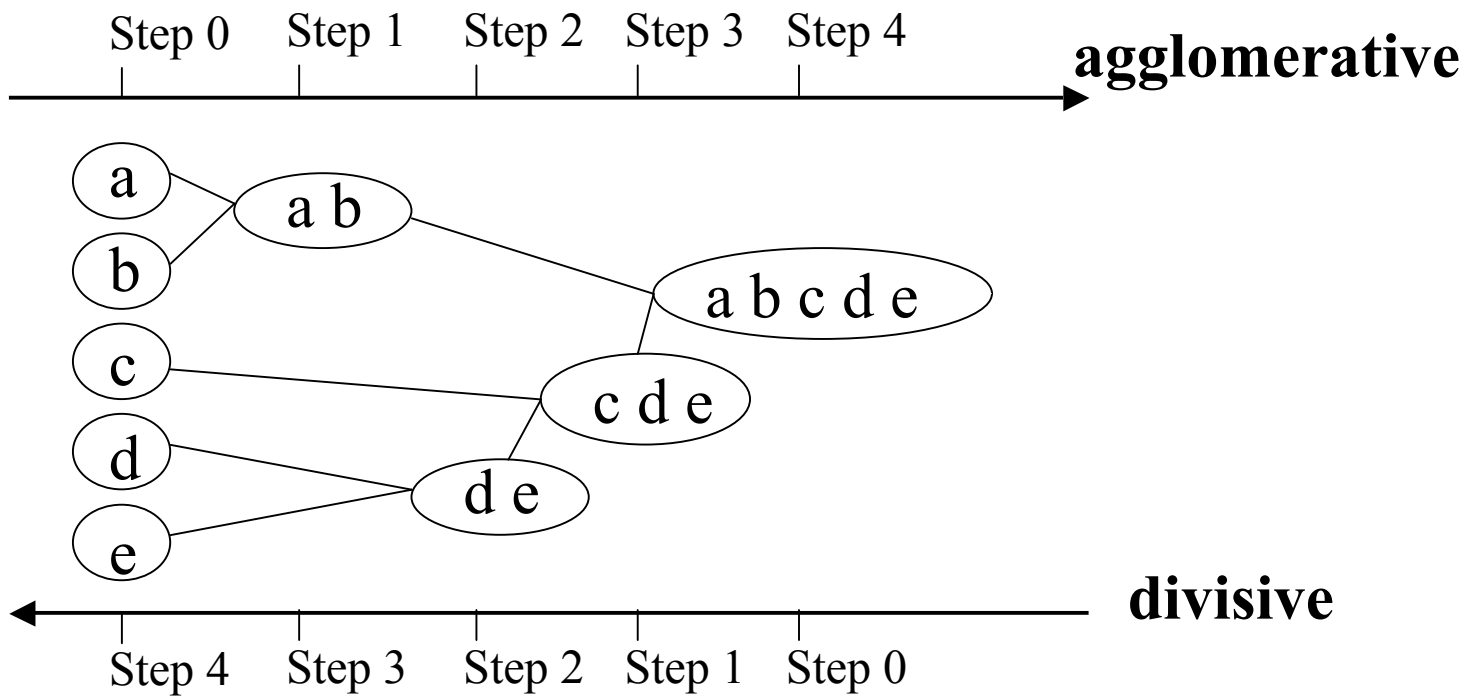
Hierarchical Clustering

- Organize the clusters in an hierarchical way
- Produces a rooted (binary) tree (**dendrogram**)



