UVA CS 6316/4501 – Fall 2016 Machine Learning

Lecture 20: Unsupervised Clustering (II)

Dr. Yanjun Qi

University of Virginia

Department of Computer Science

Where are we ? → major sections of this course

- ❑ Regression (supervised)
- ❑ Classification (supervised)
  - ❑ Feature selection
- ❑ Unsupervised models
  - ❑ Dimension Reduction (PCA)
  - ❑ Clustering (K-means, GMM/EM, Hierarchical)
- ❑ Learning theory
- ❑ Graphical models
  - ❑ (BN and HMM slides shared)

An unlabeled Dataset X

	X_1	X_2	X_3
s_1			
s_2			
s_3			
s_4			
s_5			
s_6			

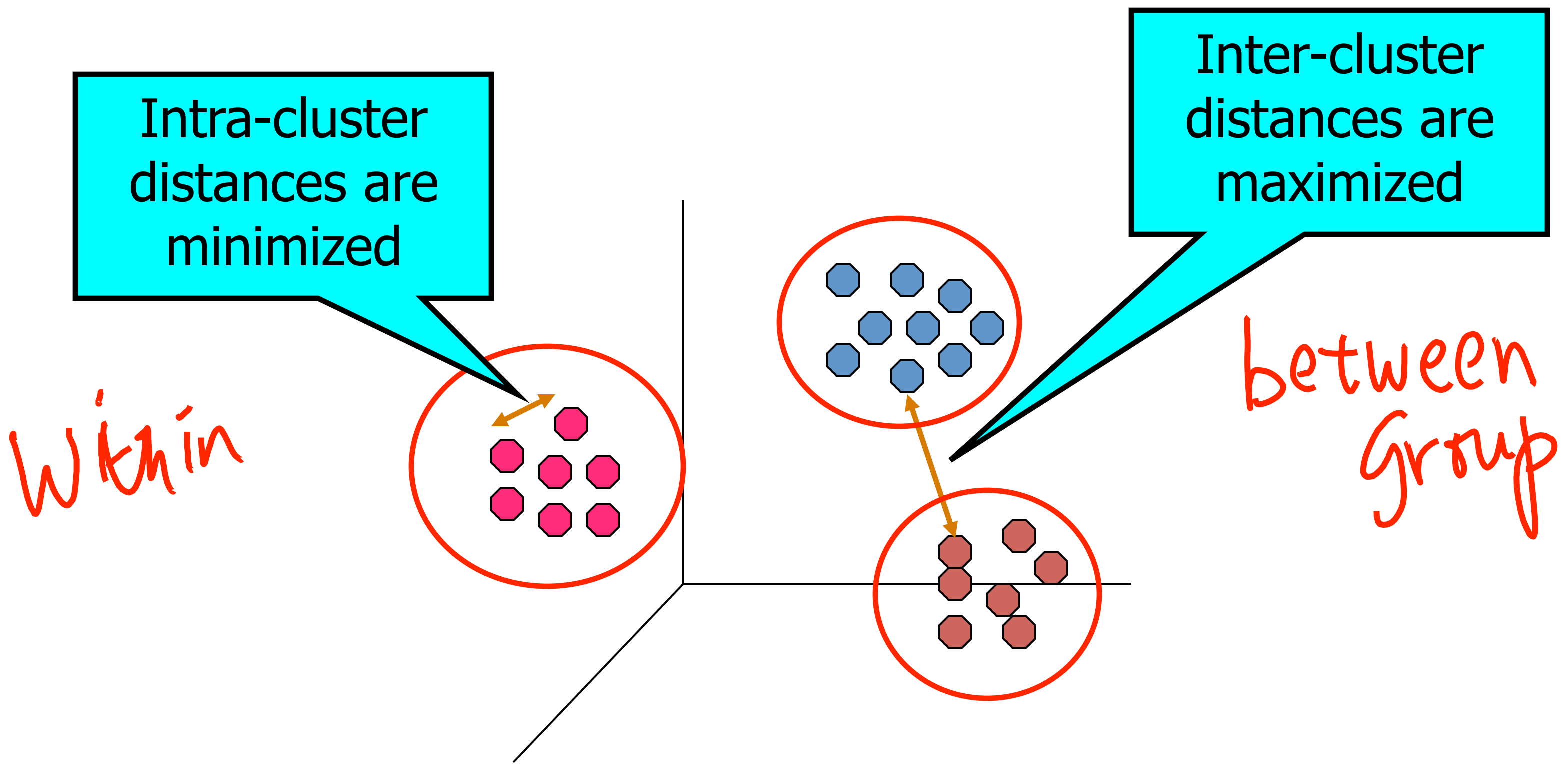
a data matrix of n observations on p variables $x_1, x_2, \ldots x_p$

Unsupervised learning = learning from raw (unlabeled, unannotated, etc) data, as opposed to supervised data where a classification label of examples is given

- **Data**/*points/instances/examples/samples/records*: [rows]
- **Features**/*attributes/dimensions/independent variables/covariates/predictors/regressors*: [columns]

What is clustering?

- Find groups (clusters) of data points such that data points in a group will be similar (or related) to one another and different from (or unrelated to) the data points in other groups

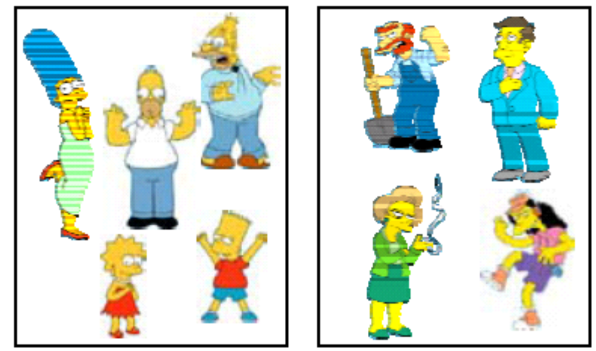


Roadmap: clustering

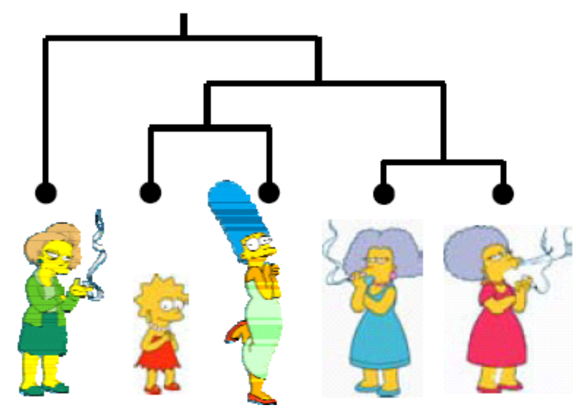
- Definition of "groupness”
- Definition of "similarity/distance"
- Representation for objects
- How many clusters?
- Clustering Algorithms
  - Partitional algorithms
  - Hierarchical algorithms
- Formal foundation and convergence

Clustering Algorithms

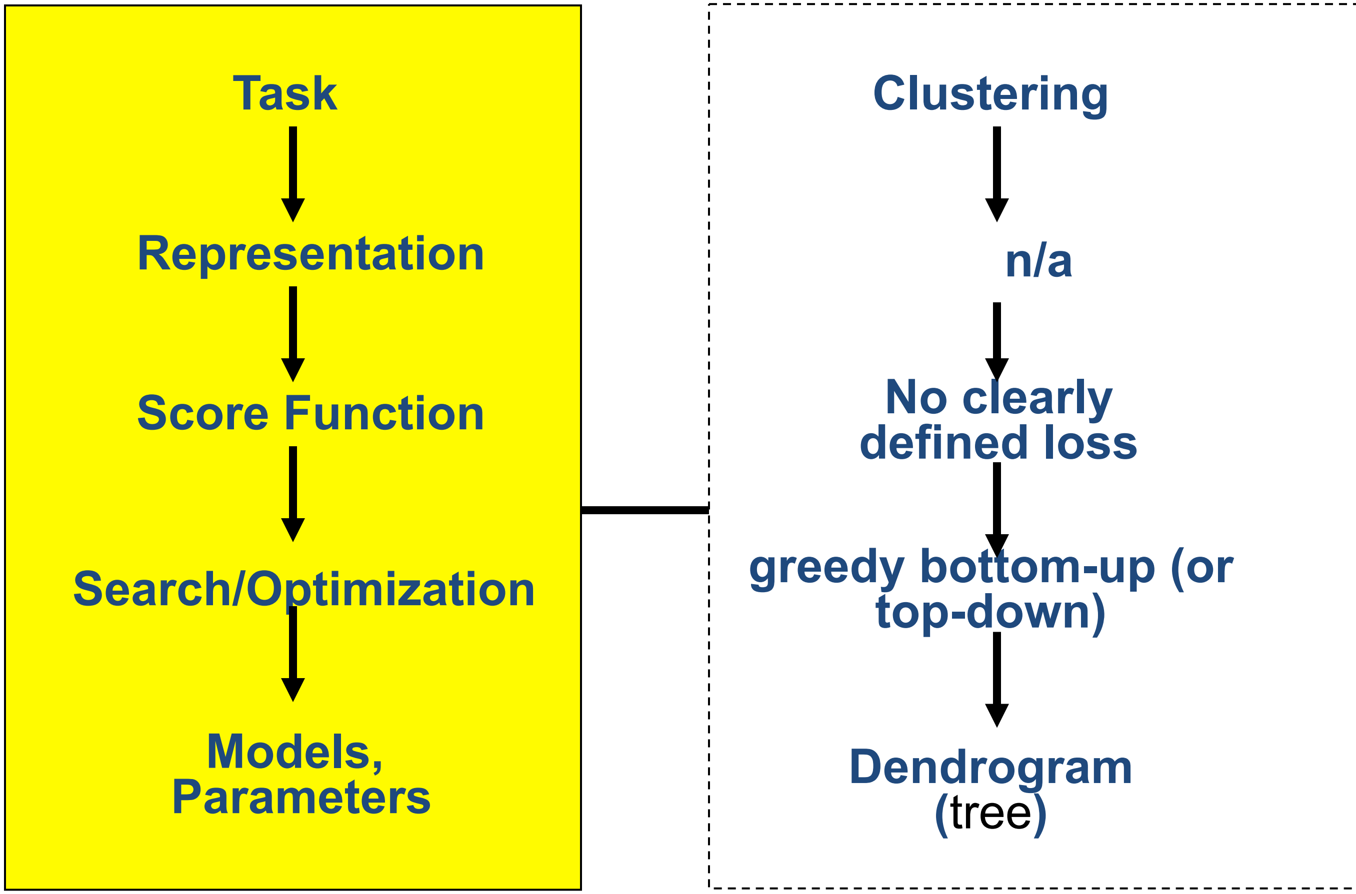
- Partitional algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - K means clustering
    - Mixture-Model based clustering



- Hierarchical algorithms
  - Bottom-up, agglomerative
  - Top-down, divisive

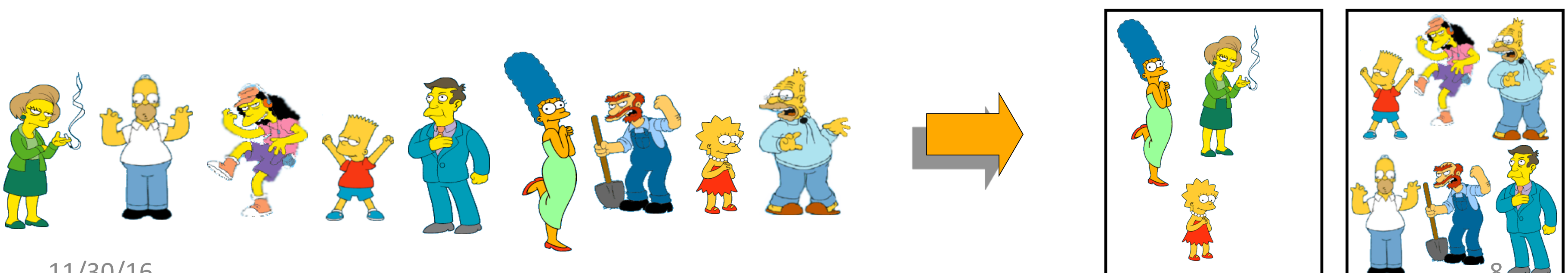


(1) Hierarchical Clustering

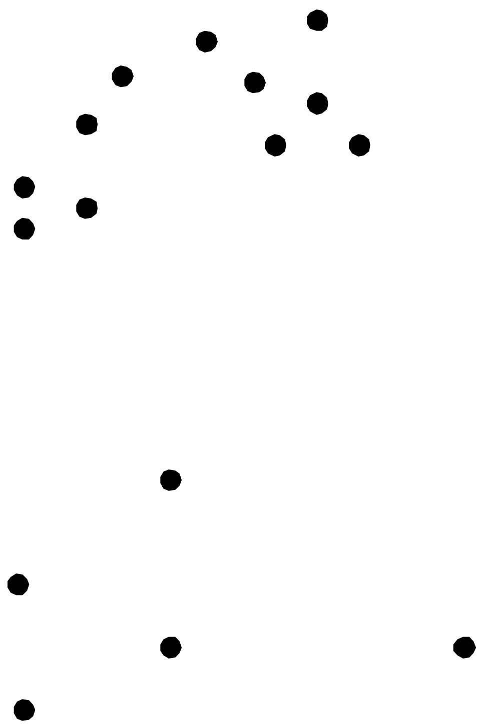


(2) Partitional Clustering

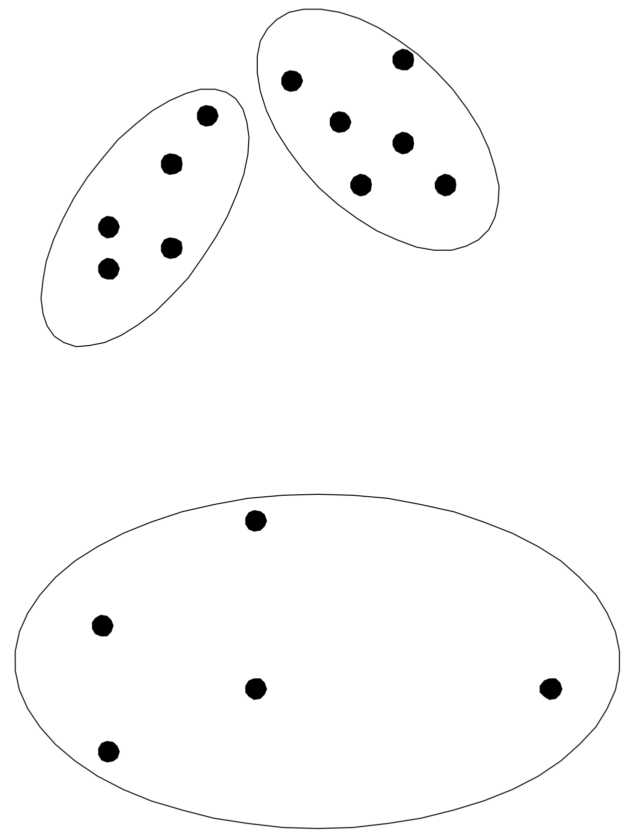
- Nonhierarchical
- Construct a partition of *n* objects into a set of *K* clusters
- User has to specify the desired number of clusters K.



Partitional clustering (e.g. K=3)

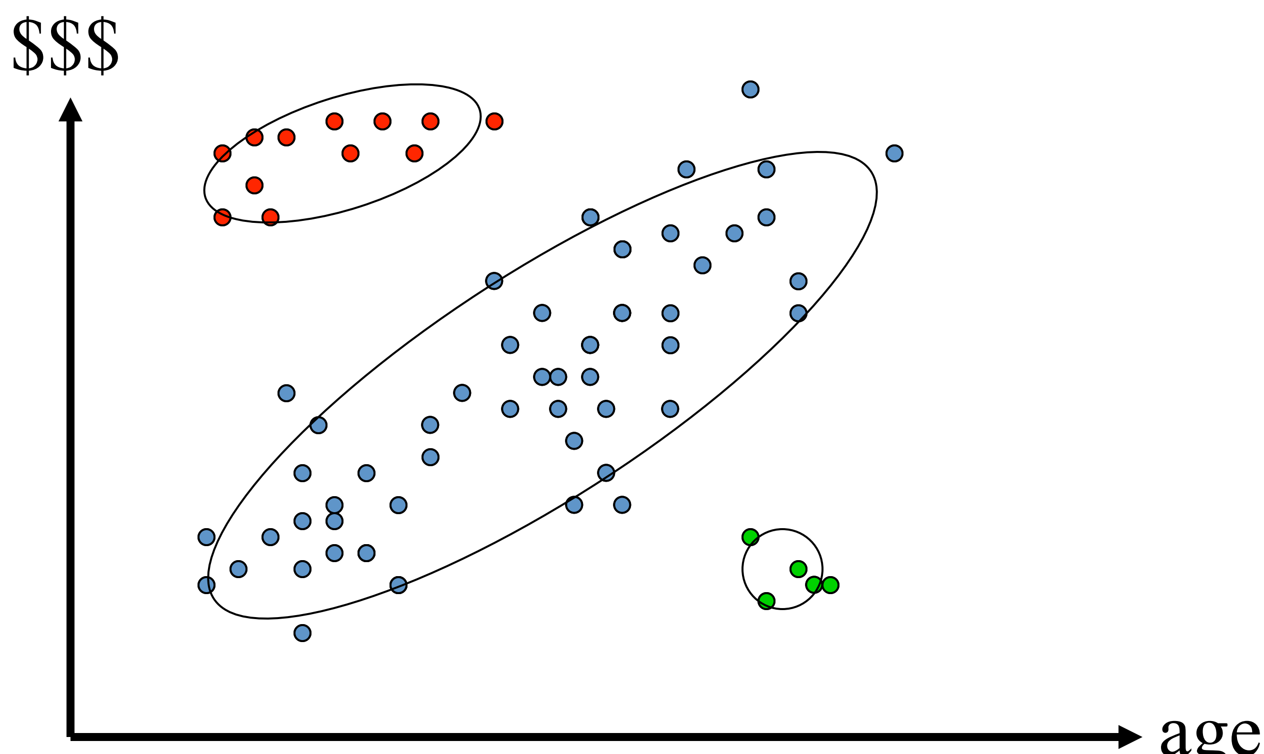


Original points



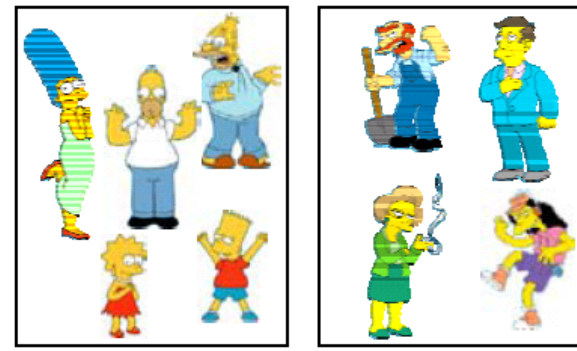
Partitional clustering

Partitional clustering (e.g. K=3)



Clustering Algorithms

- Partitional algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - K means clustering
    - Mixture-Model based clustering



Partitioning Algorithms

- Given: a set of objects and the number *K*
- Find: a partition of *K* clusters that optimizes a chosen partitioning criterion
  - Globally optimal: exhaustively enumerate all partitions
  - Effective heuristic methods: K-means and K-medoids algorithms

too expensive

K^n

K-Means

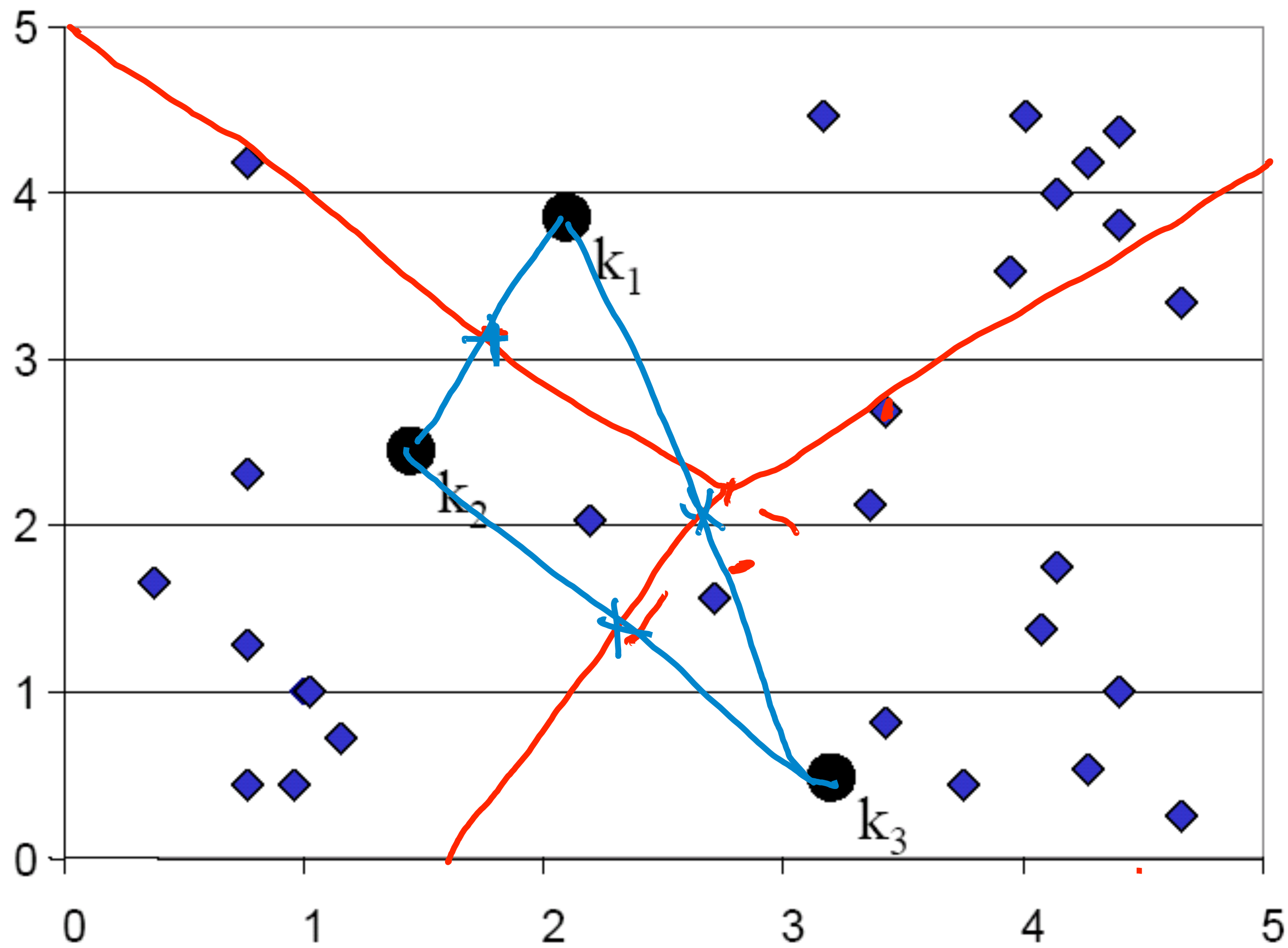
Algorithm

1. Decide on a value for *k*.
2. Initialize the *k* cluster centers randomly if necessary.
3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster centroids (aka the center of gravity or mean)

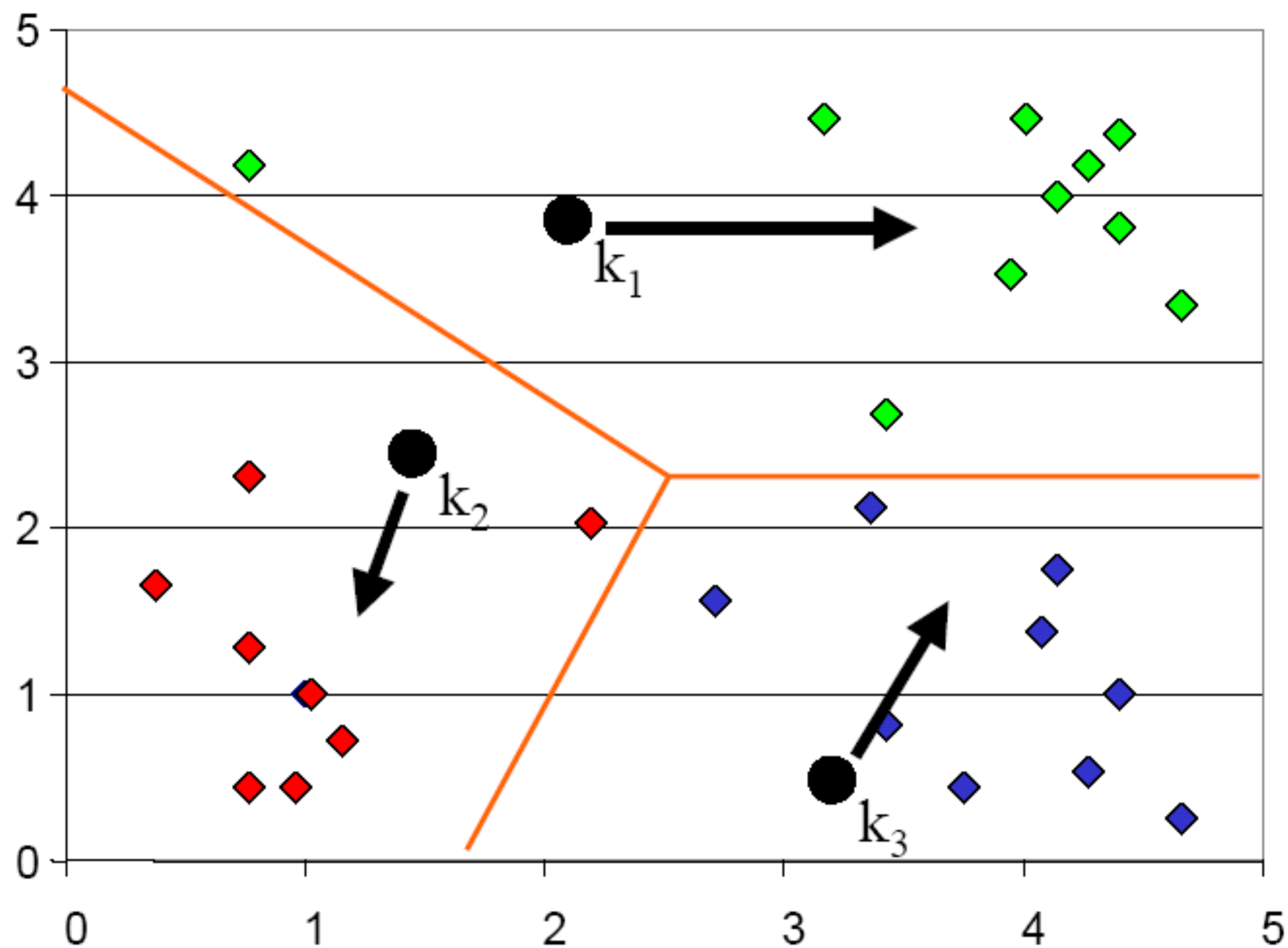
$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$

4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.
5. If none of the *N* objects changed membership in the last iteration, exit. Otherwise go to 3.