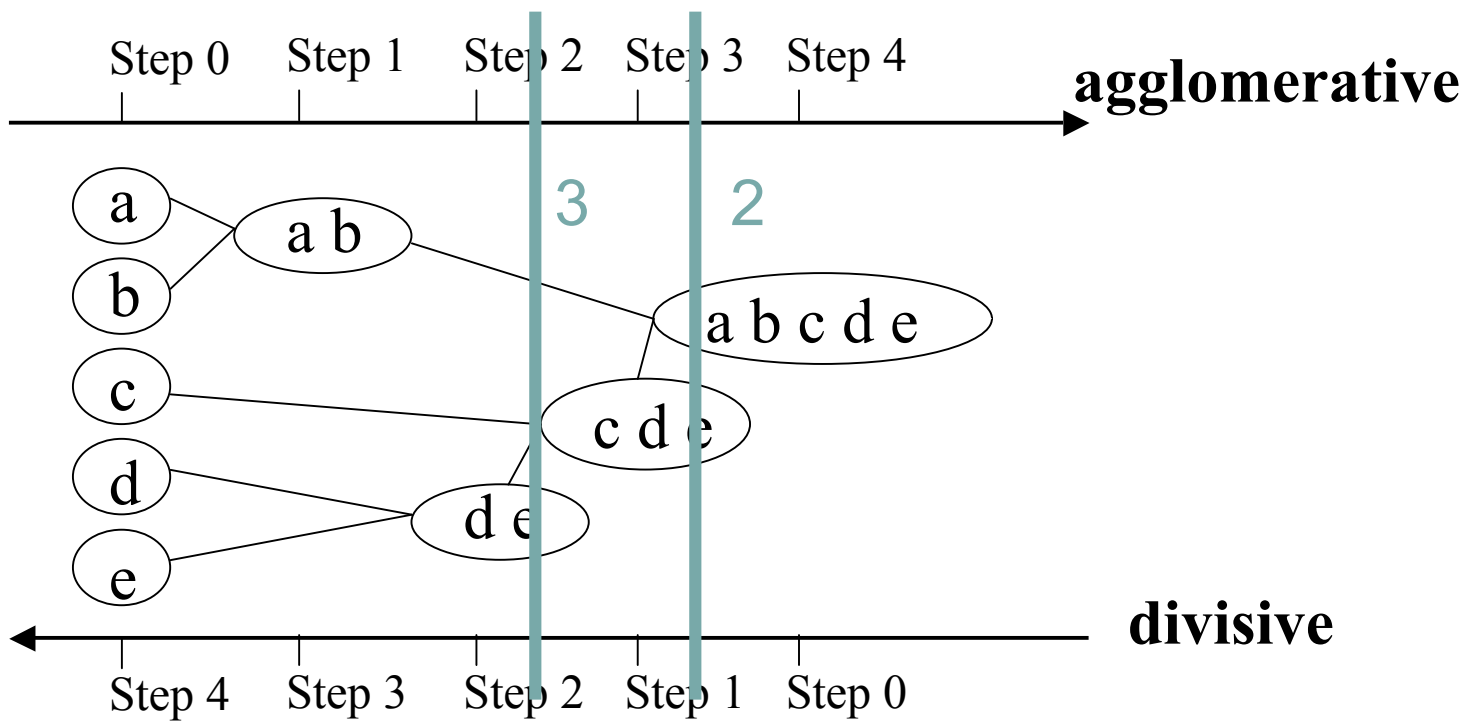
Hierarchical Clustering

- Two kinds of strategy
  - Bottom-up (agglomerative): Recursively merge two groups with the smallest between-cluster dissimilarity (defined later on).
  - Top-down (divisive): In each step, split a least coherent cluster (e.g. largest diameter); splitting a cluster is also a clustering problem (usually done in a greedy way); less popular than bottom-up way.