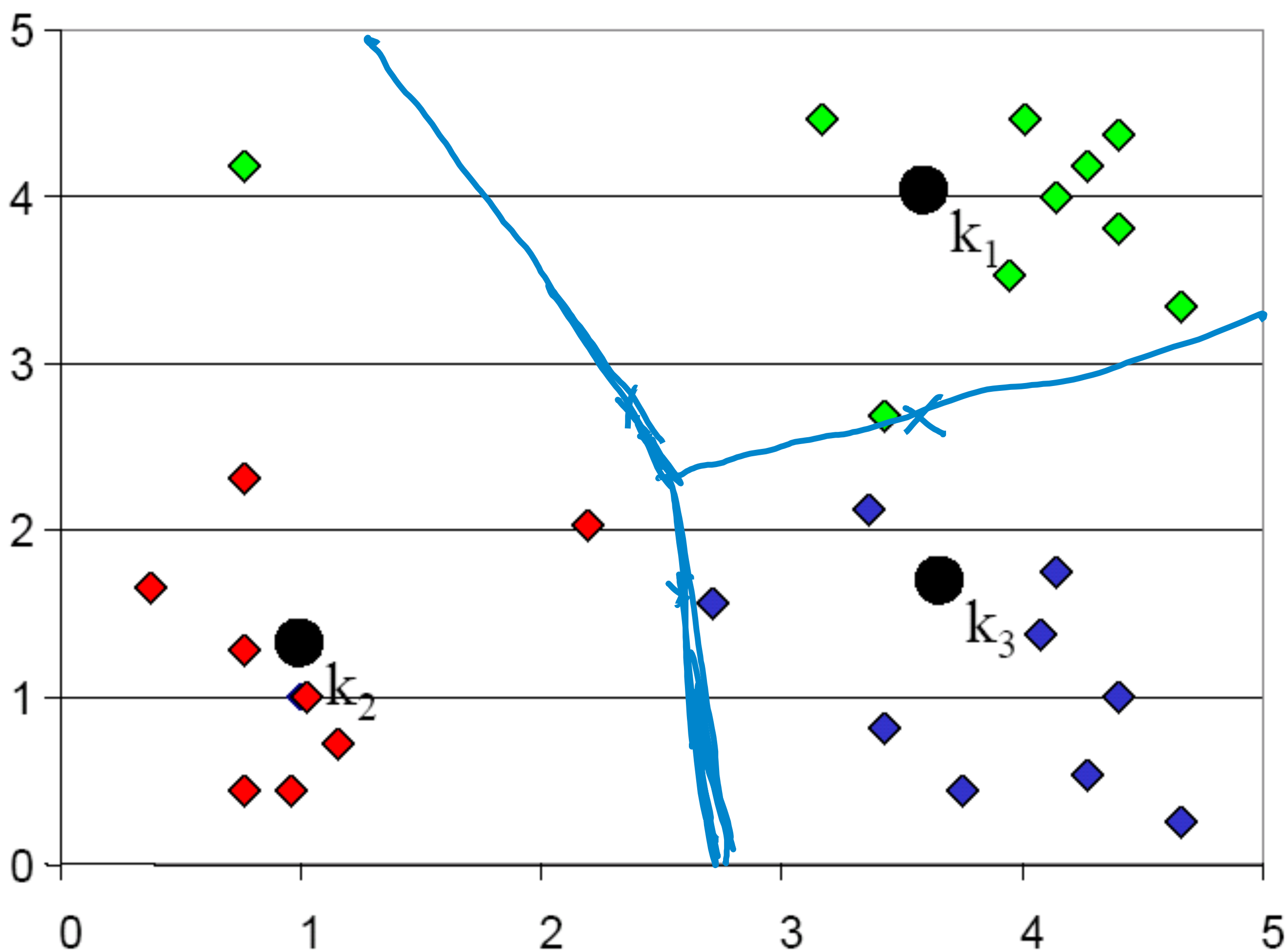
K-means Clustering: Step 1 - random guess of cluster centers



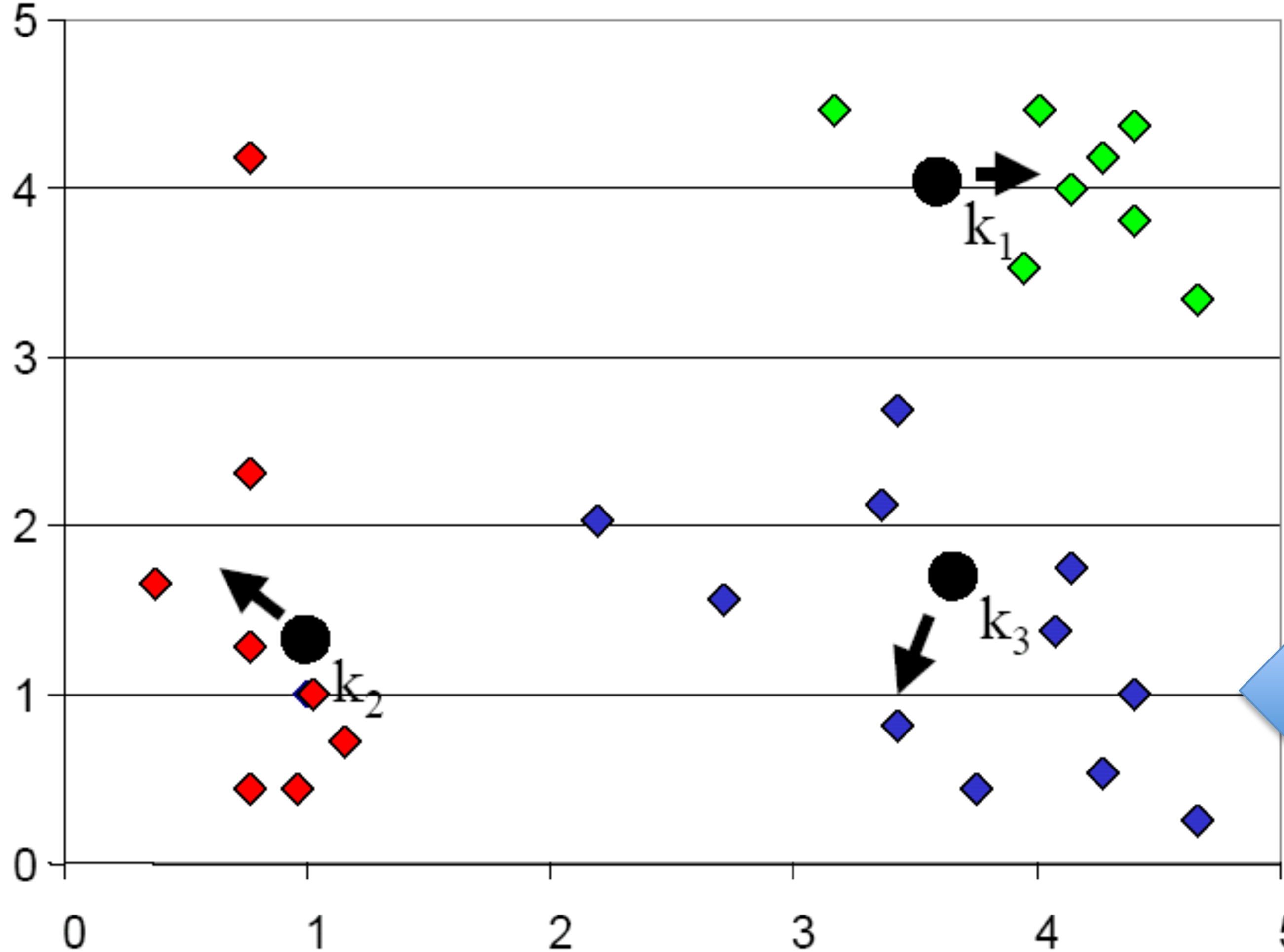
K-means Clustering: Step 2
- Determine the membership of each data points



K-means Clustering: Step 3
- Adjust the cluster centers

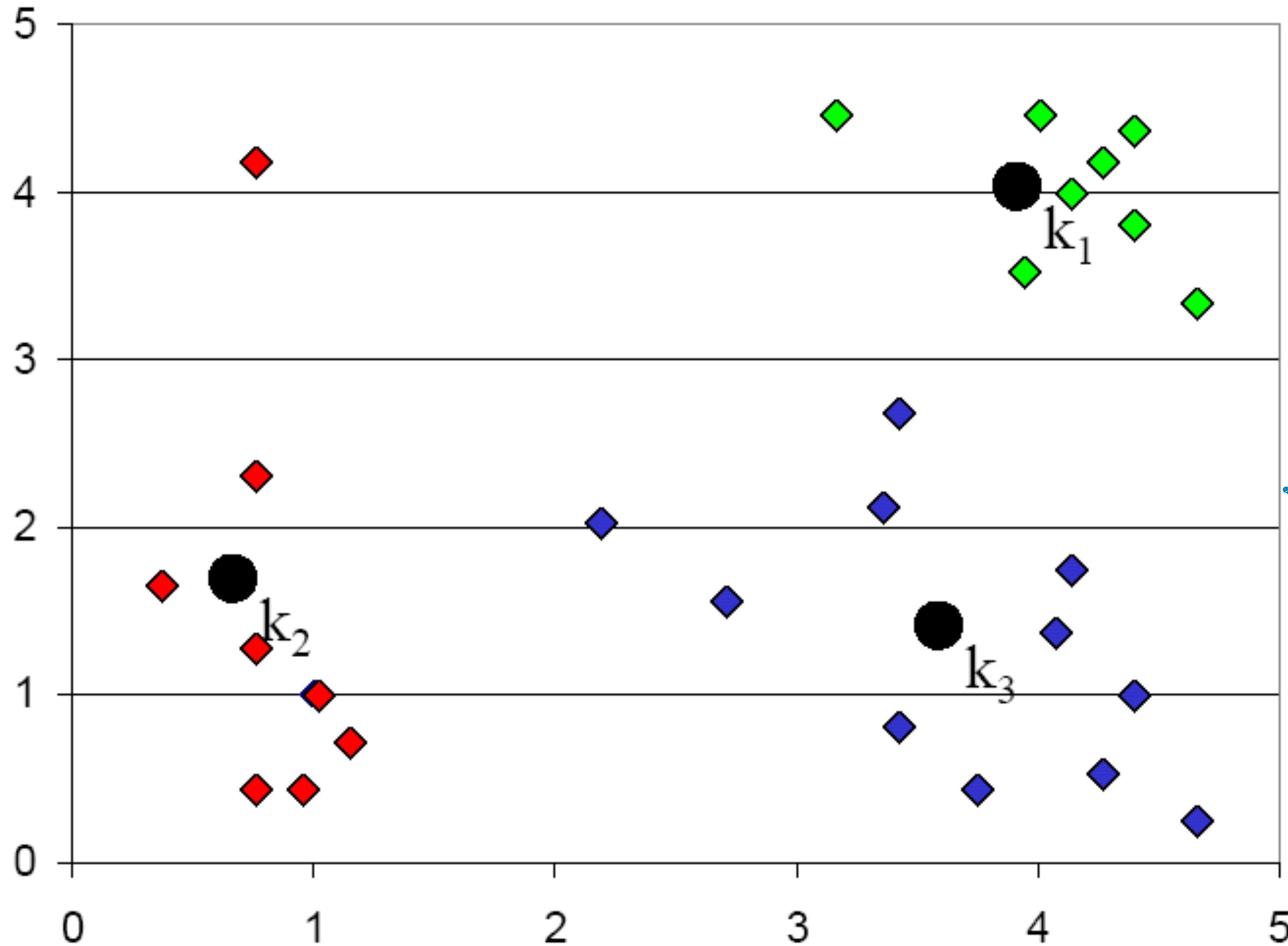


K-means Clustering: Step 4 - redetermine membership



Blue cluster gets more points

K-means Clustering: Step 5 - readjust cluster centers



$x_1, x_2, \ldots, x_n$

x_i

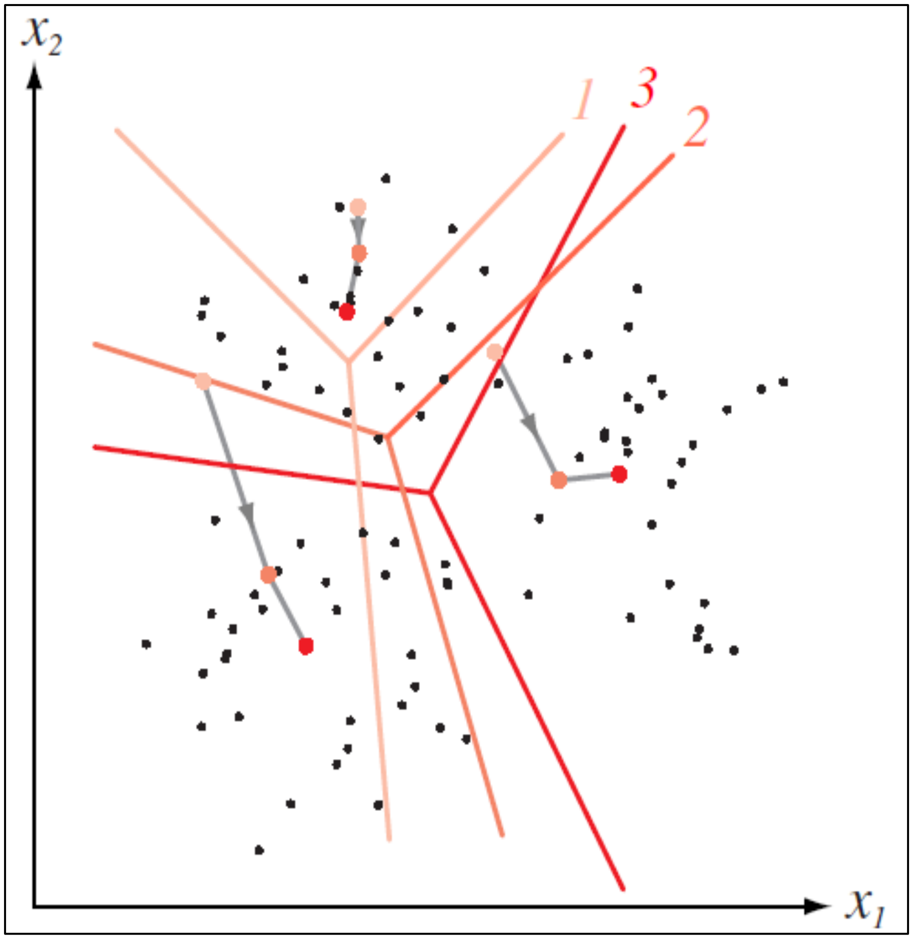
$\{k_1, k_2, k_3\}$

??

$d(k_j, x_i)$

$\underset{j=1,2,\ldots,k}{\operatorname{argmin}}$

How K-means partitions?



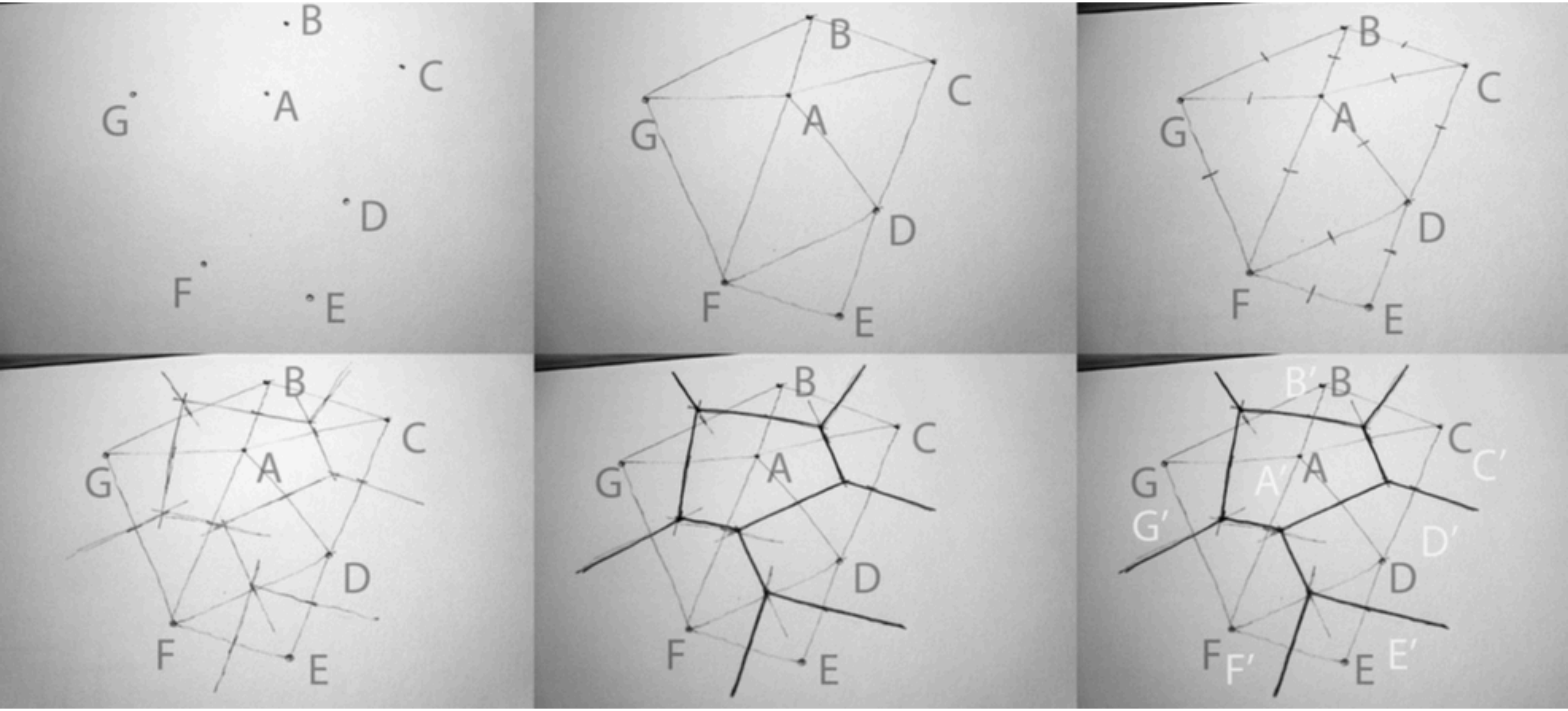
When *K* centroids are set/fixed, they partition the whole data space into *K* mutually exclusive subspaces to form a partition.

A partition amounts to a

[Voronoi Diagram](#)

Changing positions of centroids leads to a new partitioning.

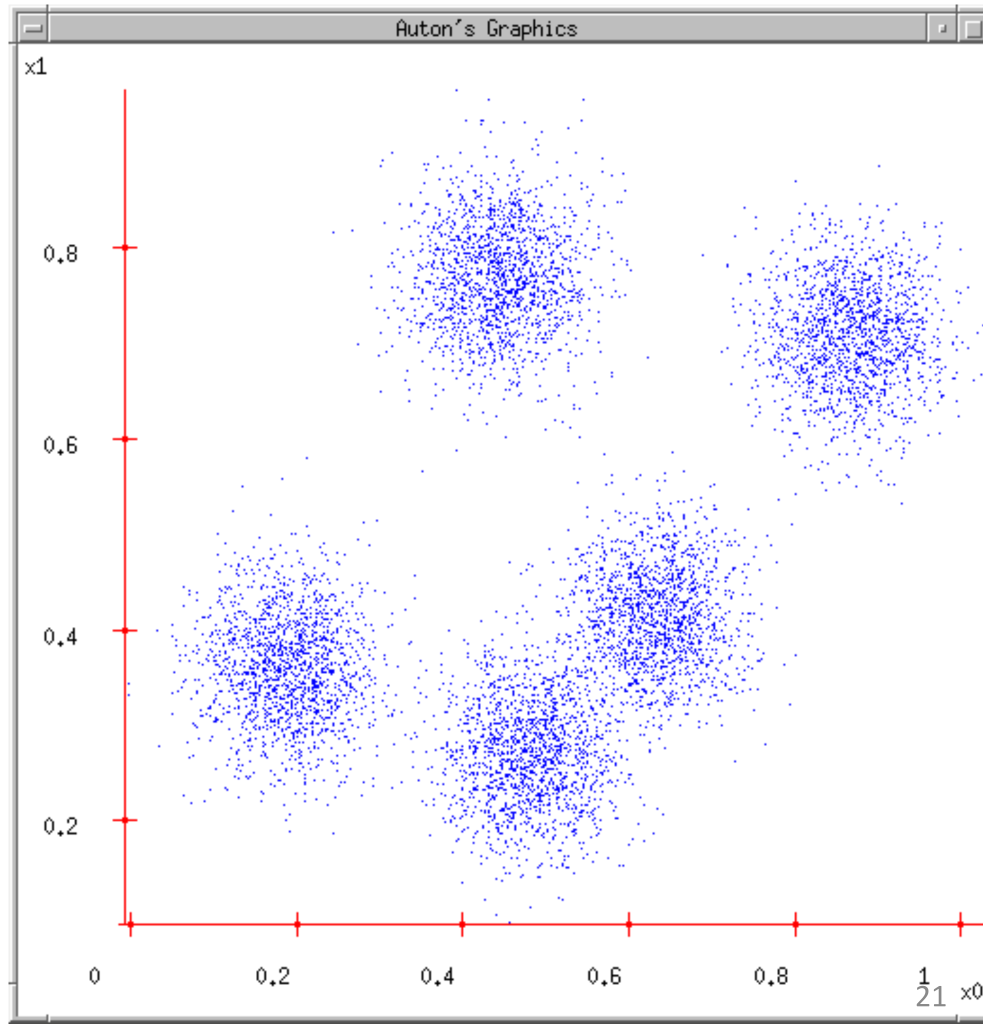
How to draw voronoi diagram



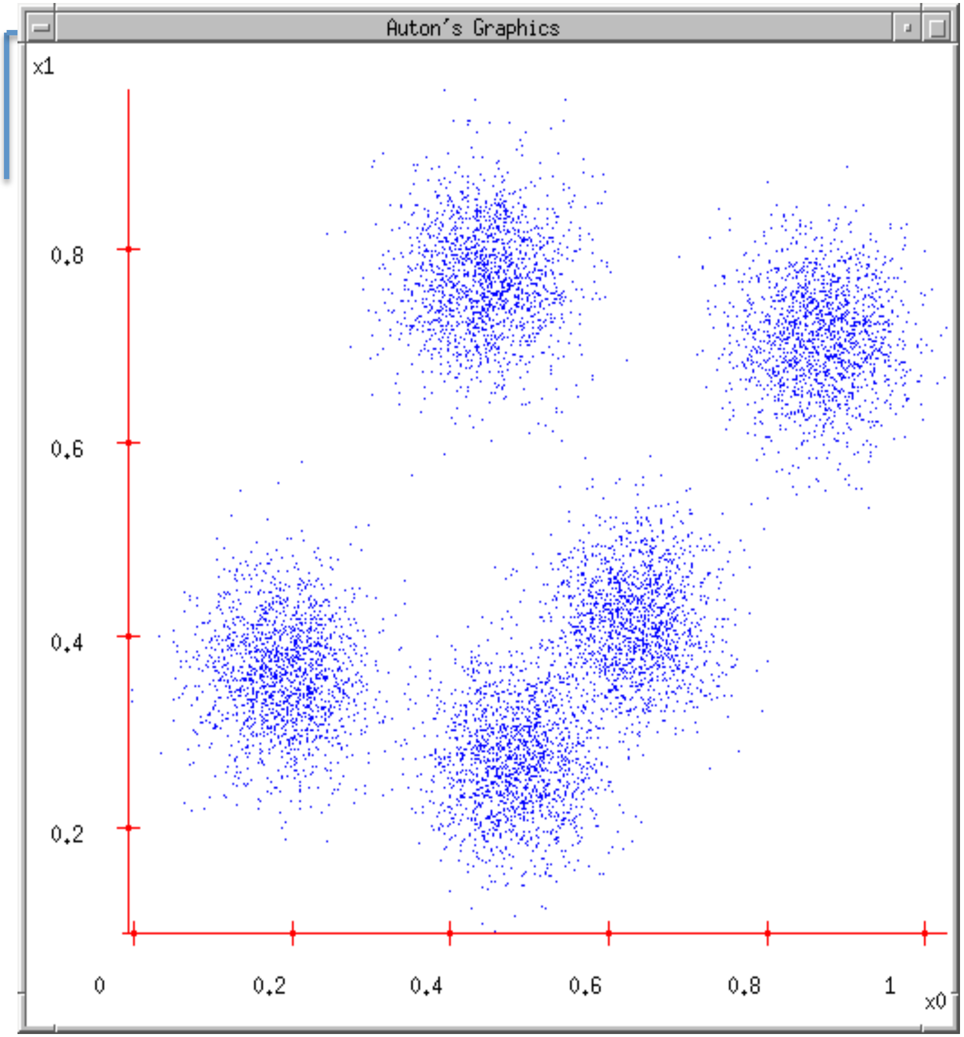
http://765.blogspot.com/2009/09/how-to-draw-voronoi-diagram.html

K-means: another Demo

- K-means
  - Start with a random guess of cluster centers
  - Determine the membership of each data points
  - Adjust the cluster centers

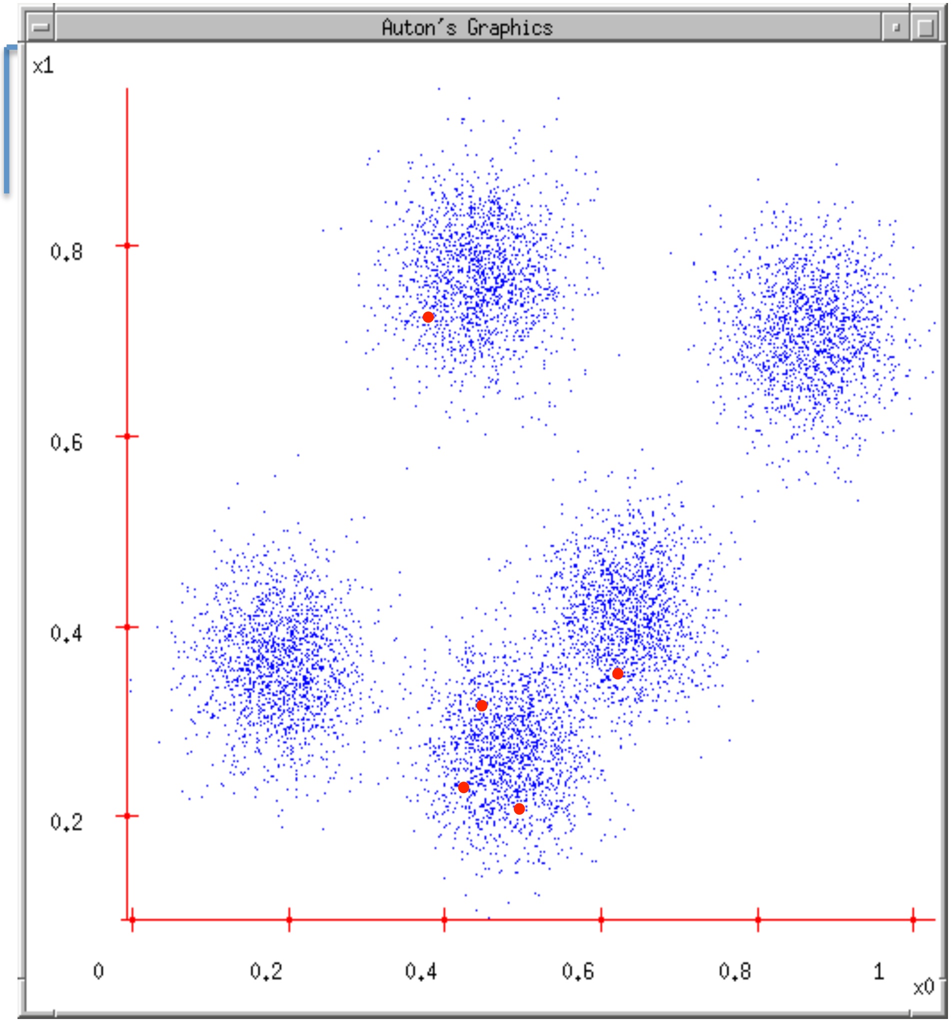


K-means: another Demo



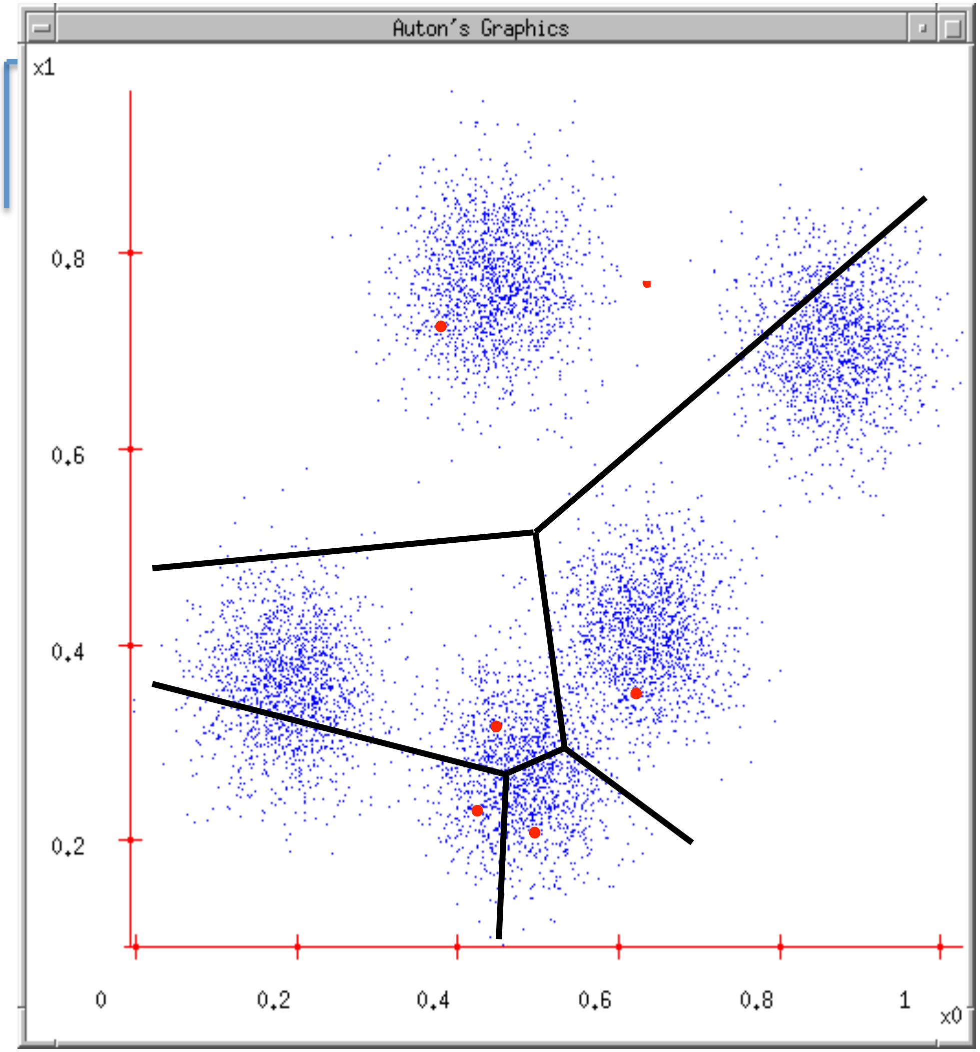
1. User set up the number of clusters they'd like. *(e.g. k=5)*

K-means: another Demo



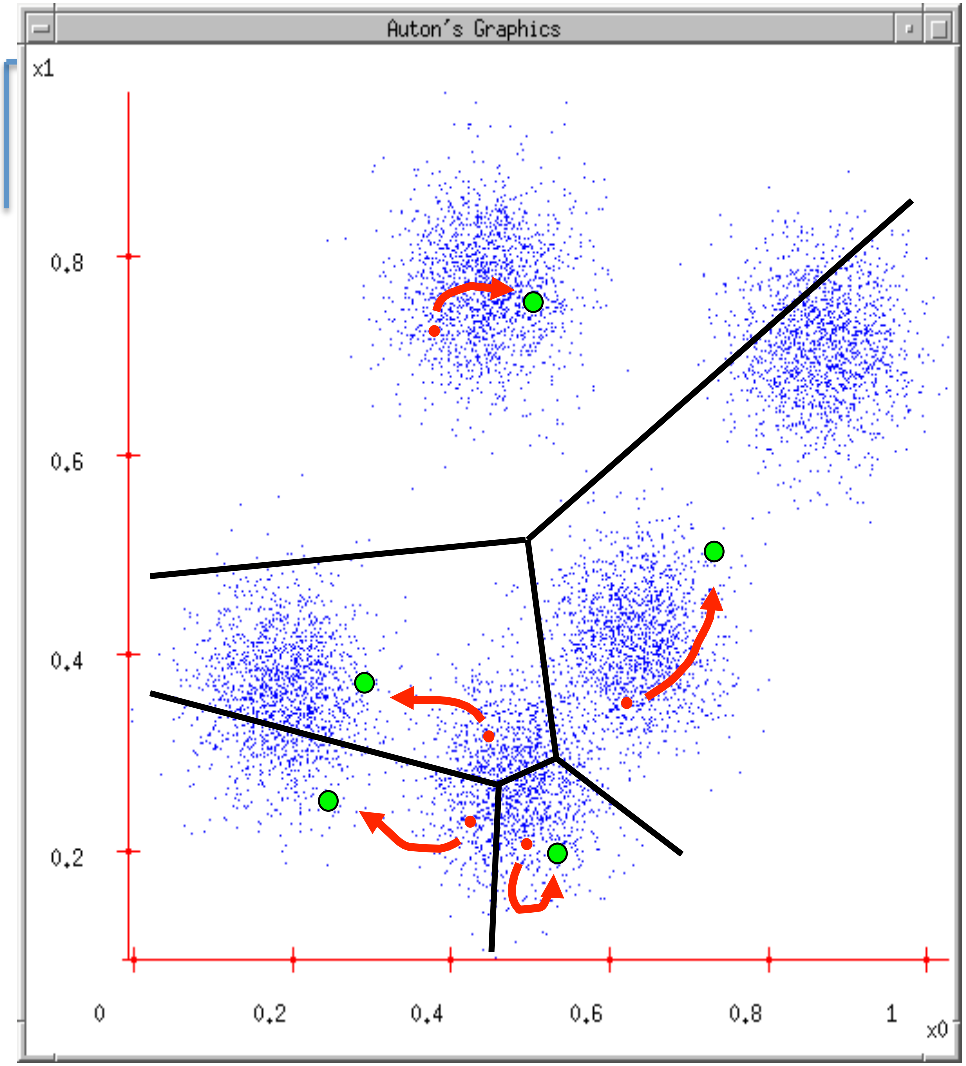
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess K cluster Center locations

K-means: another Demo



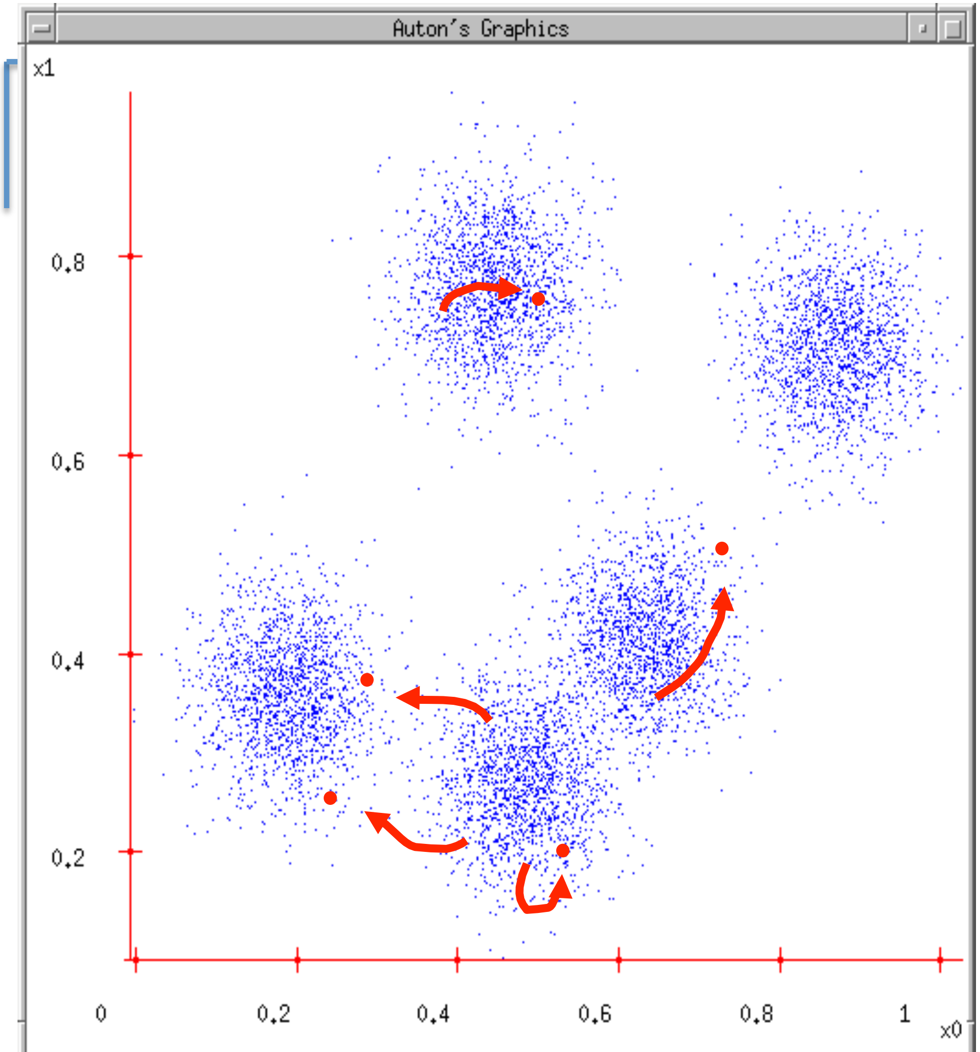
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess *K* cluster Center locations
3. Each data point finds out which Center it's closest to. (Thus each Center "owns" a set of data points)

K-means: another Demo



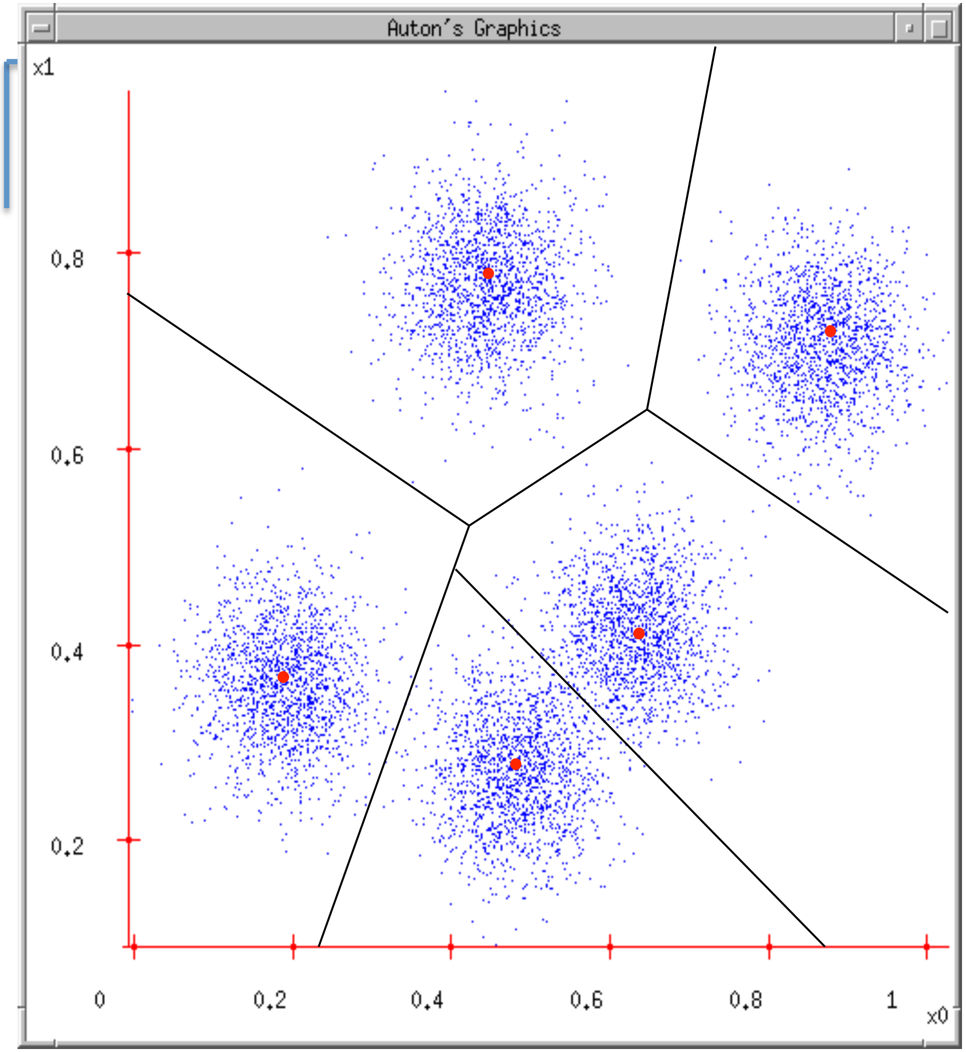
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess *K* cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each Center "owns" a set of data points)
4. Each centre finds the centroid of the points it owns

K-means: another Demo



1. User set up the number of clusters they'd like. (e.g. *K=5*)
2. Randomly guess *K* cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each centre "owns" a set of data points)
4. Each centre finds the centroid of the points it owns
5. …and jumps there

K-means: another Demo

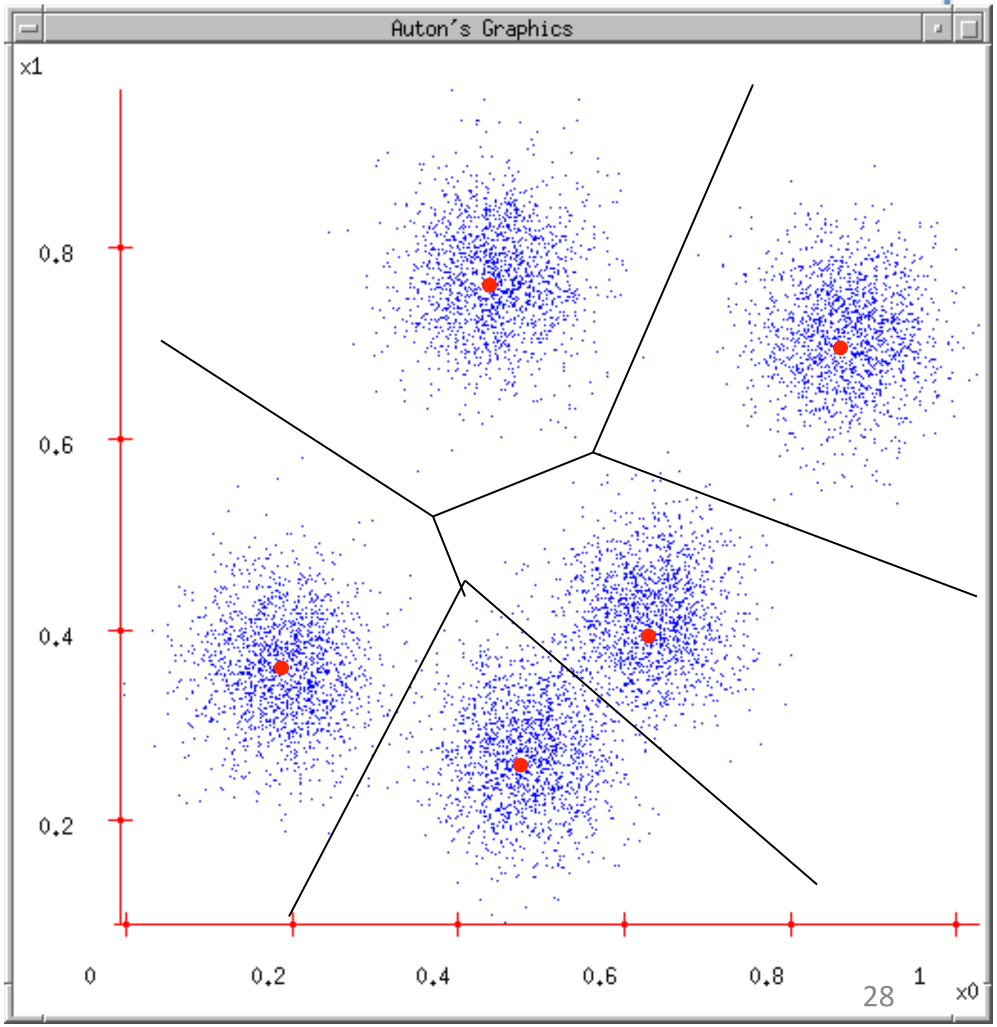


1. User set up the number of clusters they'd like. (e.g. *K=5*)
2. Randomly guess *K* cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each centre "owns" a set of data points)
4. Each centre finds the centroid of the points it owns
5. …and jumps there
6. …Repeat until terminated!

K-means

1. Ask user how many clusters they'd like. (e.g. k=5)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns

Any Computational Problem?



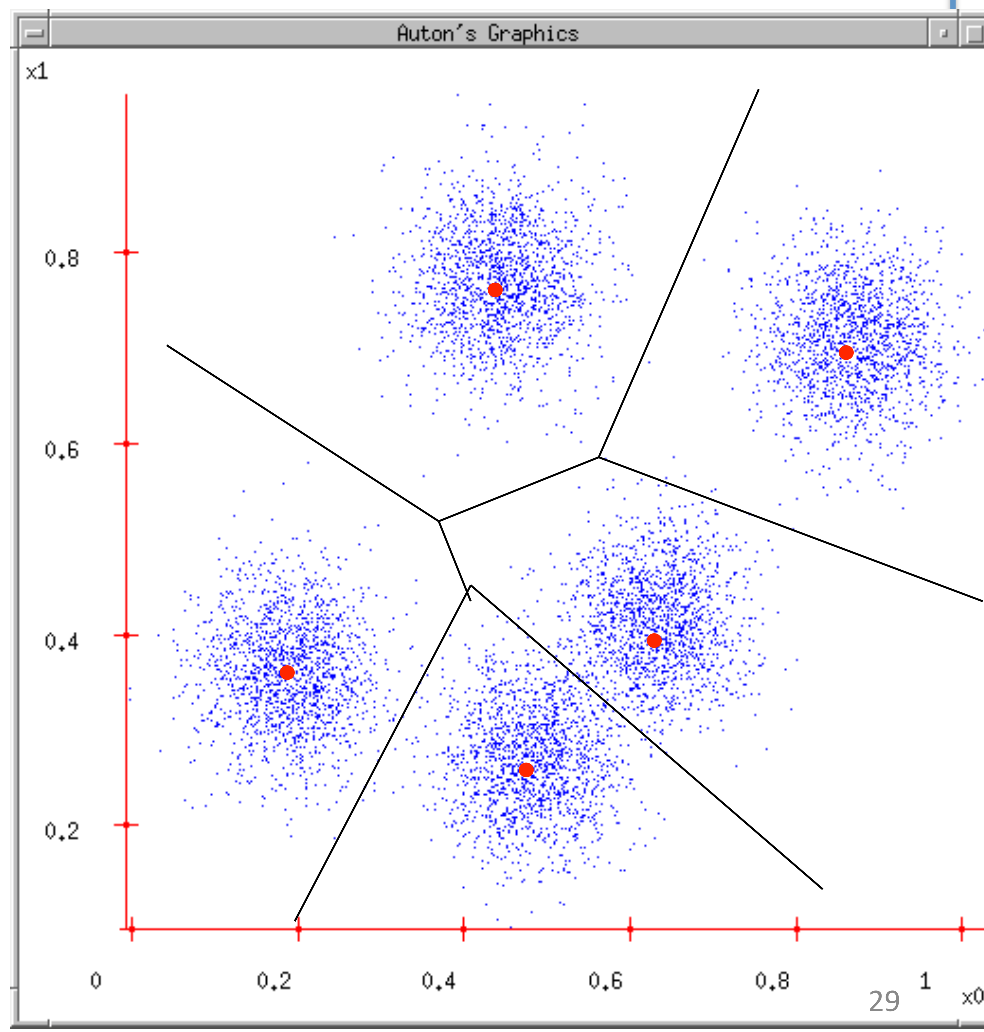
K-means

1. Ask user how many clusters

Computational Complexity: O(n) where n is the number of points?