Hierarchical Clustering

- User can choose a cut through the hierarchy to represent the most natural division into clusters
  - e.g, Choose the cut where intergroup dissimilarity exceeds some threshold



Hierarchical Clustering

- Have to measure the dissimilarity for two disjoint groups *G* and *H*, $D(G, H)$ is computed from pairwise dissimilarities $D(i, j)$ with $i \in G, j \in H$
  - Single Linkage: tends to yield extended clusters.

$$D_{SL}(G, H) = \min_{i \in G, j \in H} D(i, j)$$

  - Complete Linkage: tends to yield round clusters.

$$D_{CL}(G, H) = \max_{i \in G, j \in H} D(i, j)$$

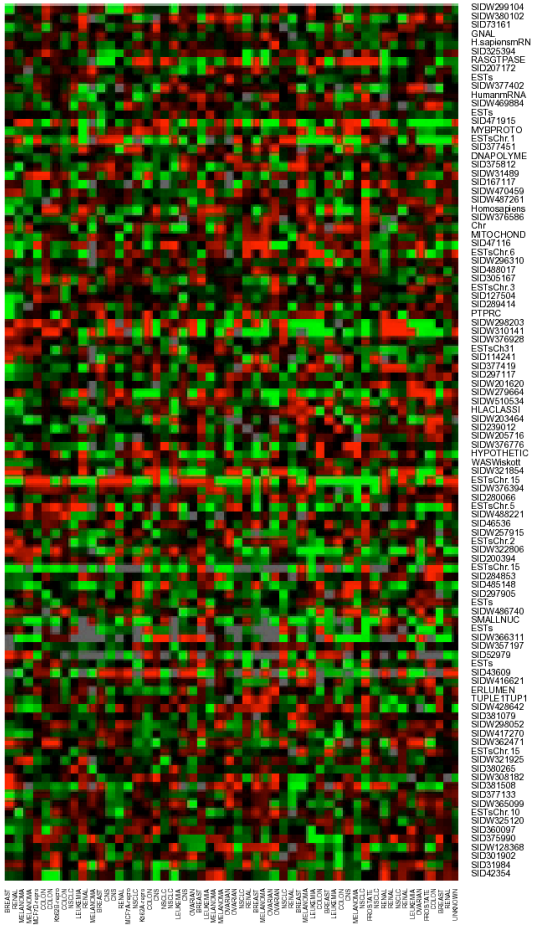
  - Group Average: tradeoff between them. Not invariant under monotone increasing transform.

$$D_{GA}(G, H) = \frac{1}{n_G n_H} \sum_{i \in G, j \in H} D(i, j)$$

Example: Human Tumor Microarray Data

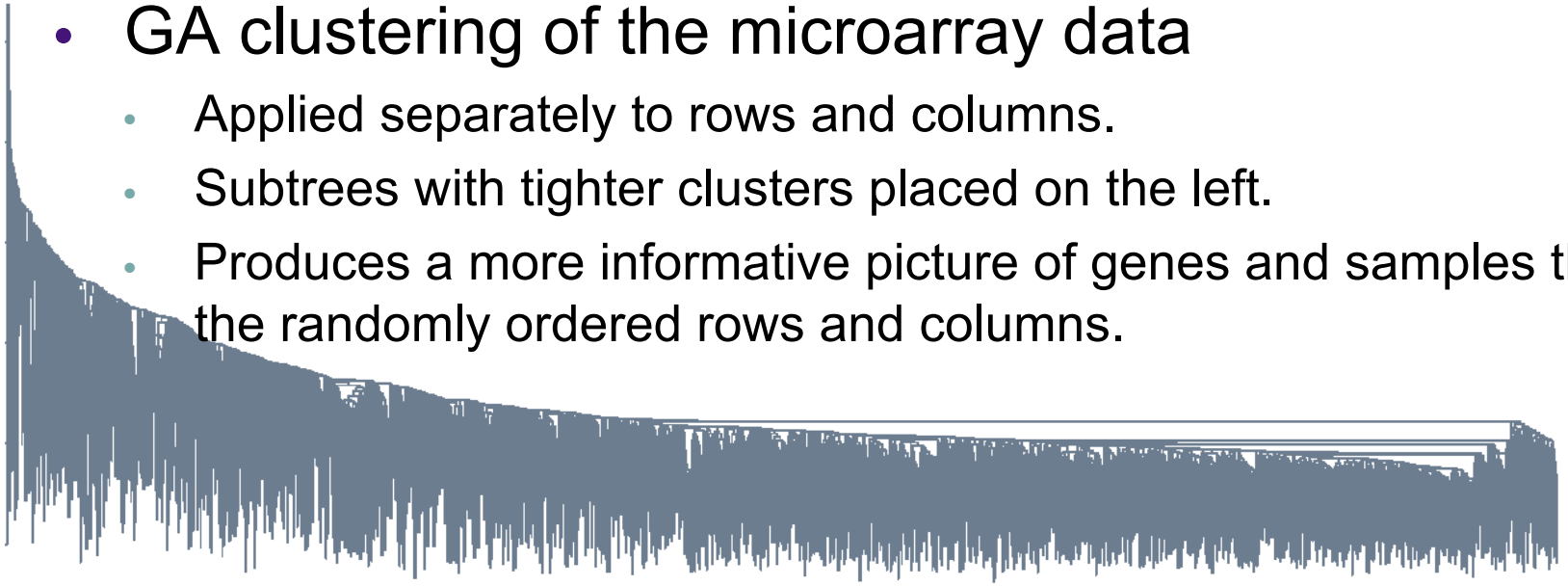
- 6830×64 matrix of real numbers.
- Rows correspond to genes, columns to tissue samples.
- Cluster rows (genes) can deduce functions of unknown genes from known genes with similar expression profiles.
- Cluster columns (samples) can identify disease profiles: tissues with similar disease should yield similar expression profiles.