2. ...ly guess k cluster Cente... ...cations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns

Any Computational Problem?



Time Complexity

- Computing distance between two objs is O(p) where p is the dimensionality of the vectors.

Step 3

- Reassigning clusters: O(Knp) distance computations,

Step 2

- Computing centroids: Each obj gets added once to some centroid: O(np).
- Assume these two steps are each done once for l iterations: O($lKnp$).

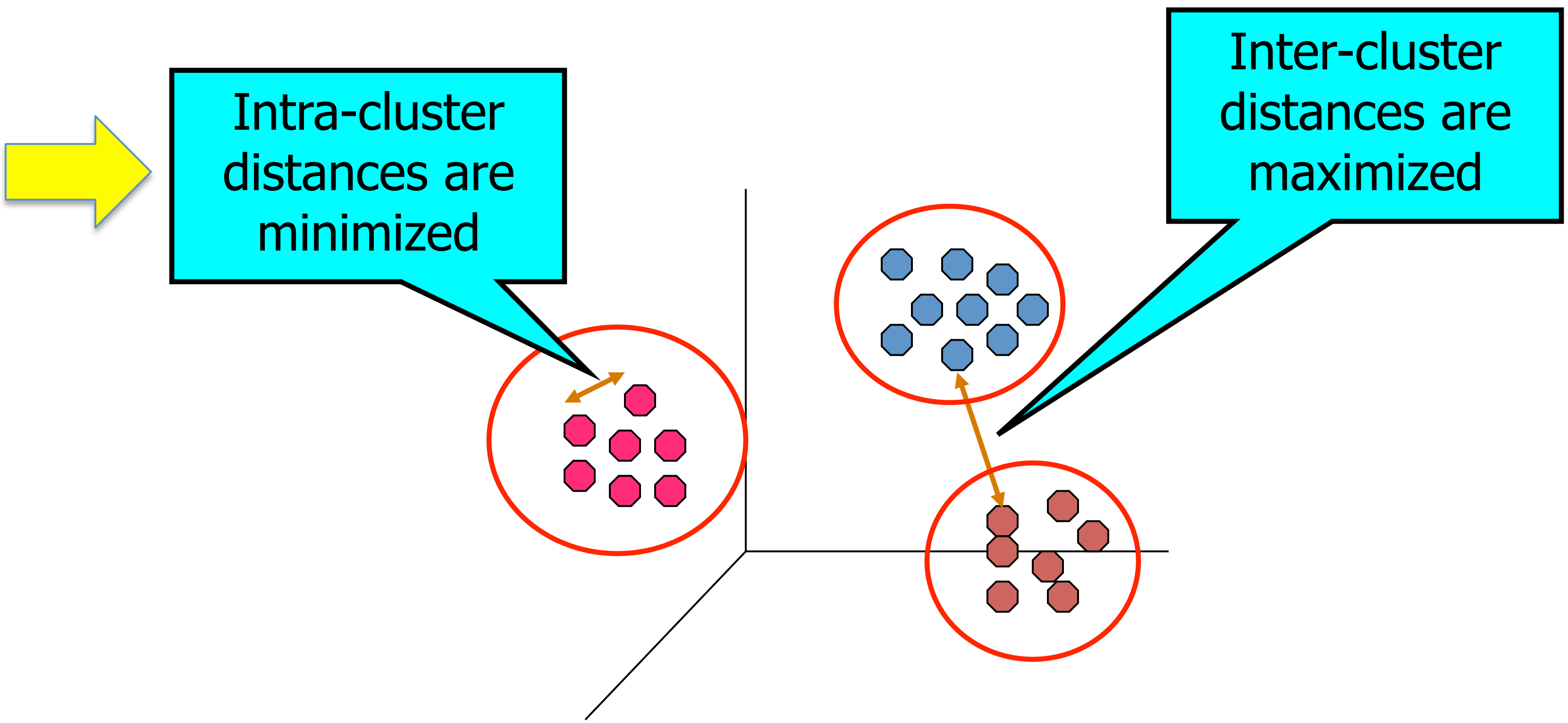
O(n^3) Hierarchical

Roadmap: clustering

- Definition of "groupness"
- Definition of "similarity/distance"
- Representation for objects
- How many clusters?
- Clustering Algorithms
  - Partitional algorithms
  - Hierarchical algorithms
- Formal foundation and convergence

How to Find good Clustering?

- Find groups (clusters) of data points such that data points in a group will be similar (or related) to one another and different from (or unrelated to) the data points in other groups



How to Find good Clustering? E.g.

- Minimize the sum of distance within clusters

$j = 1, 2, \cdots, K$

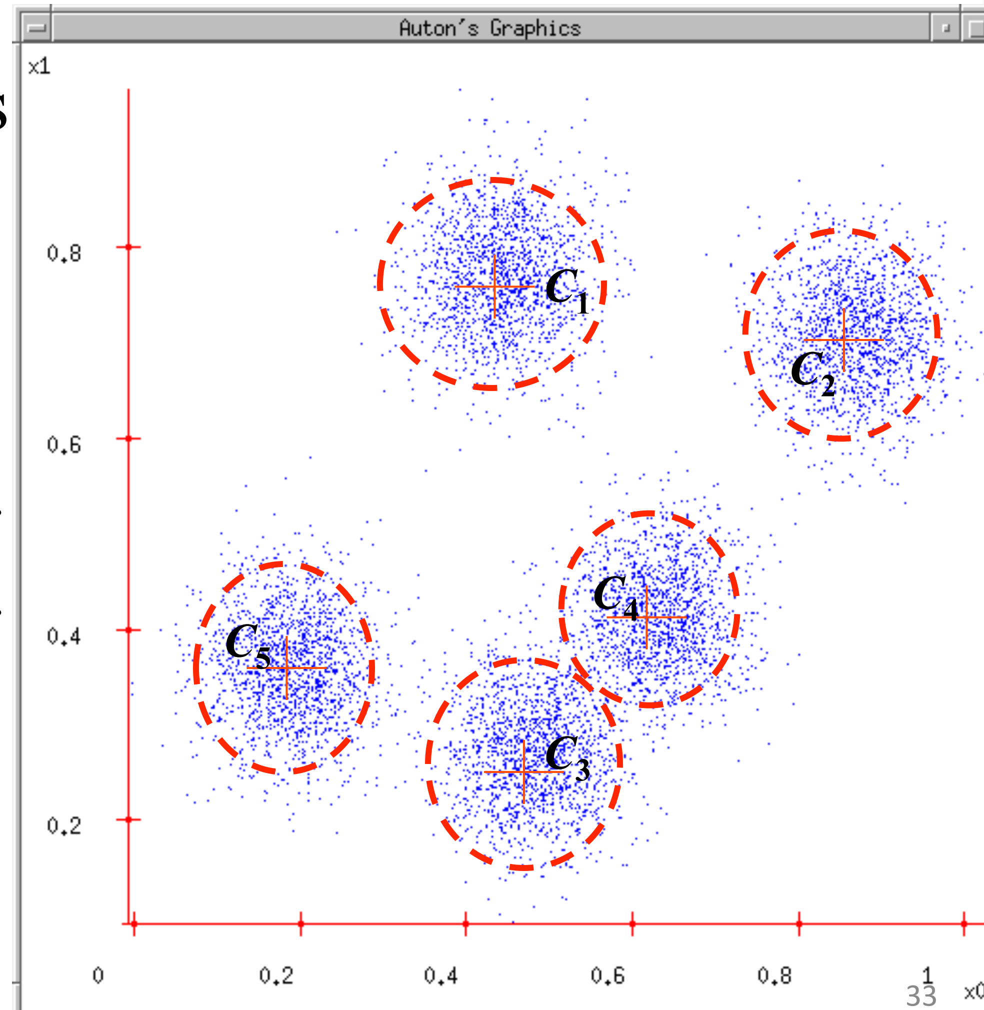
$$\underset{\{\vec{C}_j, m_{i,j}\}}{\arg\min} \sum_{j=1}^{5} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j \right)^2$$

$$m_{i,j} = \begin{cases} 1 & \vec{x}_i \in \text{ the j-th cluster} \\ 0 & \vec{x}_i \notin \text{ the j-th cluster} \end{cases}$$

$i = 1, \cdots, n$

$$\sum_{j=1}^{5} m_{i,j} = 1$$

$\rightarrow$ any $\vec{x}_i \in$ a single cluster



How to Efficiently Cluster Data?

$$\underset{\{\vec{C}_j, m_{i,j}\}}{\arg\min} \sum_{j=1}^{5} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j \right)^2$$

Memberships $\{m_{i,j}\}$ and centers $\{C_j\}$ are correlated.

Given centers $\{\vec{C}_j\}$, $m_{i,j} = \begin{cases} 1 & j = \underset{k}{\arg\min}(\vec{x}_i - \vec{C}_j)^2 \\ 0 & \text{otherwise} \end{cases}$

Given memberships $\{m_{i,j}\}$, $\vec{C}_j = \dfrac{\sum_{i=1}^{n} m_{i,j} \vec{x}_i}{\sum_{i=1}^{n} m_{i,j}}$ $= \dfrac{\text{sum of points in cluster } j}{\text{\# points in cluster } j}$

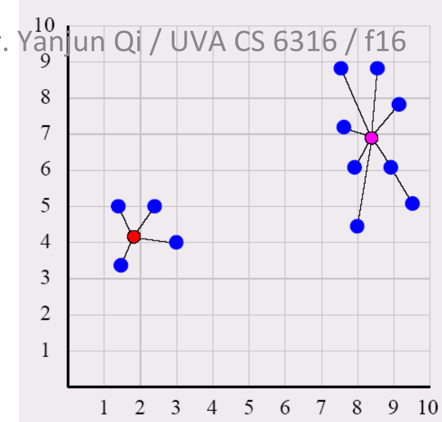
$$\arg\min_{\{\vec{C}_j, m_{i,j}\}} \sum_{j=1}^{\overset{k=5}{5}} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j\right)^2$$

$\rightarrow$ when given $\{m_{ij}\}$, $loss(\vec{C}_j) = \sum_{j=1}^{k} \sum_{i=1}^{n} m_{ij} (\vec{x}_i - \vec{C}_j)^2$

$$\frac{\partial\, loss(\vec{C}_j)}{\partial \vec{C}_j} = 0 \quad \Rightarrow \quad \vec{C}_j = \frac{\sum_{i=1}^{n} m_{i,j} \vec{x}_i}{\sum_{i=1}^{n} m_{i,j}}$$

$\rightarrow$ When given $\{\vec{C}_j\}$, $\frac{\partial\, loss(m_{ij})}{\partial m_{ij}} = 0 \Rightarrow$

$$\Rightarrow \quad m_{i,j} = \begin{cases} 1 & j = \arg\min_{k} (\vec{x}_i - \vec{C}_j)^2 \\ 0 & \text{otherwise} \end{cases}$$

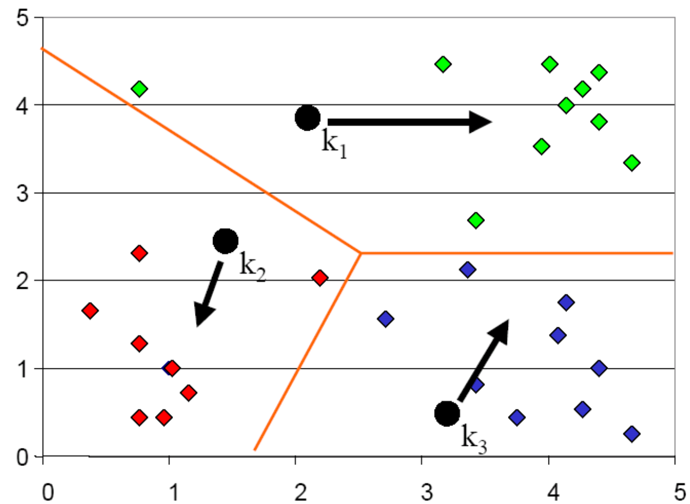


Convergence

- Why should the K-means algorithm ever reach a fixed point?
  - A state in which clusters don't change.
- K-means is a special case of a general procedure known as the Expectation Maximization (EM) algorithm.
  - EM is known to converge.
  - Number of iterations could be large.
- Cluster goodness measure / Loss function to minimize
  - sum of squared distances from cluster centroid:
- Reassignment monotonically decreases the goodness measure since each vector is assigned to the closest centroid.

Seed Choice

- Results can vary based on random seed selection.



K

$C_1, C_2, \ldots, C_K$

- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.
  - Select good seeds using a heuristic (e.g., sample least similar to any existing mean)
  - Try out multiple starting points (very important!!!)
  - Initialize with the results of another method.

(2) K-means Clustering

Task	Clustering
↓	↓
Representation	n/a
↓	↓
Score Function	Sum-of-square distance to centroid
↓	↓
Search/Optimization	K-means algorithm
↓	↓
Models, Parameters	Cluster membership & centroid

$i=1,\cdots,n$
$j=1,\cdots,k$

$\{m_{ij}\}$

$\{C_j\}$

Roadmap: clustering

- Definition of "groupness"
- Definition of "similarity/distance"
- Representation for objects
- How many clusters?
- Clustering Algorithms
  - Partitional algorithms
  - Hierarchical algorithms
- Formal foundation and convergence

Other partitioning Methods

$C_j \in$ train Set

- Partitioning around medoids (PAM): instead of averages, use multidim medians as centroids (cluster "prototypes"). Dudoit and Freedland (2002).
- Self-organizing maps (SOM): add an underlying "topology" (neighboring structure on a lattice) that relates cluster centroids to one another. Kohonen (1997), Tamayo et al. (1999).
- Fuzzy k-means: allow for a "gradation" of points between clusters; soft partitions. Gash and Eisen (2002).
- Mixture-based clustering: implemented through an EM (Expectation-Maximization)algorithm. This provides soft partitioning, and allows for modeling of cluster centroids and shapes. (Yeung et al. (2001), McLachlan et al. (2002))

$m_{ij} \in \{1, 0\} \rightarrow [0, 1]$

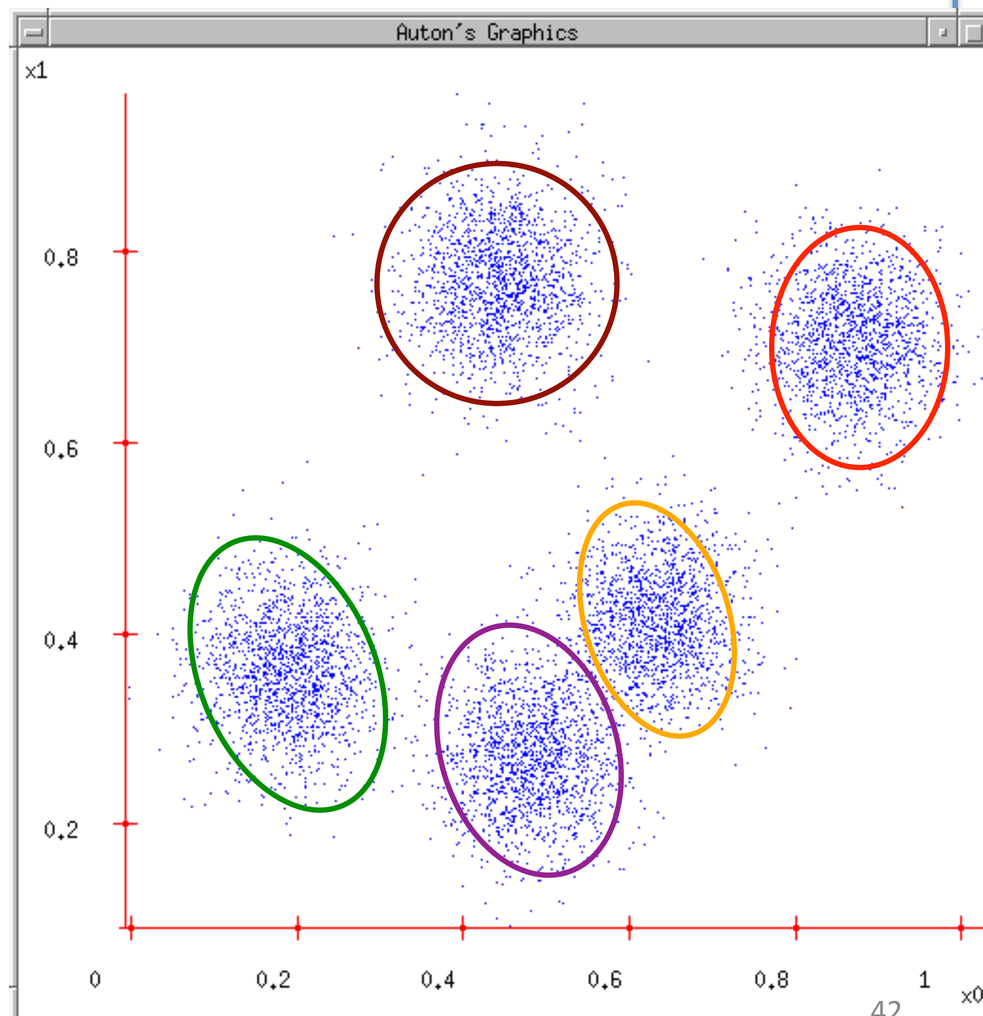
Partitional : Gaussian Mixture Model

- 1. Review of Gaussian Distribution
- 2. GMM for clustering : basic algorithm
- 3. GMM connecting to K-means
- 4. GMM examples
- 5. Applications of GMM
- 6. Problems of GMM and K-means

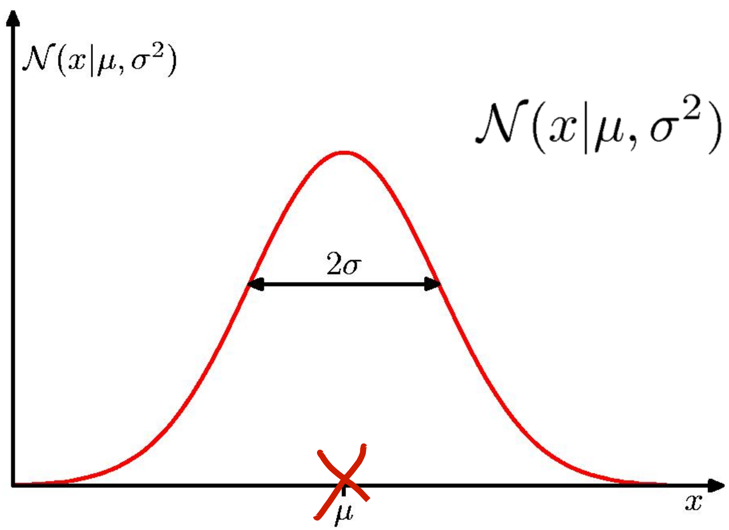
A Gaussian Mixture Model for Clustering

- Assume that data are generated from a mixture of Gaussian distributions
- For each Gaussian distribution
  - Center: μ_i
  - covariance: Σ_i
- For each data point
  - Determine membership

z_{ij} : if x_i belongs to j-th cluster

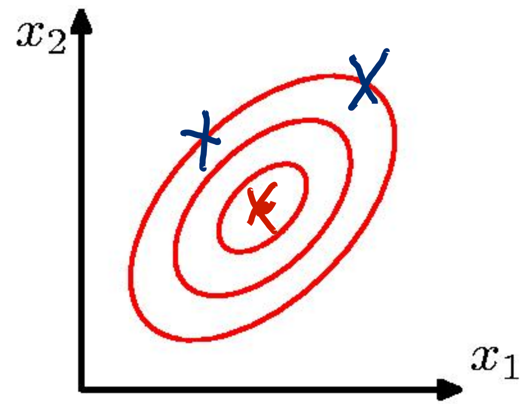


Gaussian Distribution



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

$$X \sim N(\mu, \sigma^2)$$



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

Mean

Covariance Matrix

Courtesy: http://research.microsoft.com/~cmbishop/PRML/index.htm

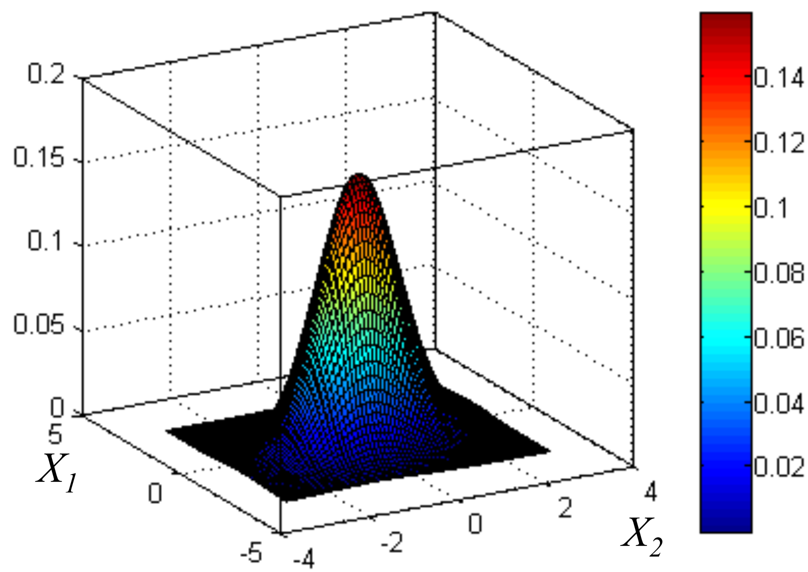
Multivariate Normal (Gaussian) PDFs

The only widely used continuous joint PDF is the multivariate normal (or Gaussian):

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

Where |*| represents determinant

Bivariate normal PDF:



- Mean of normal PDF is at peak value. Contours of equal PDF form ellipses.

- The covariance matrix captures linear dependencies among the variables

Example: the Bivariate Normal distribution

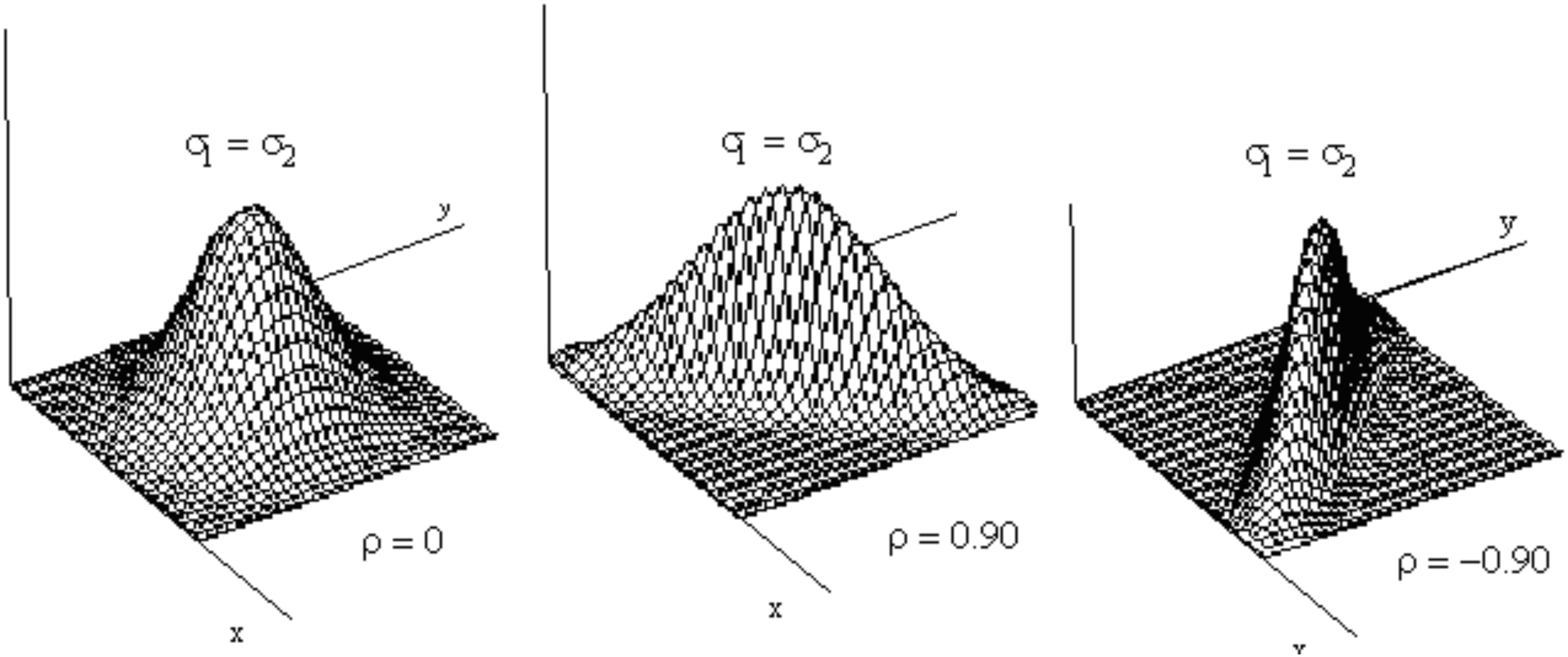
$$f(x_1, x_2) = \frac{1}{(2\pi)|\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1}(\vec{x}-\vec{\mu})}$$

with $\vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}_{2\times 1}$ and

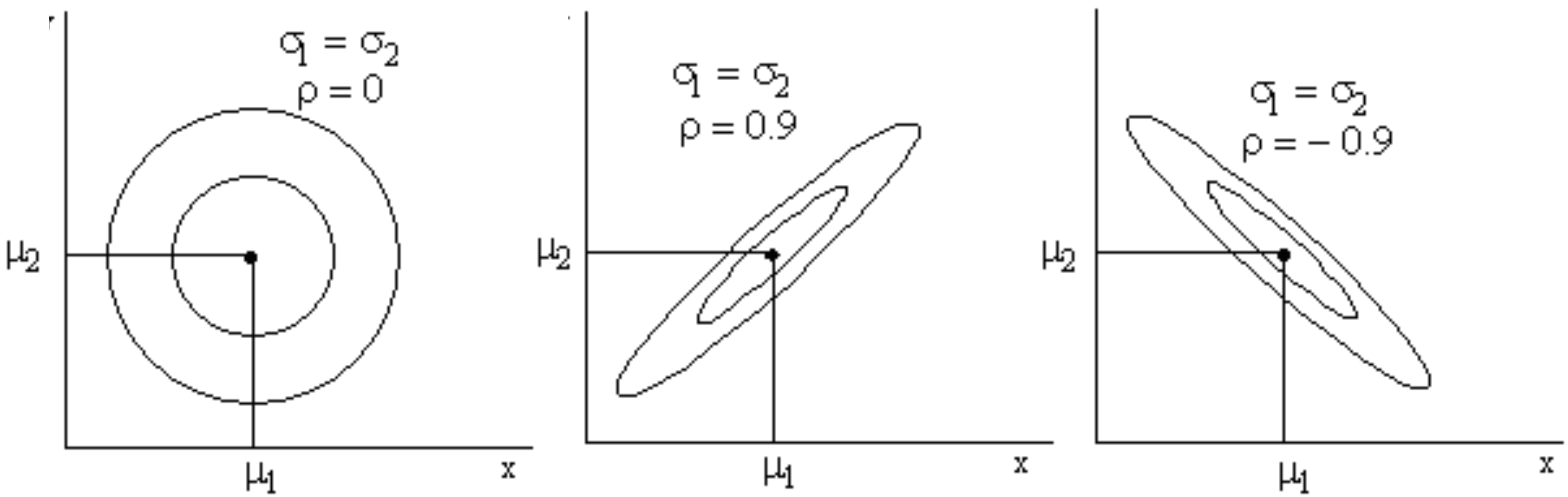
$$\underset{2\times 2}{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix} = \begin{bmatrix} \overset{V(X_1)}{\sigma_1^2} & \overset{Cov(X_1, X_2)}{\underline{\rho\sigma_1\sigma_2}} \\ \underline{\rho\sigma_1\sigma_2} & \underset{V(X_2)}{\sigma_2^2} \end{bmatrix}_{2\times 2}$$

$$|\Sigma| = \sigma_{11}\sigma_{22} - \sigma_{12}^2 = \sigma_1^2\sigma_2^2(1-\rho^2)$$

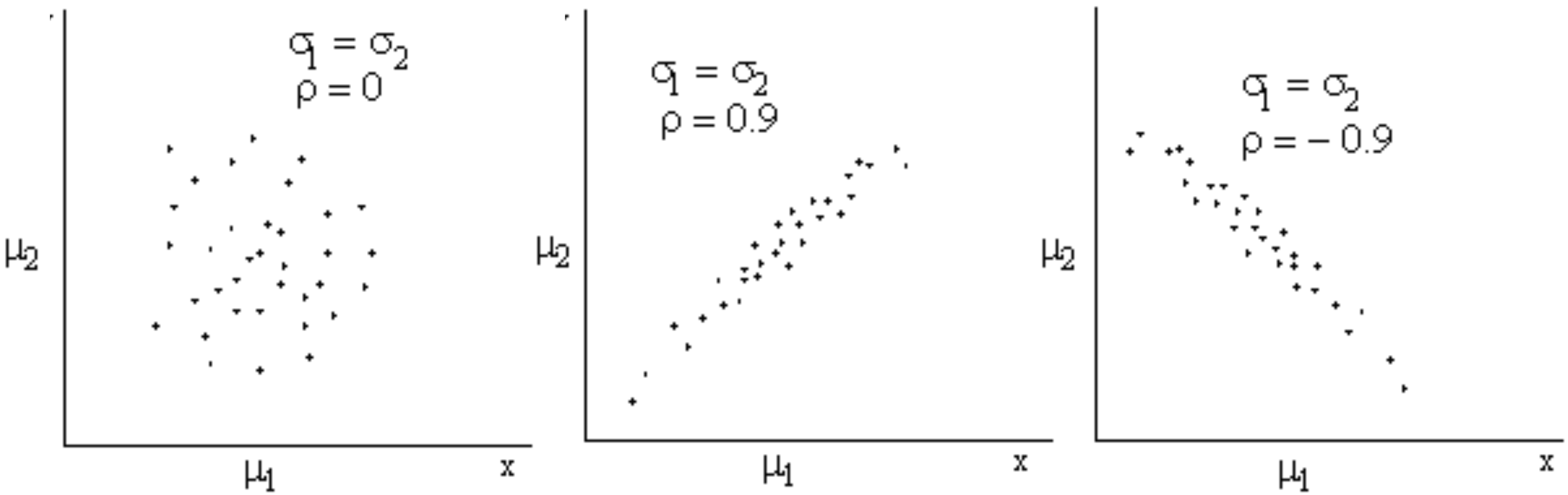
Surface Plots of the bivariate Normal distribution



Contour Plots of the bivariate Normal distribution



Scatter Plots of data from the bivariate Normal distribution



Partitional : Gaussian Mixture Model

- 1. Review of Gaussian Distribution
- 2. GMM for clustering : basic algorithm
- 3. GMM connecting to K-means
- 4. GMM examples
- 5. Applications of GMM
- 6. Problems of GMM and K-means

Learning a Gaussian Mixture (assuming with known shared covariance)

- Probability $p(x = x_i)$

$i = \{1, \dots, n\}$

$$p(x = x_i) = \sum_{\mu_j} p(x = x_i, \mu = \mu_j) = \sum_{\mu_j} p(\mu = \mu_j) p(x = x_i \mid \mu = \mu_j)$$

$j = 1, \dots, K$

[Total low of probability]

[chain rule]

Learning a Gaussian Mixture (assuming with known shared covariance)

- Probability $p(x = x_i)$

$i = \{1, \ldots, n\}$

$$p(x = x_i) = \sum_{\mu_j} p(x = x_i, \mu = \mu_j) = \sum_{\mu_j} p(\mu = \mu_j) p(x = x_i \mid \mu = \mu_j)$$

$j = 1, \ldots, K$

[Total low of probability]

[chain rule]

- Each cluster is model with a Gaussian (here assuming known Σ)

$$p(x = x_i \mid \mu = \mu_j) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{x} - \vec{\mu}_j)^T \Sigma^{-1} (\vec{x} - \vec{\mu}_j)}$$

Assuming

Log-likelihood of Observed Data Samples

- Log-likelihood of data $logp(x_1, x_2, x_3, ..., x_n) =$

$$\sum_i \log p(x = x_i) = \sum_i \log \left[\sum_{\mu_j} p(\mu = \mu_j) \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_j)^T \Sigma^{-1} (\vec{x}-\vec{\mu}_j)} \right]$$

- Apply MLE to find optimal parameters $\left\{ p(\mu = \mu_j), \mu_j \right\}_j$

Learning a Gaussian Mixture

(with known covariance)

$m_{ij} = \{0, 1\}$

[Bayes Rule]

assignment. soft

E-Step

$$E[z_{ij}] = p(\mu = \mu_j \mid x = x_i)$$

$$= \frac{p(x = x_i \mid \mu = \mu_j)\,p(\mu = \mu_j)}{\sum_{s=1}^{k} p(x = x_i \mid \mu = \mu_s)\,p(\mu = \mu_s)}$$

$$= \frac{\cancel{\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}}\, e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_j)^T \Sigma^{-1}(\vec{x}-\vec{\mu}_j)}\; p(\mu = \mu_j)}{\sum_{s=1}^{k} \cancel{\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}}\, e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_s)^T \Sigma^{-1}(\vec{x}-\vec{\mu}_s)}\; p(\mu = \mu_s)}$$

Soft assignment
$p(u = \mu_j \mid x = x_i)$
How x_i belongs in proportion to cluster $\{1, 2, \ldots, k\}$.

vs. m_{ij} Hard assignment in k-means

Learning a Gaussian Mixture

(with known covariance)

K mean => centroid $= \frac{1}{N_j} \sum_{i=1} m_{ij} X_i$

M-Step

$$\mu_j^{(t+1)} \leftarrow \frac{1}{\sum_{i=1}^{n} E[z_{ij}]} \sum_{i=1}^{n} \underbrace{E[z_{ij}]^{(t)}}_{\to [0,1]} x_i$$

$$\sum_{j=1}^{K} E[z_{ij}] = 1$$

$$p(\mu = \mu_j^{(t+1)}) \leftarrow \frac{1}{n} \sum_{i=1}^{n} E[z_{ij}]^{(t)}$$

Covariance: Σ_j (j: 1 to K) will also be derived in the M-step under a full setting

M-step for Estimating unknown Covariance Matrix (more general, details in EM-Extra lecture)

$$\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^{n} E[z_{ij}]^{(t)} (x_i - \mu_j^{(t+1)})(x_i - \mu_j^{(t+1)})^T}{\sum_{i=1}^{n} E[z_{ij}]^{(t)}}$$

for small TrainSet
too many parameters
to estimate

$j = 1, \ldots, K$

$\Sigma_j \Rightarrow O(p^2/2)$

Σ_j $\Rightarrow O(Kp^2/2)$

μ_j $O(Kp+K)$ $p(\mu=\mu_j)$

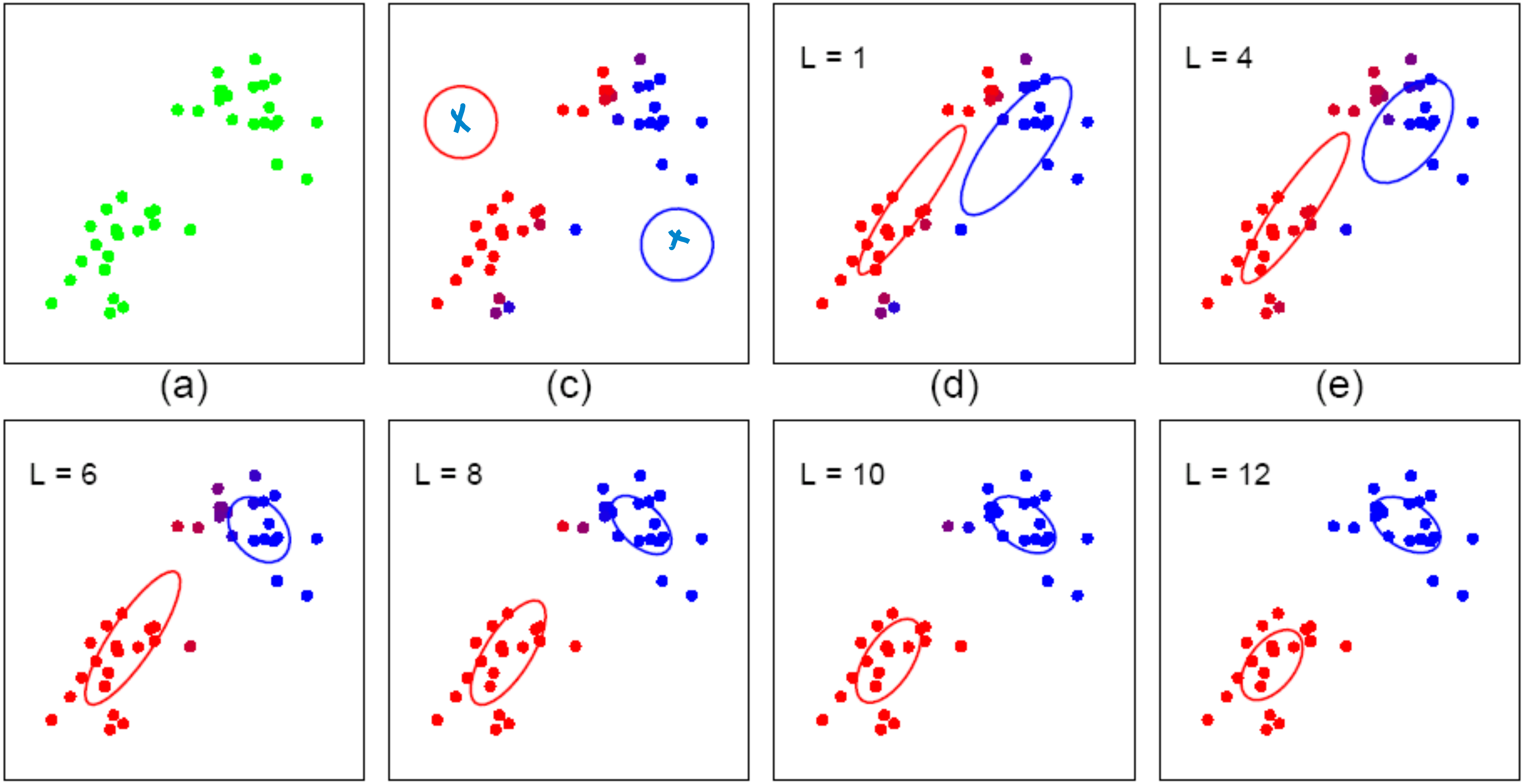
$E[z_{ij}]$ $O(Kn)$

Expectation-Maximization for training GMM

- Start:
  - "Guess" the centroid and covariance for each of the K clusters
  - "Guess" the proportion of clusters, e.g., uniform prob 1/K

- Loop
  - For each point, revising its proportions belonging to each of the K clusters
  - For each cluster, revising both the mean (centroid position) and covariance (shape)

each cluster, revising both the mean (centroid position) and covariance (shape)



(a) (c) (d) (e)

(f) (g) (h) (i)

Detour for HW6: Learning a Gaussian Mixture

(with known covariance and multi-variable and multi-cluster case)

- We assume in HW6, K clusters shared the same known covariance matrix (to reduce the total number of estimated parameters)

  O(kp+k)

- We just use the sample covariance calculating from all samples
  - Full case:

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

  - Diagonal case: to simply use the diagonal of the above sample covariance

E-Step:

Detour for HW6:

Learning a Gaussian Mixture

(with known covariance and multi-variable and multi-cluster case)

$$\boxed{E[z_{ij}] = p(\mu = \mu_j \mid x = x_i) = \frac{p(x = x_i \mid \mu = \mu_j) p(\mu = \mu_j)}{\sum_{s=1}^{k} p(x = x_i \mid \mu = \mu_s) p(\mu = \mu_s)}}$$

O(kn)

$$p(x = x_i | \mu = \mu_j) = \frac{1}{\sqrt{(2\pi)^p det(\Sigma)}} exp(-\frac{1}{2}(x_i - \mu_j)^T \Sigma^{-1}(x_i - \mu_j))$$

$$\mathbb{E}[z_{ij}] = \frac{\cancel{\frac{1}{\sqrt{(2\pi)^p det(\Sigma)}}} exp(-\frac{1}{2}(x_i - \mu_j)^T \Sigma^{-1}(x_i - \mu_j)) p(\mu = \mu_j)}{\sum_{s=1}^{k} \cancel{\frac{1}{\sqrt{(2\pi)^p det(\Sigma)}}} exp(-\frac{1}{2}(x_i - \mu_s)^T \Sigma^{-1}(x_i - \mu_s)) p(\mu = \mu_s)}$$

Detour for HW6: Learning a Gaussian Mixture

(with known covariance and multi-variable and multi-cluster case)

k mean ⇒ centroid $\frac{1}{N_j}\sum_{i=1} m_{ij} X_i$

M-Step

$O(KP) \leftarrow$

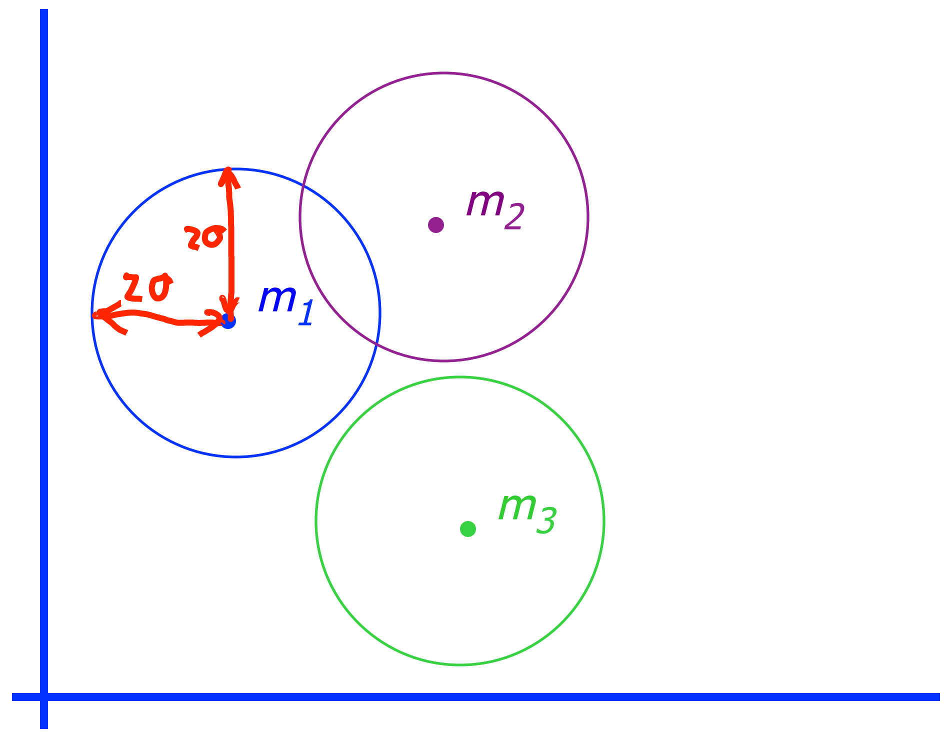
$$\mu_j \leftarrow \frac{1}{\sum_{i=1}^{n} E[z_{ij}]} \sum_{i=1}^{n} E[z_{ij}] x_i$$

$O(K) \leftarrow$

$$\pi_j = p(\mu = \mu_j) \leftarrow \frac{1}{n} \sum_{i=1}^{n} E[z_{ij}]$$

The Simplest GMM assumption

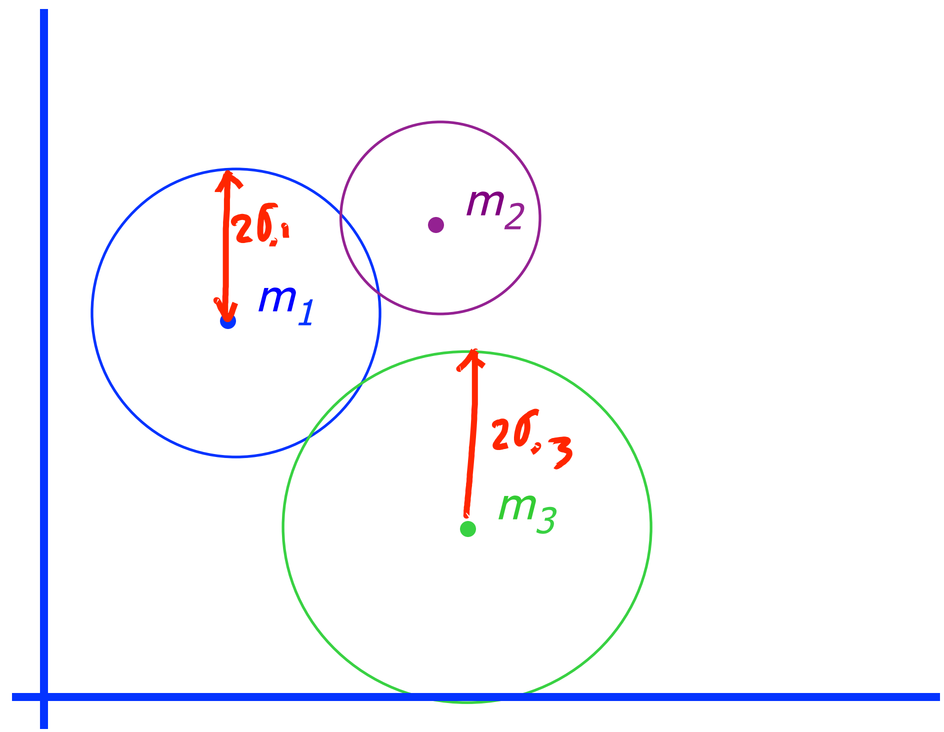
- Each component generates data from a Gaussian with
  - mean μ_i
  - Shared covariance matrix $\sigma^2\boldsymbol{I}$



$$\Sigma_j = \Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

A Simple GMM assumption

- Each component generates data from a Gaussian with
  - mean μ_i
  - Cluster-specific covariance matrix as $\sigma_j^2 \boldsymbol{I}$

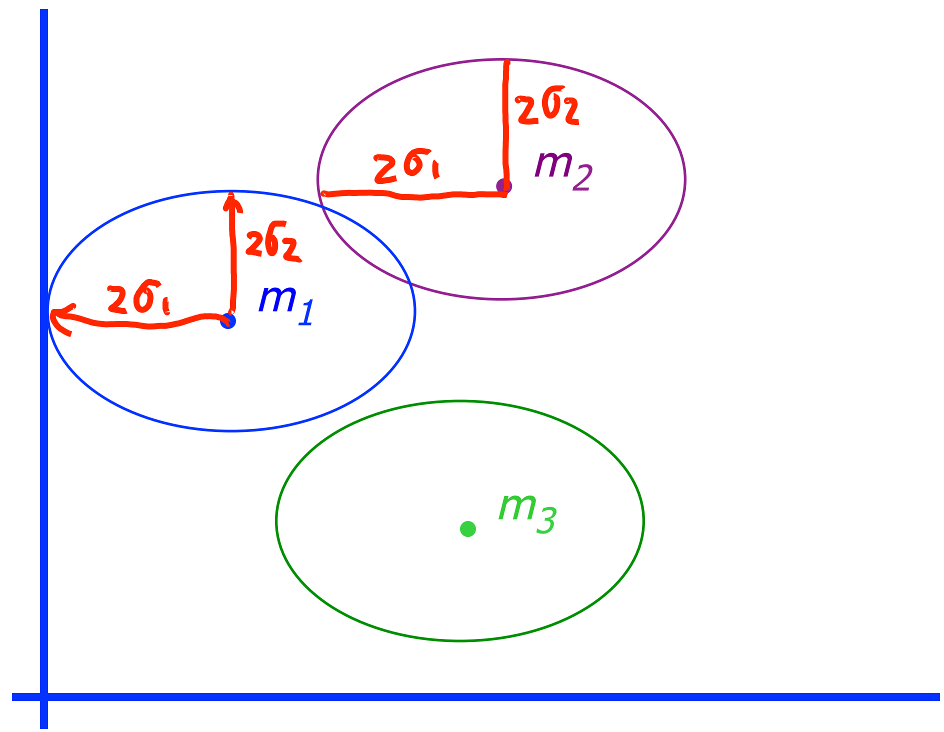


$$\Sigma_j = \sigma_j^2 \boldsymbol{I} = \begin{bmatrix} \sigma_j^2 & 0 \\ 0 & \sigma_j^2 \end{bmatrix}$$