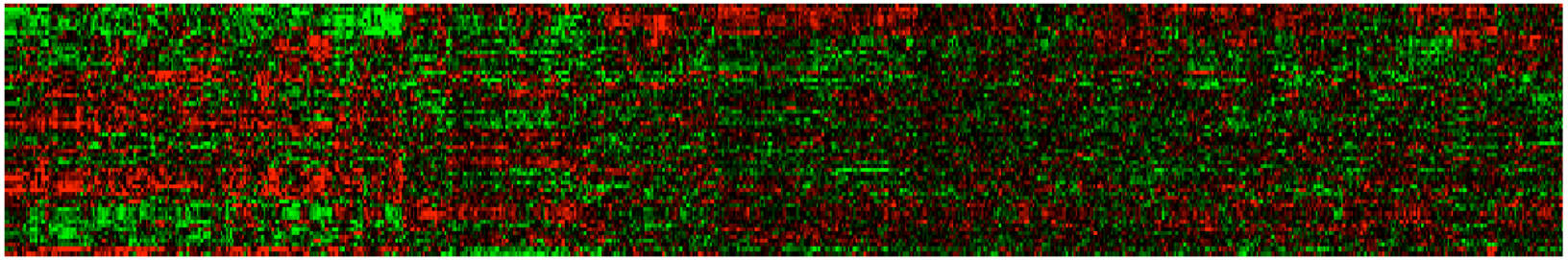
Gene expression matrix



Example: Human Tumor Microarray Data

- 6830×64 matrix of real numbers
- GA clustering of the microarray data
  - Applied separately to rows and columns.
  - Subtrees with tighter clusters placed on the left.
  - Produces a more informative picture of genes and samples than the randomly ordered rows and columns.



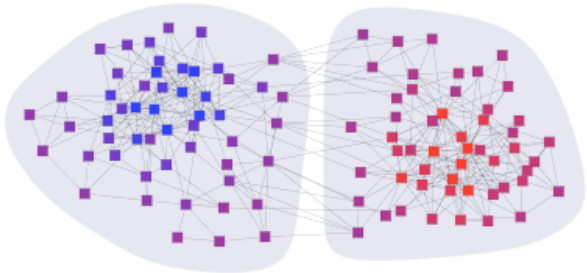


Outline

- Introduction
  - Unsupervised learning
  - What is cluster analysis?
  - Applications of clustering
- Dissimilarity (similarity) of samples
- Clustering algorithms
  - K-means
  - Gaussian mixture model (GMM)
  - Hierarchical clustering
  - **Spectral clustering**

Spectral Clustering

- Represent datapoints as the vertices *V* of a graph *G*.
- All pairs of vertices are connected by an edge *E*.
- Edges have weights *W*.
  - Large weights mean that the adjacent vertices are very similar; small weights imply dissimilarity.