Adapted From Andrew W. Moore

Another Simple GMM assumption

- Each component generates data from a Gaussian with
  - mean μ_i
  - Shared covariance matrix as diagonal matrix

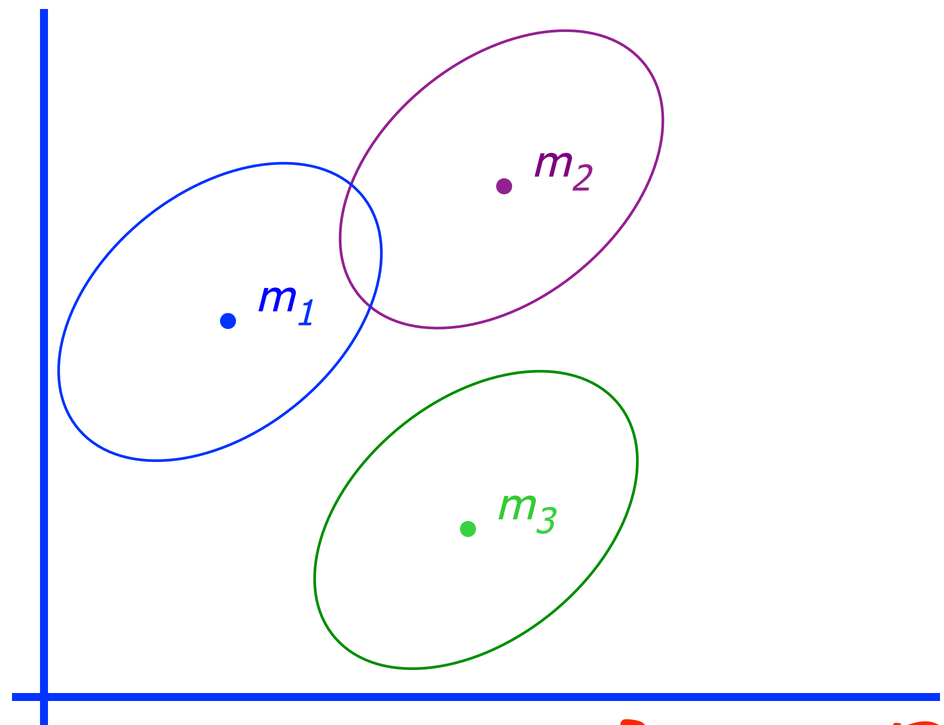


$$\Sigma_j = \Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

Adapted From Andrew W. Moore

A bit More General GMM assumption

- Each component generates data from a Gaussian with
  - mean μ_i
  - Shared covariance matrix as full matrix

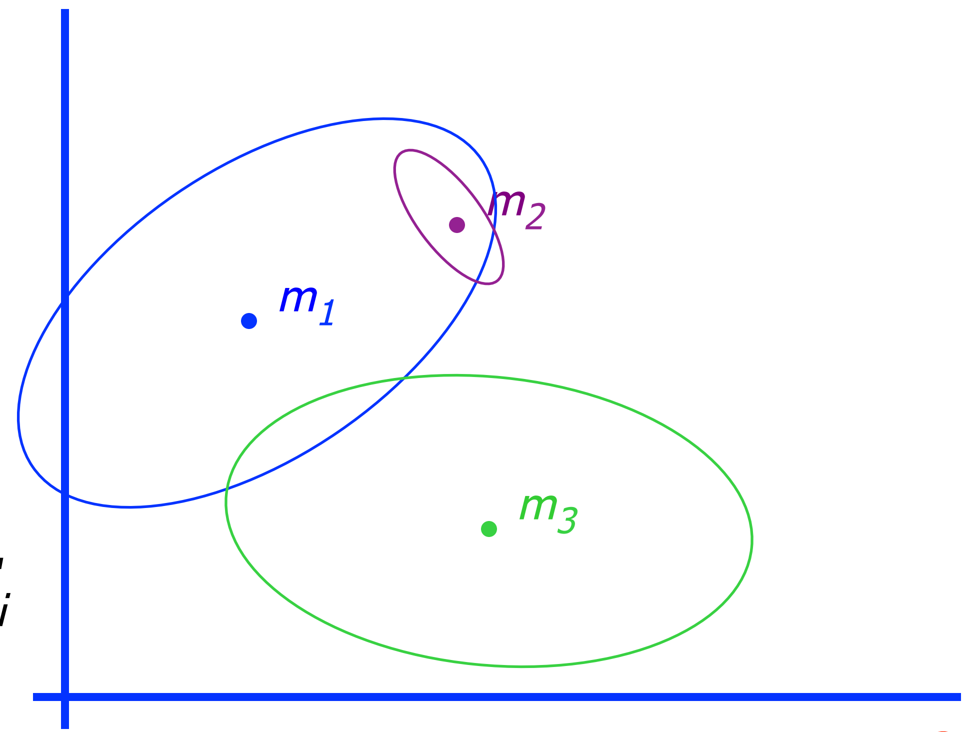


$$\Sigma_j = \Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

Adapted From Andrew W. Moore

The General GMM assumption

- Each component generates data from a Gaussian with
  - mean μ_i
  - covariance matrix Σ_i



$$\Sigma_j = \begin{bmatrix} \sigma_{1j} & \mathrm{cov}_j(x_1, x_2) \\ \mathrm{cov}_j(x_1, x_2) & \sigma_{2j} \end{bmatrix}$$

Partitional : Gaussian Mixture Model

- 1. Review of Gaussian Distribution
- 2. GMM for clustering : basic algorithm
- 3. GMM connecting to K-means
- 4. GMM examples
- 5. Applications of GMM
- 6. Problems of GMM and K-means

Recap: K-means iterative learning

$$\underset{\{\vec{C}_j, m_{i,j}\}}{\arg\min} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j \right)^2$$

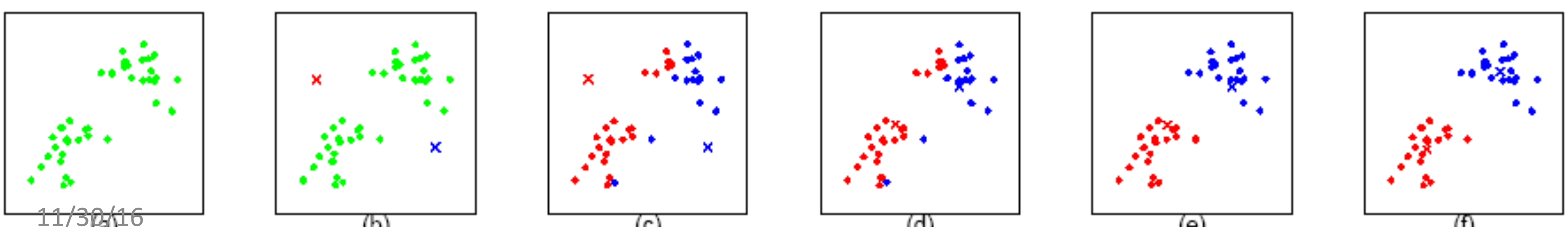
Memberships $\{m_{i,j}\}$ and centers $\{C_j\}$ are correlated.

E-Step Given centers $\{\vec{C}_j\}$, $m_{i,j} = \begin{cases} 1 & j = \underset{k}{\arg\min} (\vec{x}_i - \vec{C}_j)^2 \\ 0 & \text{otherwise} \end{cases}$

M-Step Given memberships $\{m_{i,j}\}$, $\vec{C}_j = \dfrac{\sum_{i=1}^{n} m_{i,j} \vec{x}_i}{\sum_{i=1}^{n} m_{i,j}}$

Compare: K-means

- The EM algorithm for mixtures of Gaussians is like a "soft version" of the K-means algorithm.
- In the K-means "E-step" we do hard assignment:
- In the K-means "M-step" we update the means as the weighted sum of the data, but now the weights are 0 or 1:



K-means: $$\arg\min_{\{\vec{C}_j, m_{i,j}\}} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j \right)^2$$

$$m_{ij} = \begin{cases} 0 \\ 1 \end{cases}$$

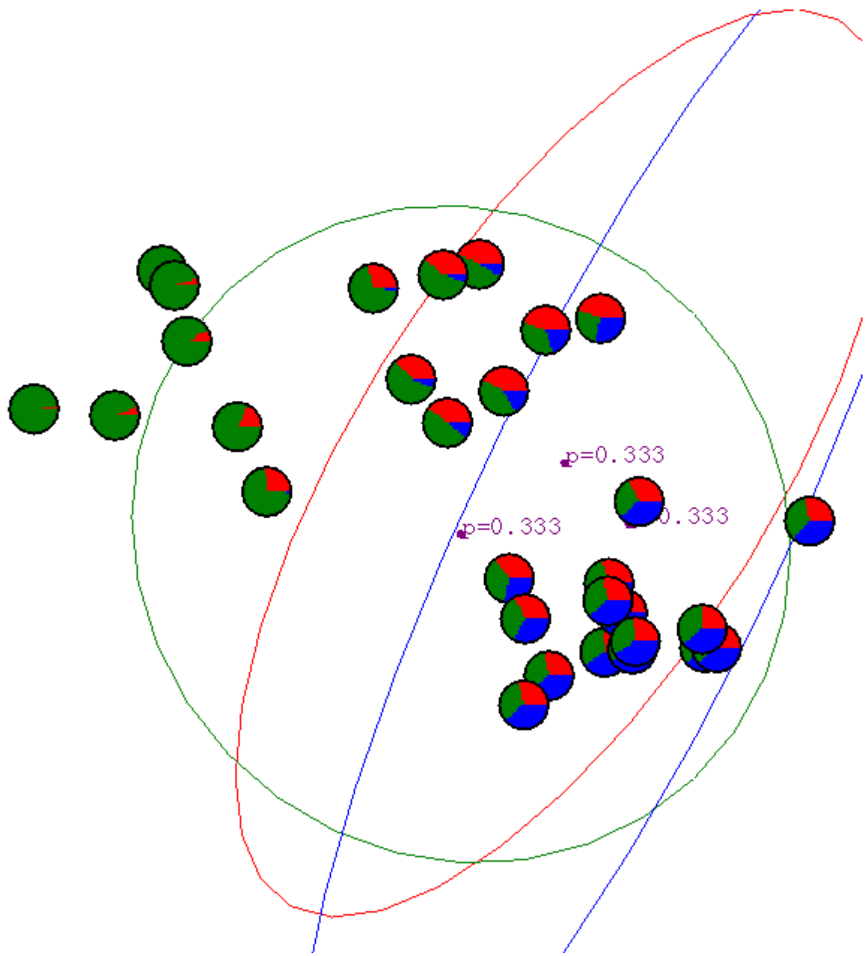
GMM: $$\sum_i \log \prod_{i=1}^{n} p(x = x_i) = \overset{i=1..n}{\sum_i} \log \left[\overset{j=1,..K}{\sum_{\mu_j}} p(\mu = \mu_j) \frac{1}{(2\pi)\left|\Sigma\right|^{1/2}} \mathrm{e}^{-\frac{1}{2}\left(\vec{x}-\vec{\mu}_j\right)^T \Sigma^{-1} \left(\vec{x}-\vec{\mu}_j\right)} \right]$$

K-Mean only detect spherical clusters.
GMM can adjust its self to elliptic shape clusters.

Partitional : Gaussian Mixture Model

- 1. Review of Gaussian Distribution
- 2. GMM for clustering : basic algorithm
- 3. GMM connecting to K-means
- 4. GMM examples
- 5. Applications of GMM
- 6. Problems of GMM and K-means

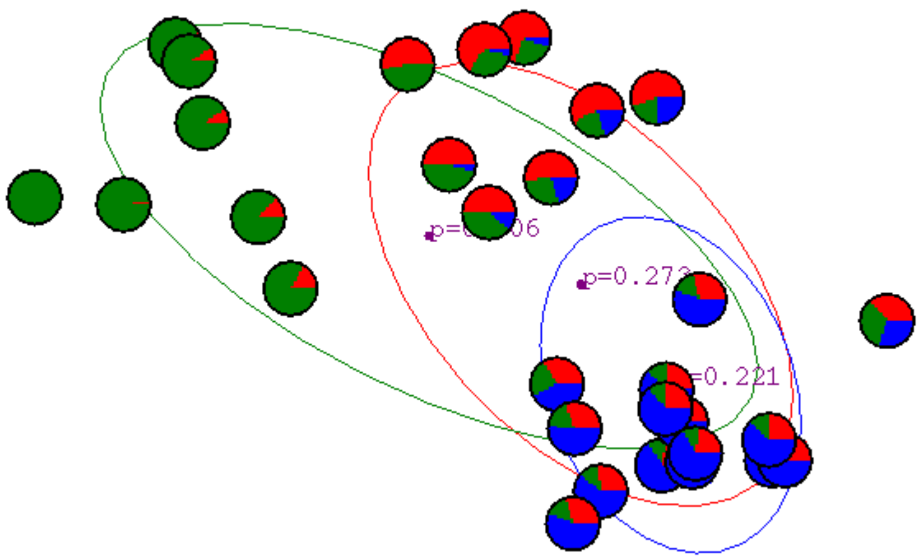
Gaussian Mixture Example: Start



$p(\mu_j | x_i)$

After First Iteration

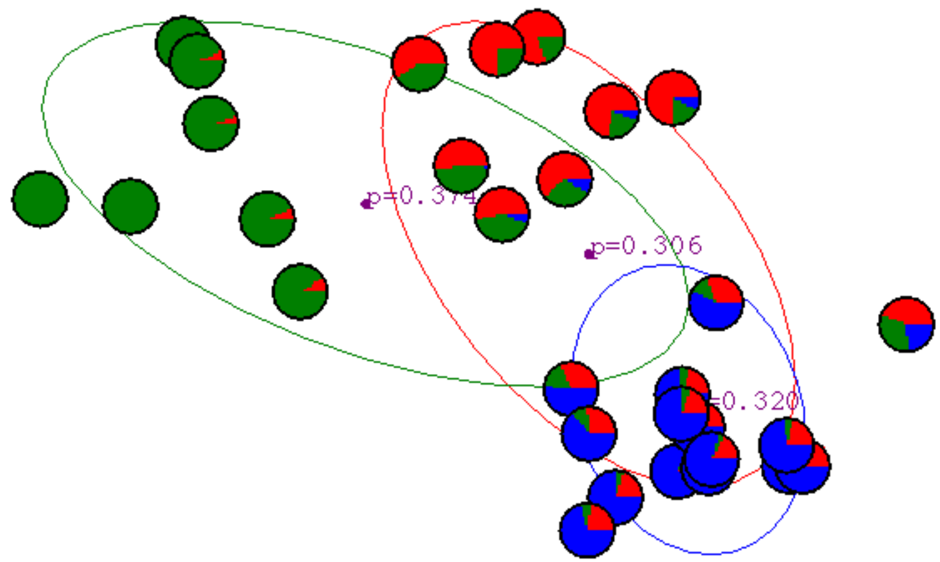
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

After 2nd Iteration

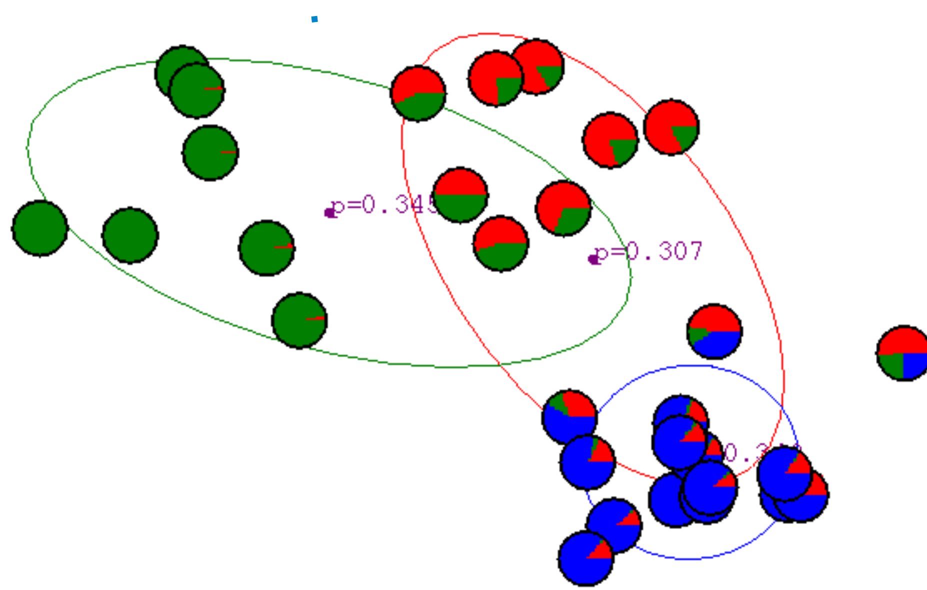
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

After 3rd Iteration

For each point, revising its proportions belonging to each of the K clusters

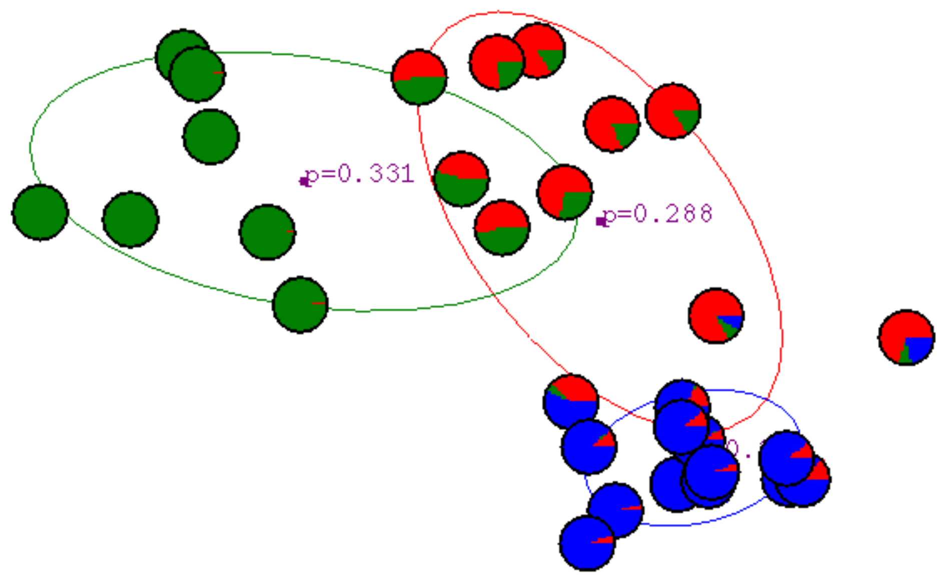


$p(\mu_j)$

For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

After 4th Iteration

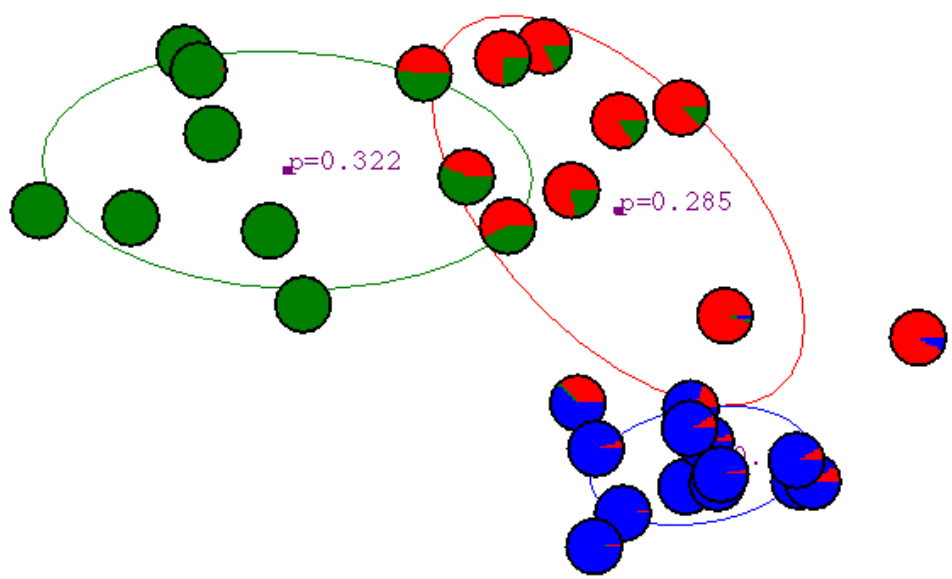
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

After 5th Iteration

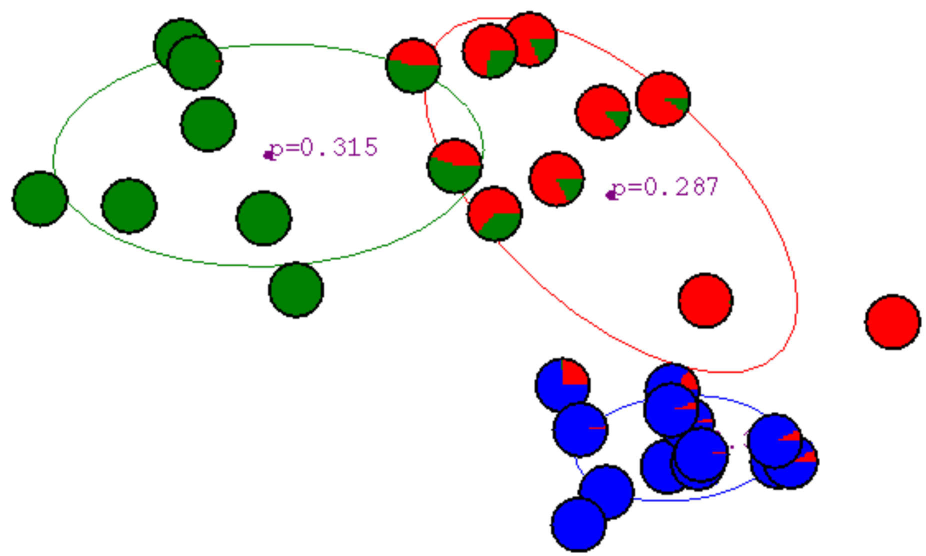
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

After 6th Iteration

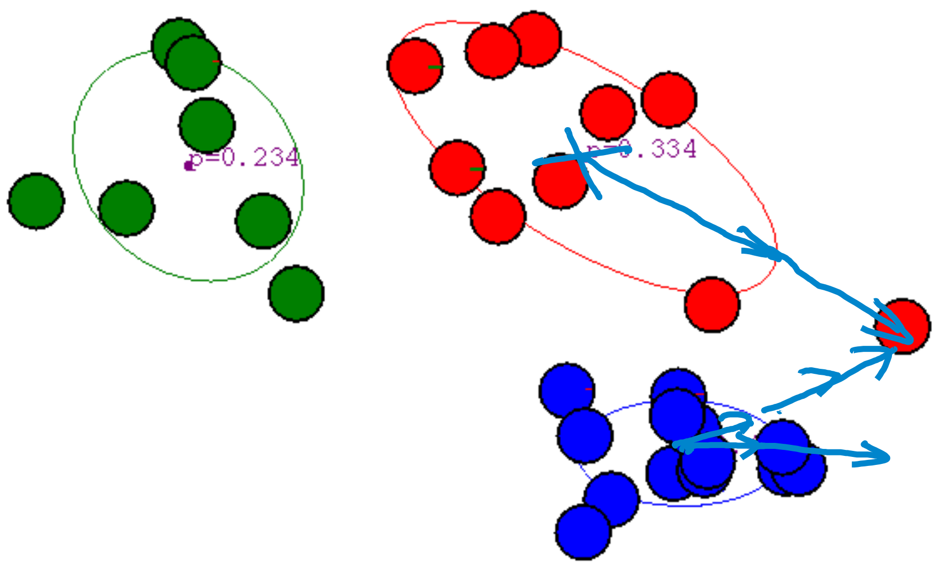
For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

After 20th Iteration

For each point, revising its proportions belonging to each of the K clusters



For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

(3) GMM Clustering

Task	Clustering
↓	↓
Representation	Mixture of Gaussian
↓	↓
Score Function	Likelihood
↓	↓
Search/Optimization	EM algorithm
↓	↓
Models, Parameters	Each point's soft membership & mean / covariance per cluster