Graph partitioning

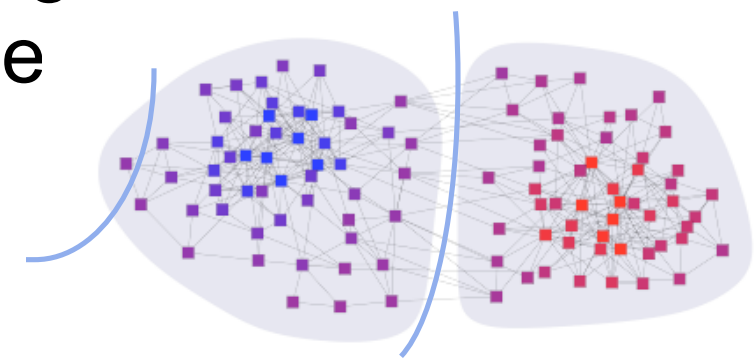
- Clustering on a graph is equivalent to partitioning the vertices of the graph.
- A loss function for a partition of *V* into sets *A* and *B*

$$cut(A, B) = \sum_{u \in A, v \in B} W_{u,v}$$

- In a good partition, vertices in different partitions will be dissimilar.
  - Mincut criterion: Find a partition A, B that minimizes $cut(A, B)$

Graph partitioning

- Mincut criterion ignores the size of the subgraphs forme



- Normalized cut criterion favors balanced partitions.

$$Ncut(A,B) = \frac{cut(A,B)}{\sum_{u \in A, v \in V} W_{u,v}} + \frac{cut(A,B)}{\sum_{u \in B, v \in V} W_{u,v}}$$

- Minimizing the normalized cut criterion exactly is NP-hard.

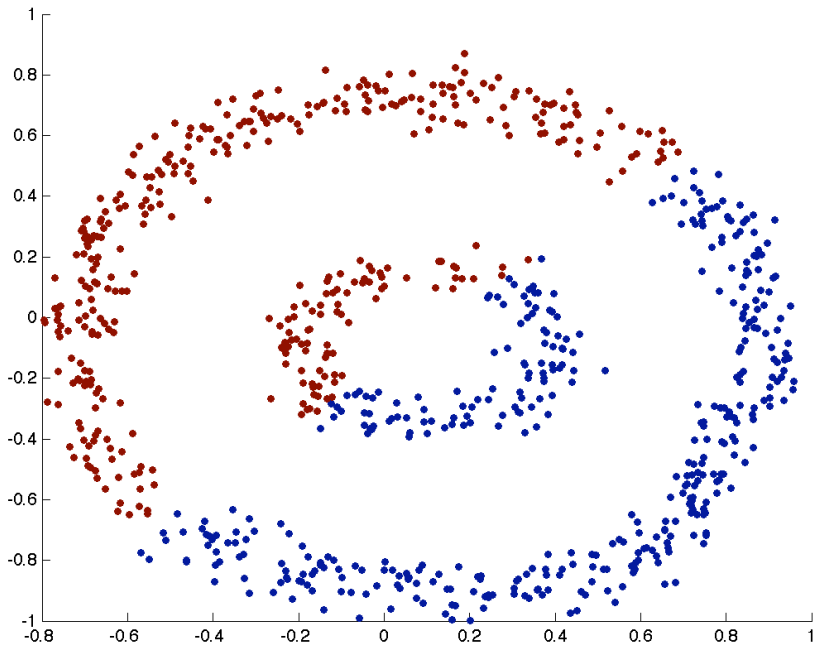
Spectral Clustering

- One way of approximately optimizing the normalized cut criterion leads to spectral clustering.
- Spectral clustering
  - Find a new representation of the original data points.
  - Cluster the points in this representation using any clustering scheme (say 2-means).
- The representation involves forming the row-normalized matrix Y using the largest 2 eigenvectors of the matrix L