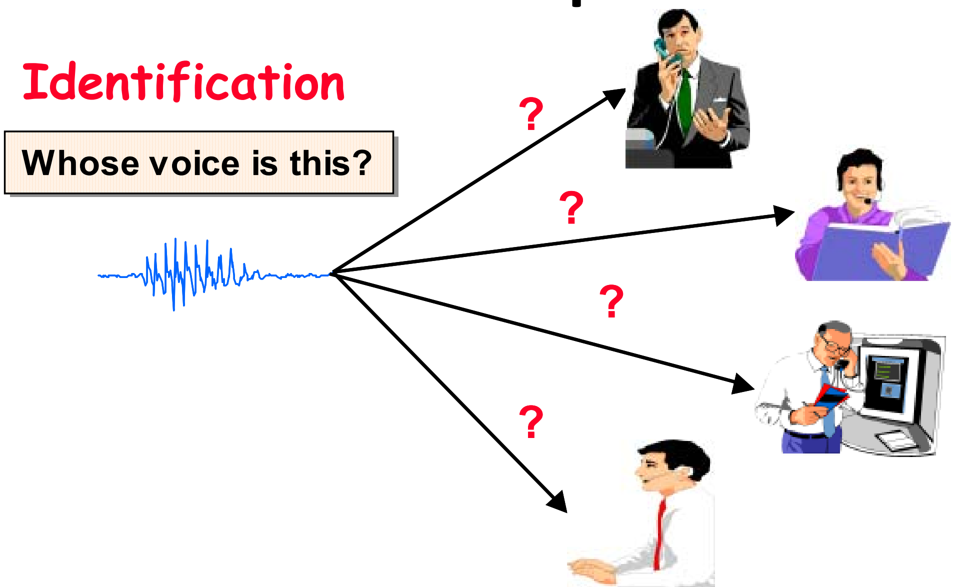
$$\sum_i \log \prod_{i=1}^{n} p(x=x_i) = \sum_i \log\left[\sum_{\mu_j} p(\mu=\mu_j)\frac{1}{(2\pi)\left|\Sigma_j\right|^{1/2}} e^{-\frac{1}{2}\left(\vec{x}-\vec{\mu}_j\right)^T \Sigma_j^{-1}\left(\vec{x}-\vec{\mu}_j\right)}\right]$$

Partitional : Gaussian Mixture Model

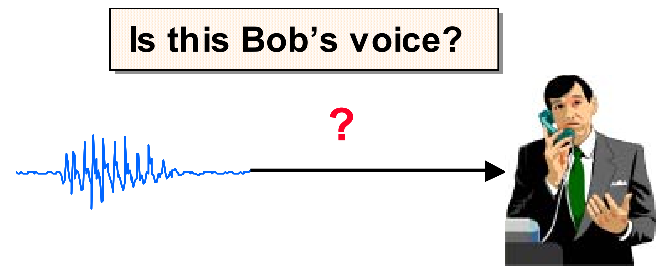
- 1. Review of Gaussian Distribution
- 2. GMM for clustering : basic algorithm
- 3. GMM connecting to K-means
- 4. GMM examples
- 5. Applications of GMM
- 6. Problems of GMM and K-means

Application (I) :
Three Speaker Recognition Tasks

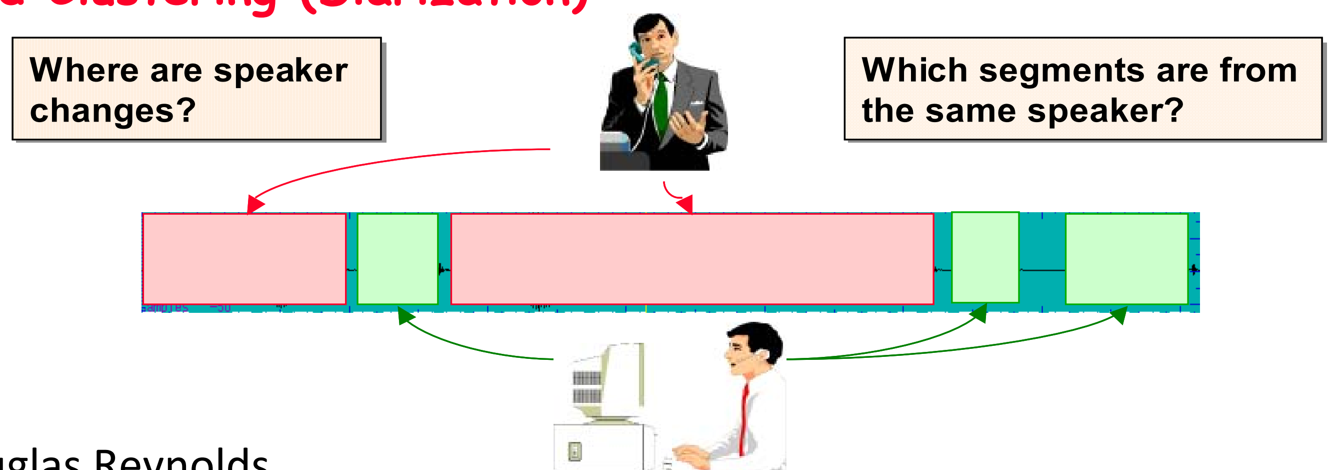
Identification



Verification/Authentication/ Detection



Segmentation and Clustering (Diarization)

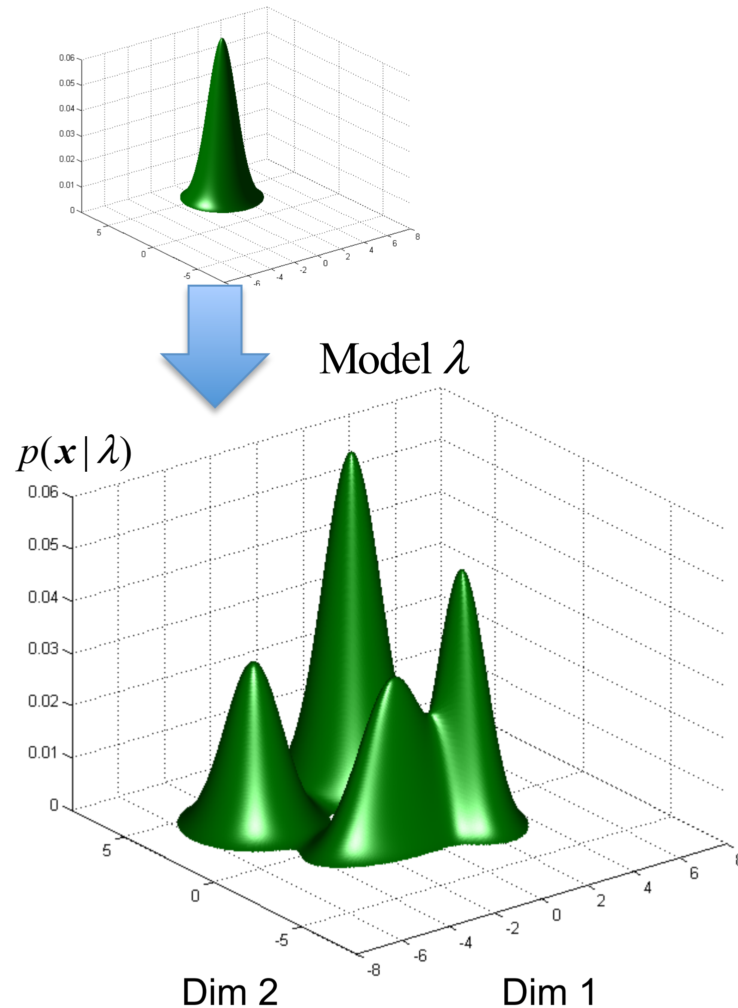


slide from Douglas Reynolds

Application (I) : GMMs for speaker recognition

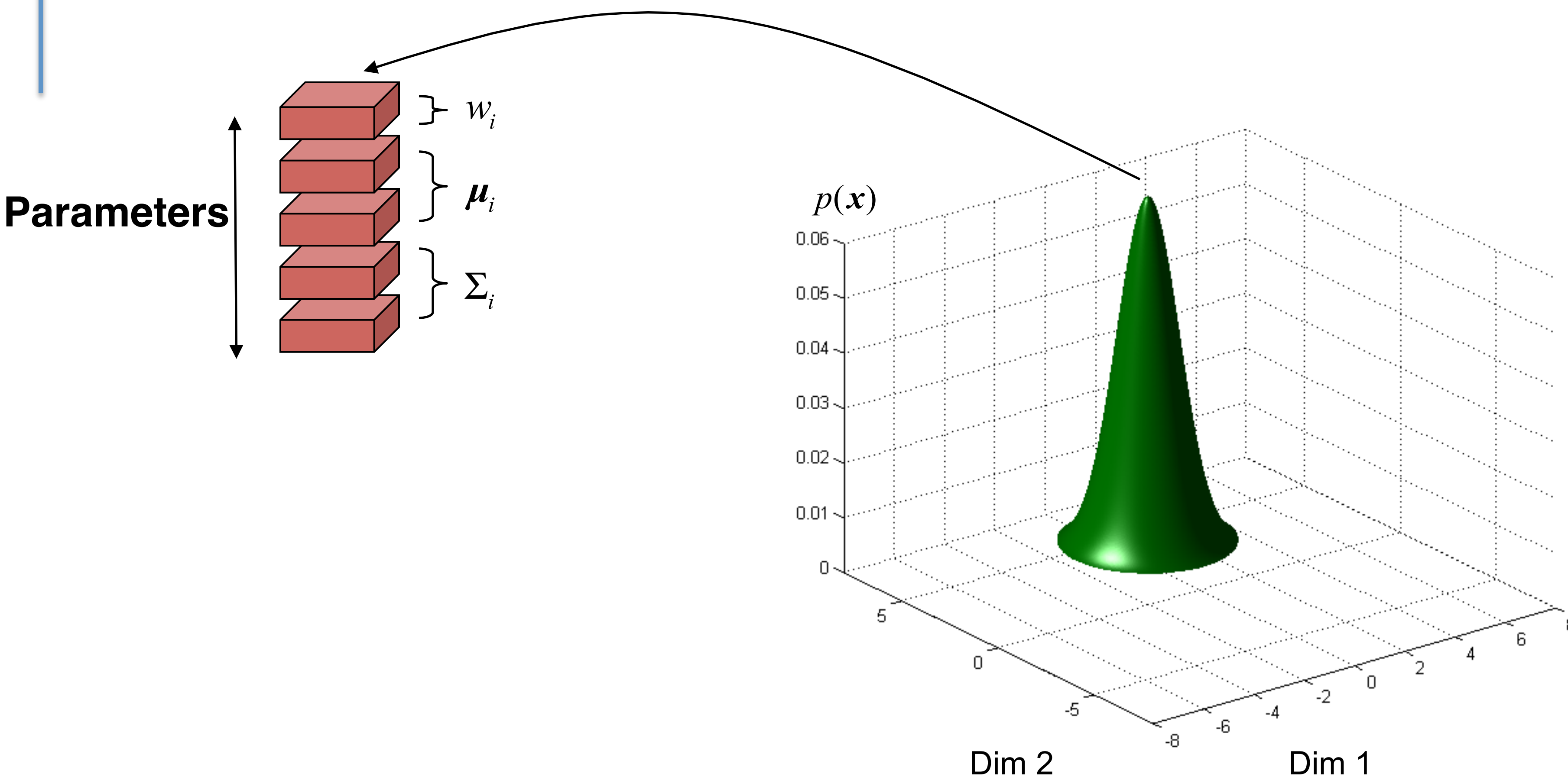
- A Gaussian mixture model (GMM) represents features as the weighted sum of multiple Gaussian distributions

- Each Gaussian state i has a
  - Mean $\boldsymbol{\mu}_i$
  - Covariance Σ_i
  - Weight w_i



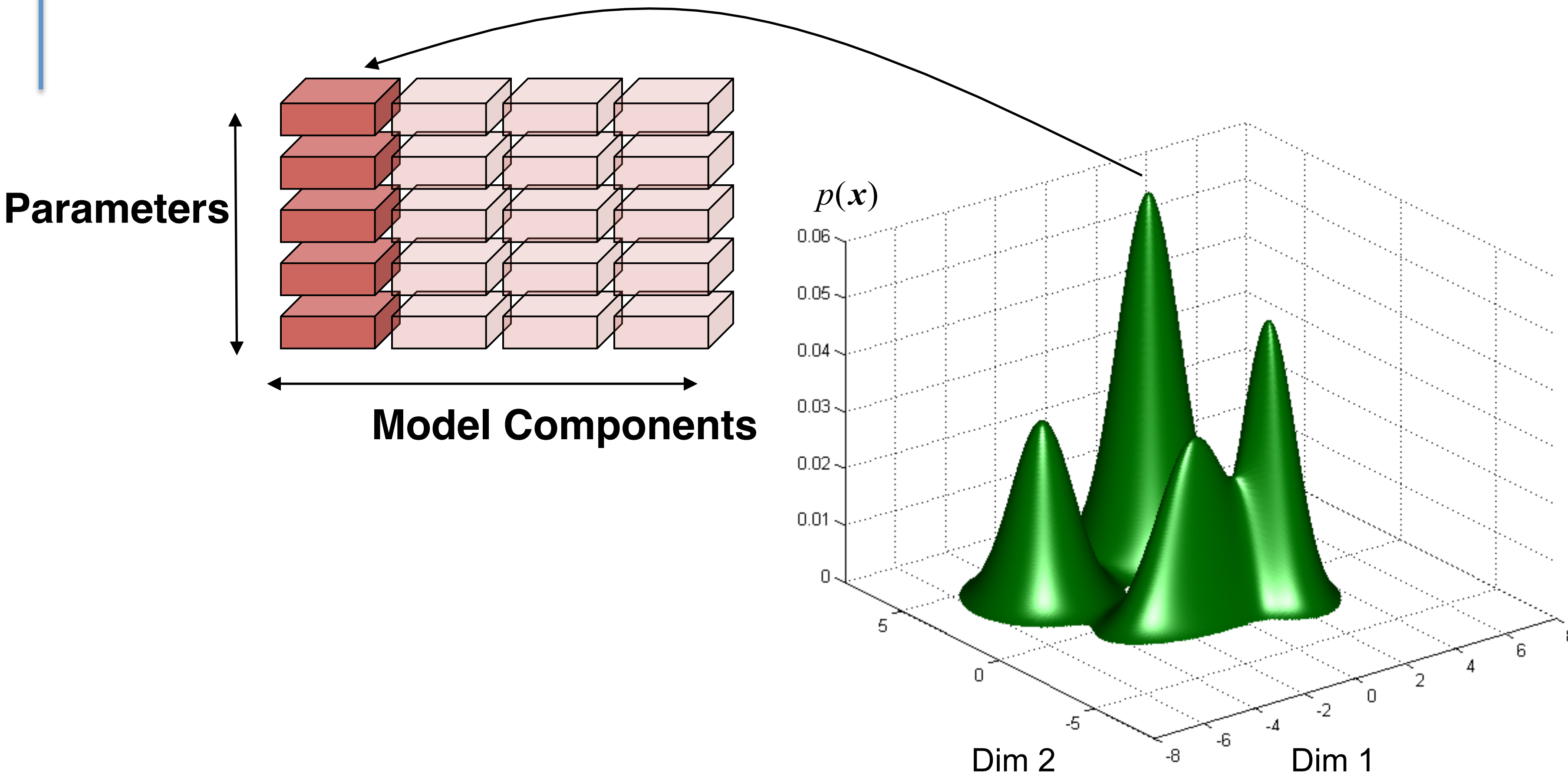
Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

Recognition Systems
Gaussian Mixture Models



Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

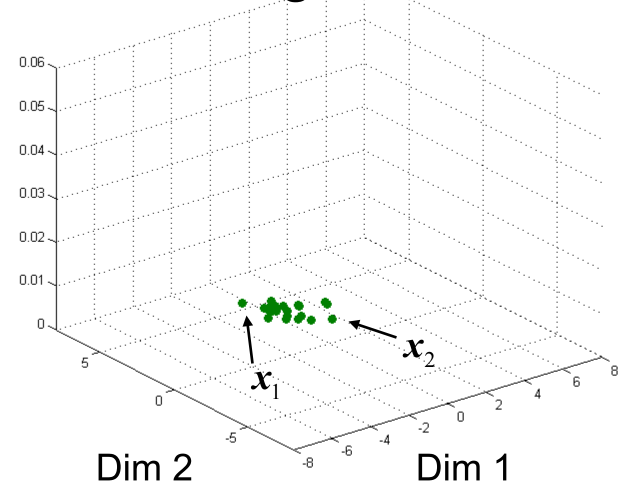
Recognition Systems
Gaussian Mixture Models

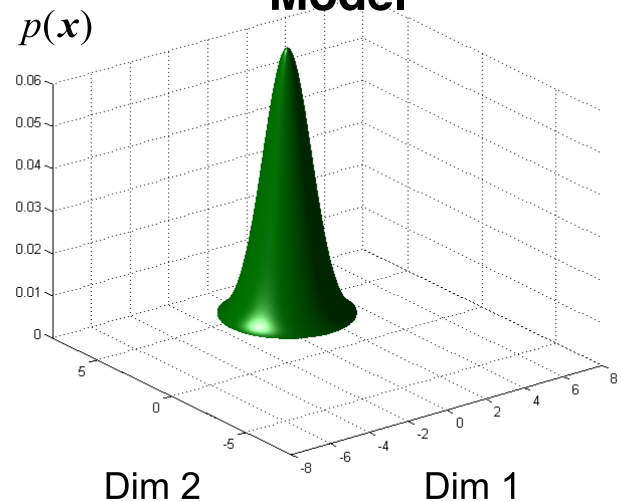


Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

GMM training

- During training, the system learns about the data it uses to make decisions
  - A set of features are collected from a speaker (or language or dialect)

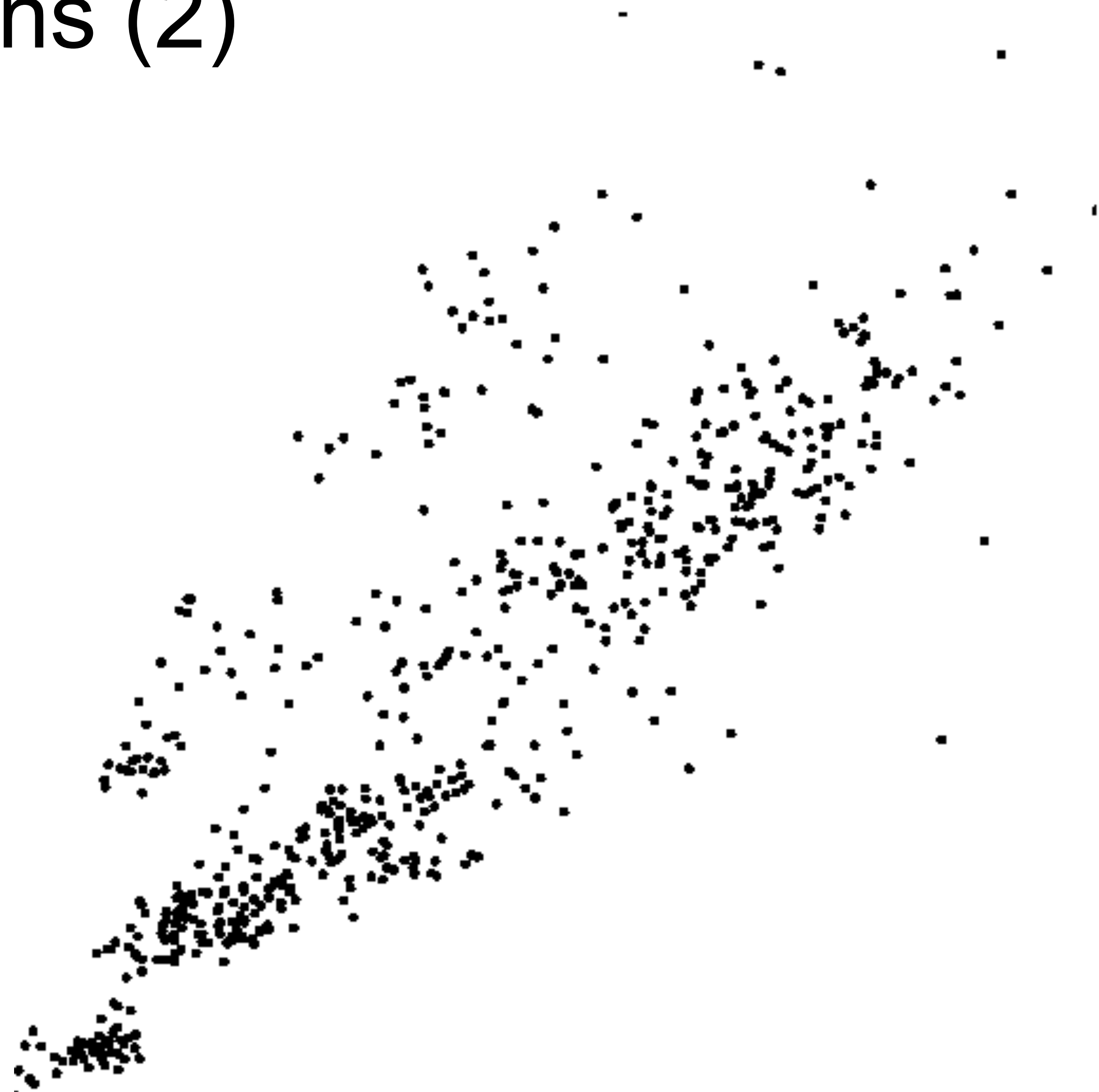




Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

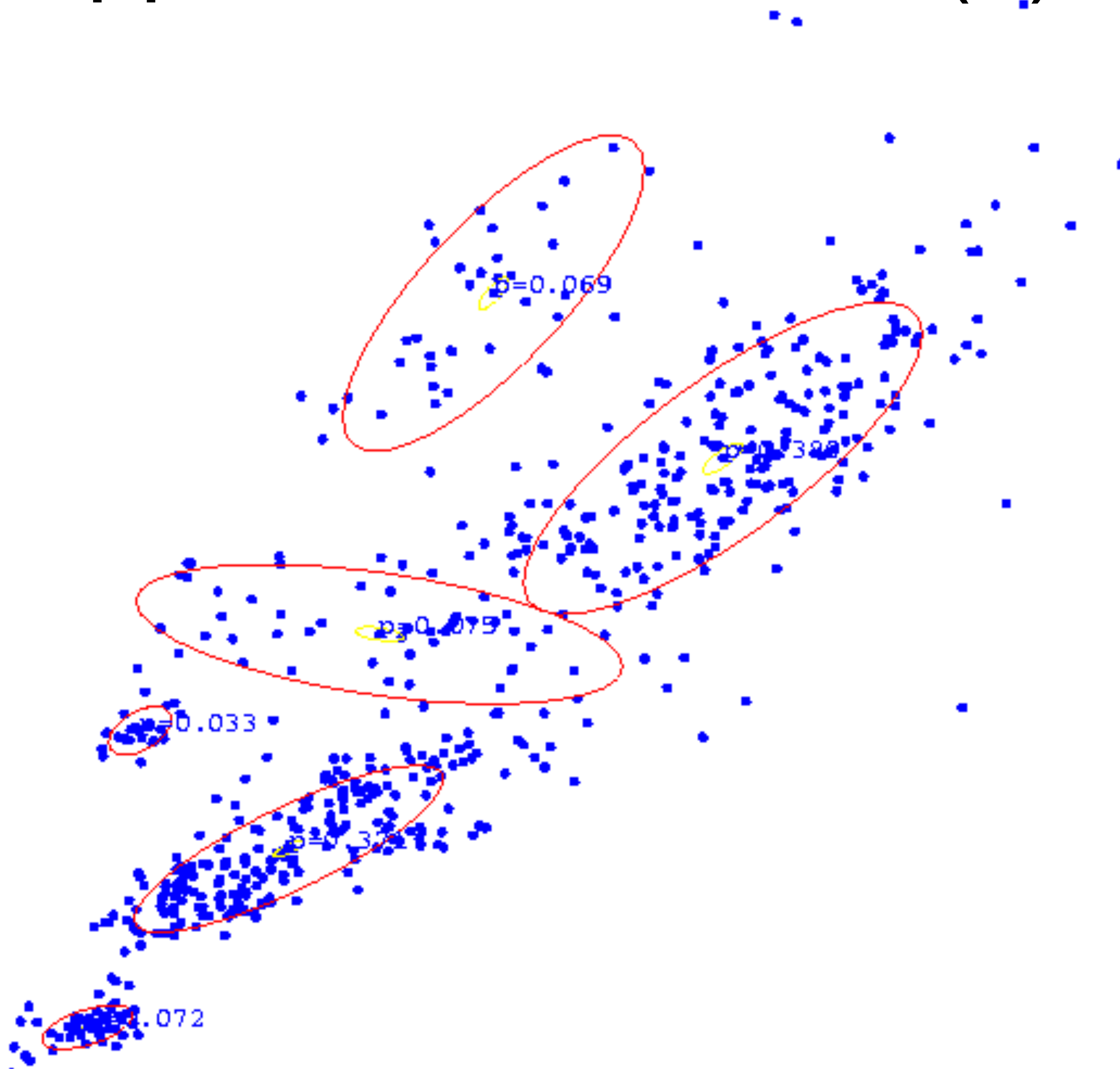
Applications (2)

Some Bio Assay data



Applications of GMM (2)