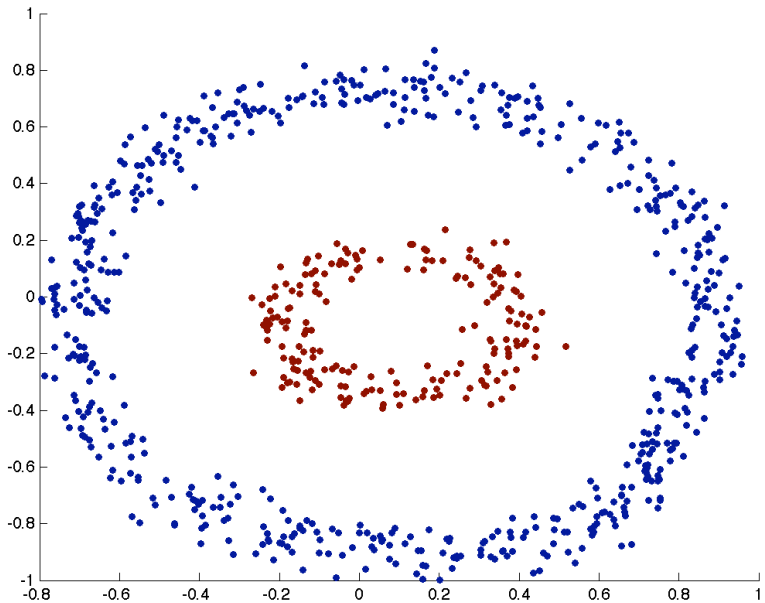
$$D = diag(W1) \quad \text{and} \quad L = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

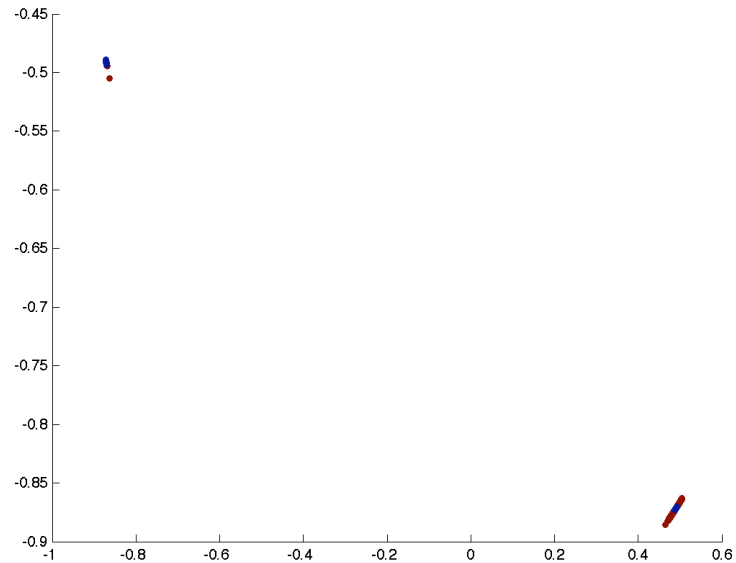
$$W_{uv} = exp(- \| s_u - s_j \|^2 / (2\sigma^2))$$

Example: 2-means



Example: Spectral clustering





Learning Dissimilarity

- Suppose a user indicates certain objects are considered by him to be "similar":

$$(x_i, x_j) \in \mathcal{S} \text{ if } x_i \text{ and } x_j \text{ are similar}$$

- Consider learning a dissimilarity that respects this subjectivity

$$D(x_i, x_j) = ||x_i - x_j||_A = \sqrt{(x_i - x_j)^T A (x_i - x_j)}$$

  - If *A* is identity matrix, it corresponds to Euclidean distance
  - Generally, *A* parameterizes a family of Mahalanobis distance

- Leaning such a dissimilarity is equivalent to finding a rescaling of data by replacing x with $A^{1/2}x$, and then applying the standard Euclidean distance

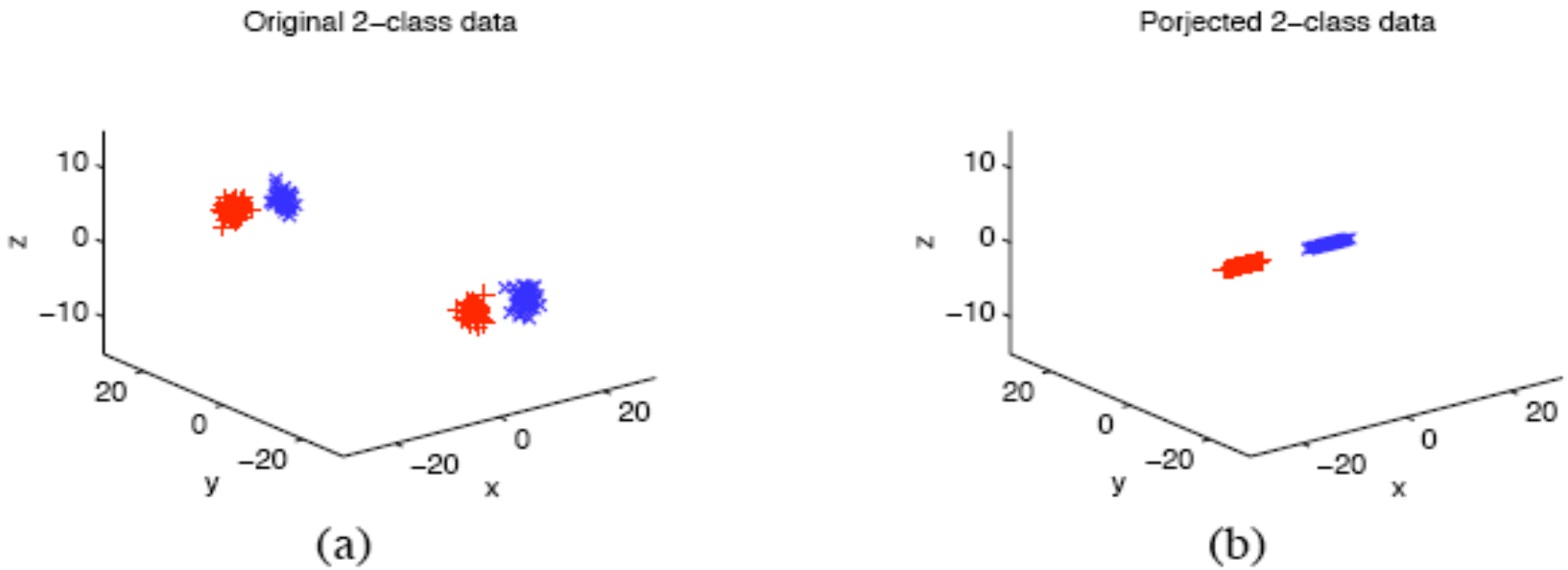
Learning Dissimilarity

- A simple way to define a criterion for the desired dissimilarity:

$$\begin{aligned} \min_{A} \quad & \sum_{(x_i,x_j)\in\mathcal{S}} ||x_i - x_j||_A^2 \\ \text{s.t.} \quad & \sum_{(x_i,x_j)\in\mathcal{D}} ||x_i - x_j||_A \geq 1, \\ & A \succeq 0. \end{aligned}$$

- A convex optimization problem, could be solved by gradient descent and iterative projection
- For details, see [Xing, Ng, Jordan, Russell ’03]

Learning Dissimilarity



References

- Hastie, Tibshirani and Friedman, The Elements of Statistical Learning, Chapter 14
- Bishop, Pattern Recognition and Machine Learning, Chapter 9