GMM clustering of the assay data



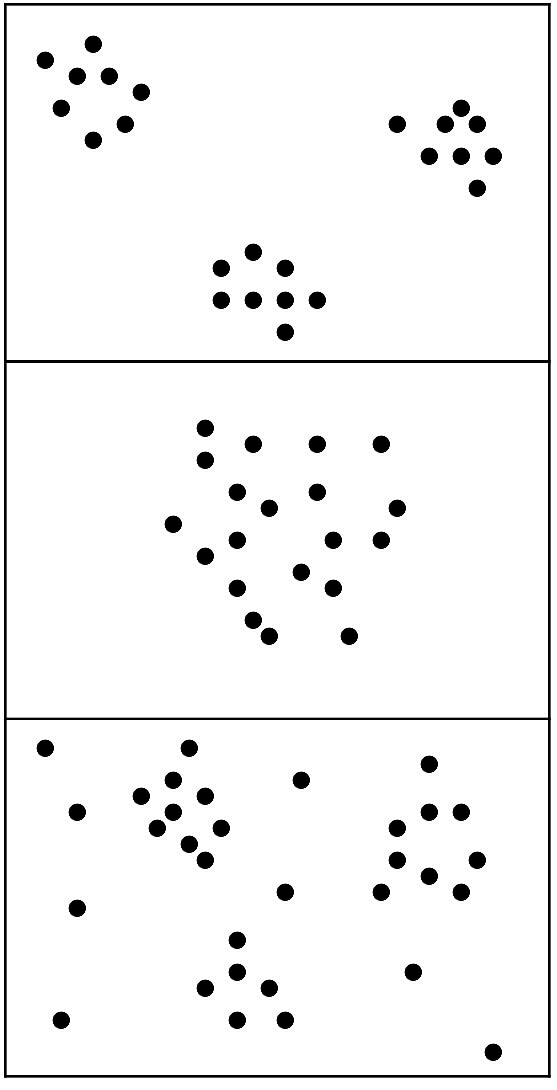
Applications of GMM (2)

Resulting Clusters Density Plot

Partitional : Gaussian Mixture Model

- 1. Review of Gaussian Distribution
- 2. GMM for clustering : basic algorithm
- 3. GMM connecting to K-means
- 4. GMM examples
- 5. Applications of GMM
- 6. Problems of GMM and K-means

Unsupervised Learning: not as hard as it looks



Sometimes easy

Sometimes impossible

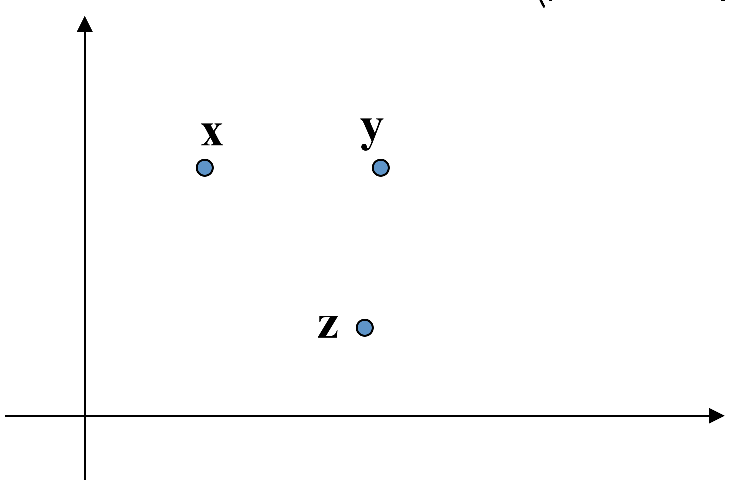
and sometimes in between

Problems (I)

- Both k-means and mixture models need to compute centers of clusters and explicit distance measurement
  - Given strange distance measurement, the center of clusters can be hard to compute

E.g.,

$$\left\|\vec{x}-\vec{x}'\right\|_\infty = \max\left(\left|x_1 - x_1'\right|, \left|x_2 - x_2'\right|, \ldots, \left|x_p - x_p'\right|\right)$$



$$\left\|\mathbf{x}-\mathbf{y}\right\|_\infty = \left\|\mathbf{x}-\mathbf{z}\right\|_\infty$$

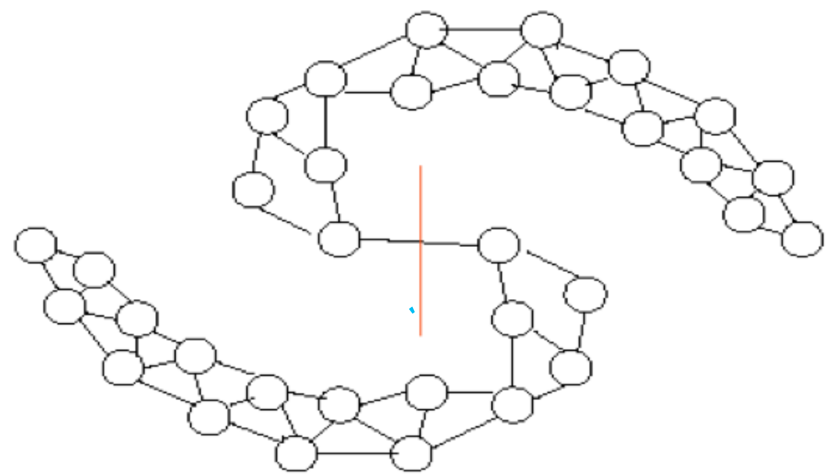
Problem (II)

tight

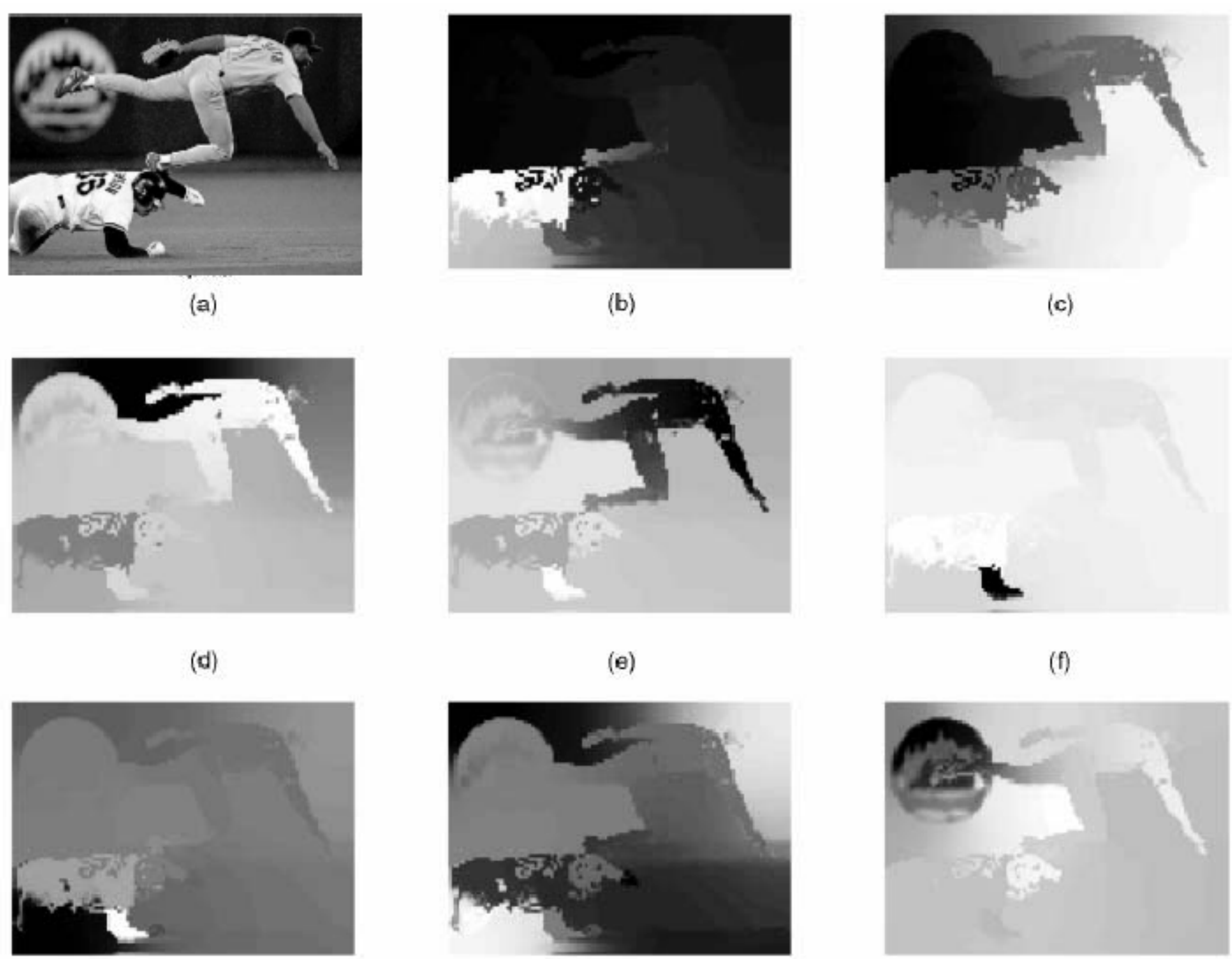
- Both k-means and mixture models look for compact clustering structures
  - In some cases, connected clustering structures are more desirable

Graph based clustering

e.g. MinCut, Spectral clustering



e.g. Image Segmentation through minCut

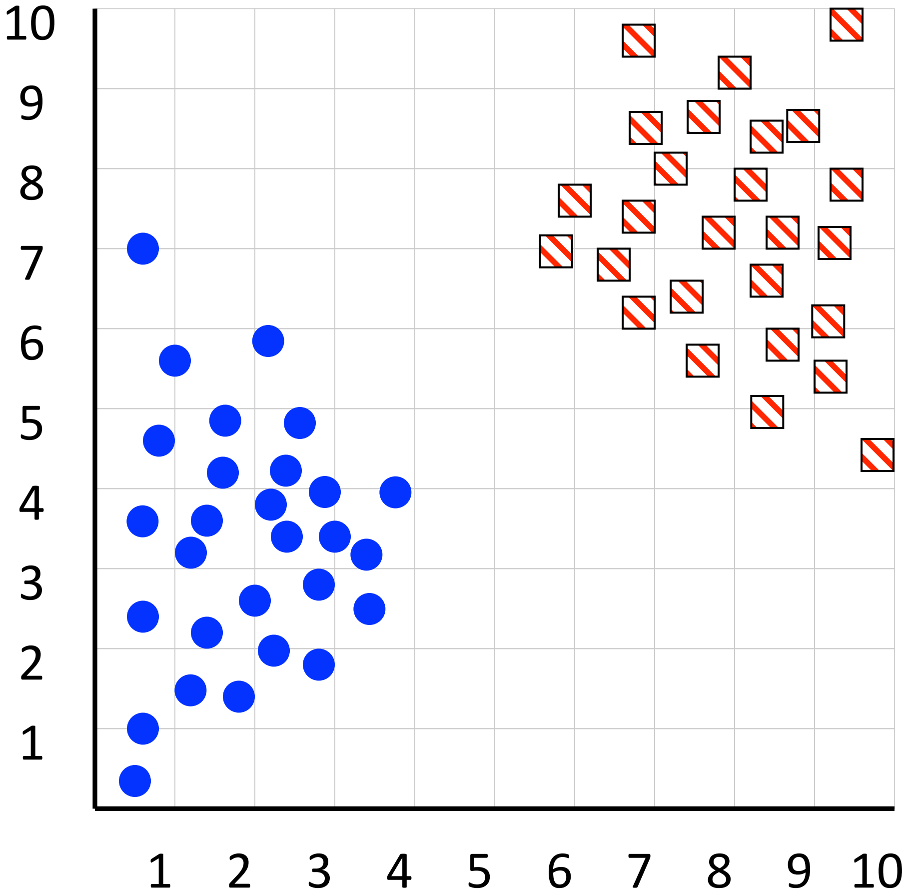


Roadmap: clustering

- Definition of "groupness”
- Definition of "similarity/distance"
- Representation for objects
- How many clusters?
- Clustering Algorithms
  - Partitional algorithms
  - Hierarchical algorithms
- Formal foundation and convergence

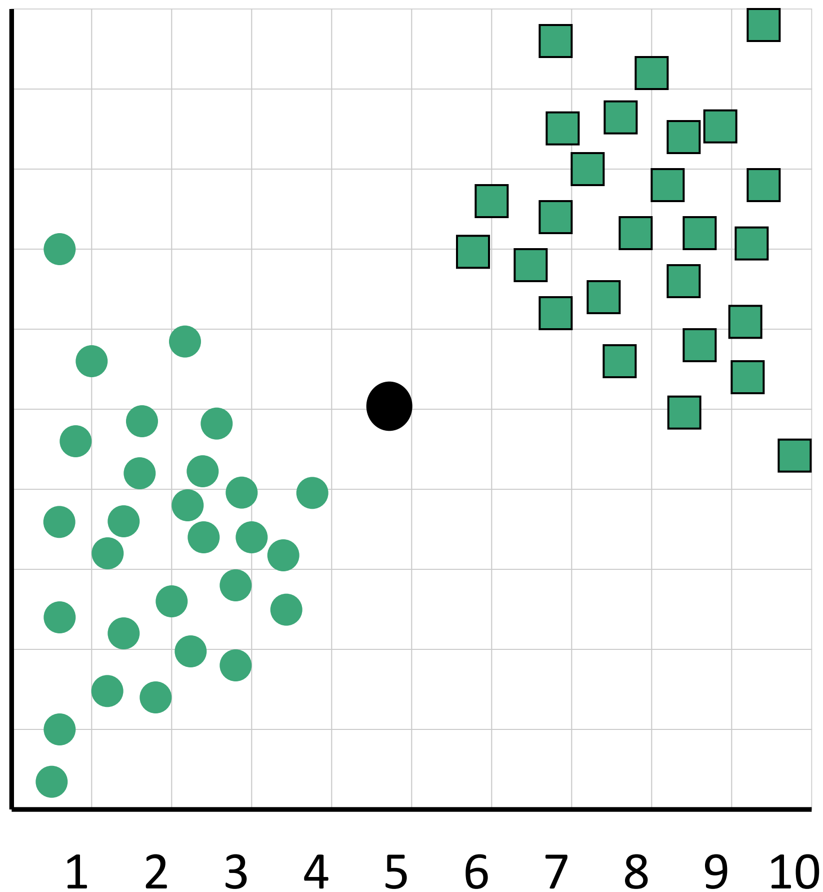
How can we tell the *right* number of clusters?

In general, this is a unsolved problem. However there exist many approximate methods.



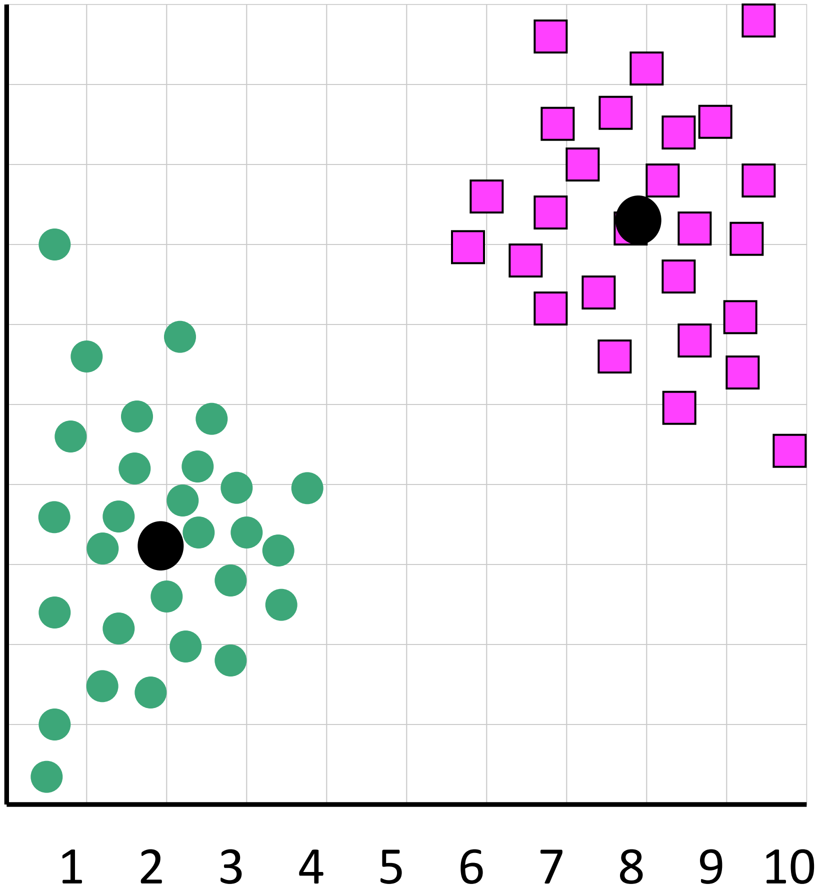
$$\underset{\{\vec{C}_j, m_{i,j}\}}{\arg\min} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j \right)^2$$

When k = 1, the objective function is 873.0



$$\underset{\{\vec{C}_j, m_{i,j}\}}{\arg\min} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j \right)^2$$

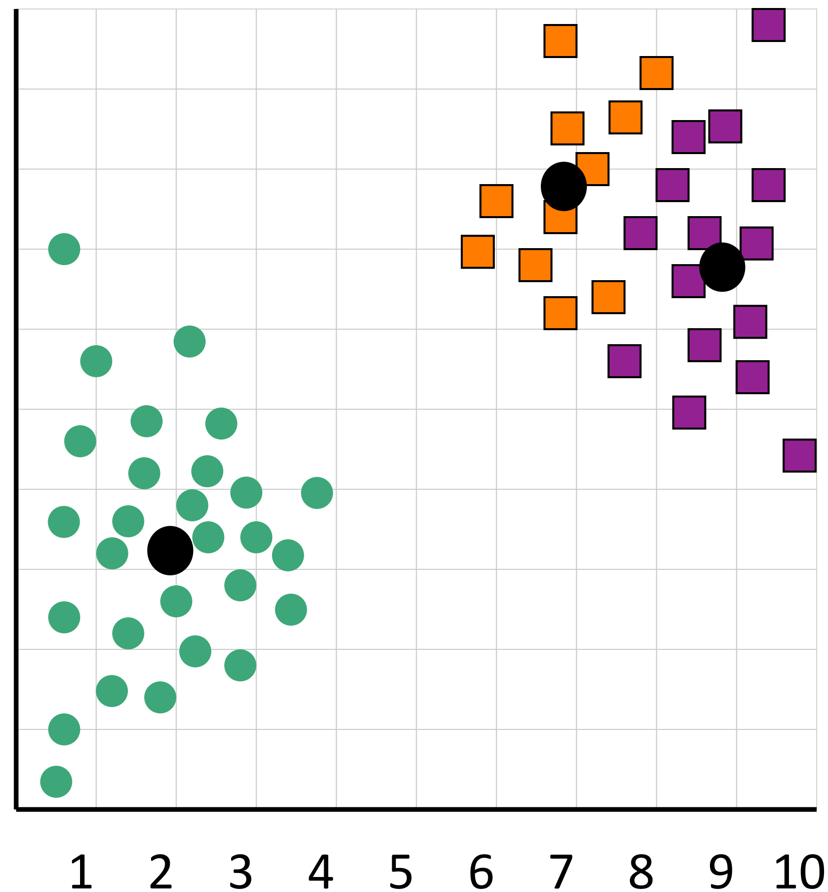
When k = 2, the objective function is 173.1



$$\underset{\{\vec{C}_j, m_{i,j}\}}{\arg\min} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j \right)^2$$

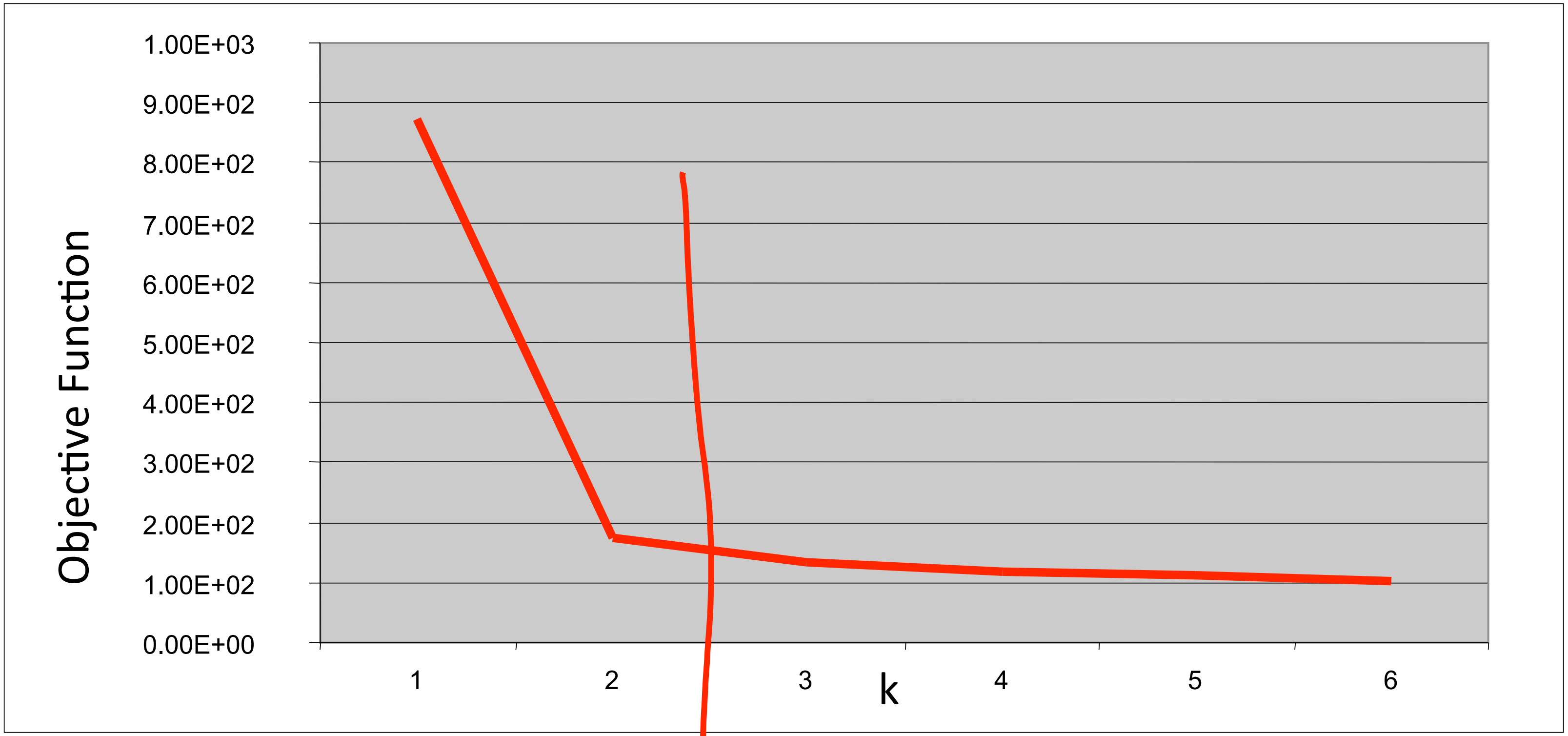
When k = 3, the objective function is 133.6

K = n , obj = 0



We can plot the objective function values for k equals 1 to 6…

The abrupt change at k = 2, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as "knee finding" or "elbow finding".



Note that the results are not always as clear cut as in this toy example

What Is A Good Clustering?

- Internal criterion: A good clustering will produce high quality clusters in which:
  - the intra-class (that is, intra-cluster) similarity is high
  - the inter-class similarity is low
  - The measured quality of a clustering depends on both the data representation and the similarity measure used

- External criteria for clustering quality
  - Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data
  - Assesses a clustering with respect to ground truth
  - Example:
    - Purity
    - entropy of classes in clusters (or mutual information between classes and clusters)

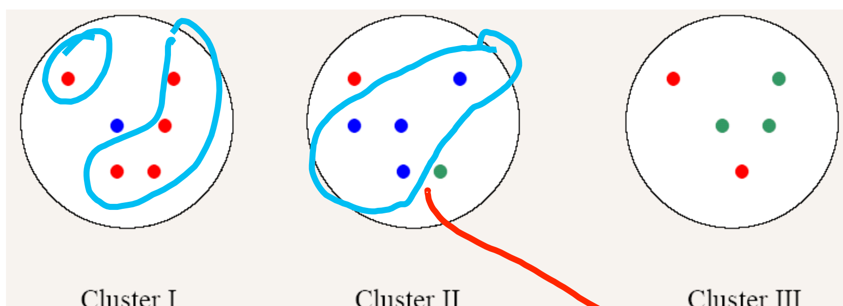
External Evaluation of Cluster Quality, e.g. using purity

- Simple measure: **purity**, the ratio between the dominant class in the cluster and the size of cluster
  - Assume data samples with C gold standard classes/groups, while the clustering algorithms produce K clusters, $\omega_1, \omega_2, \ldots, \omega_K$ with n_i members.

$$Purity(w_i) = \frac{1}{n_i}\max_j(n_{ij}) \quad j \in C$$

3 classes

  - Example

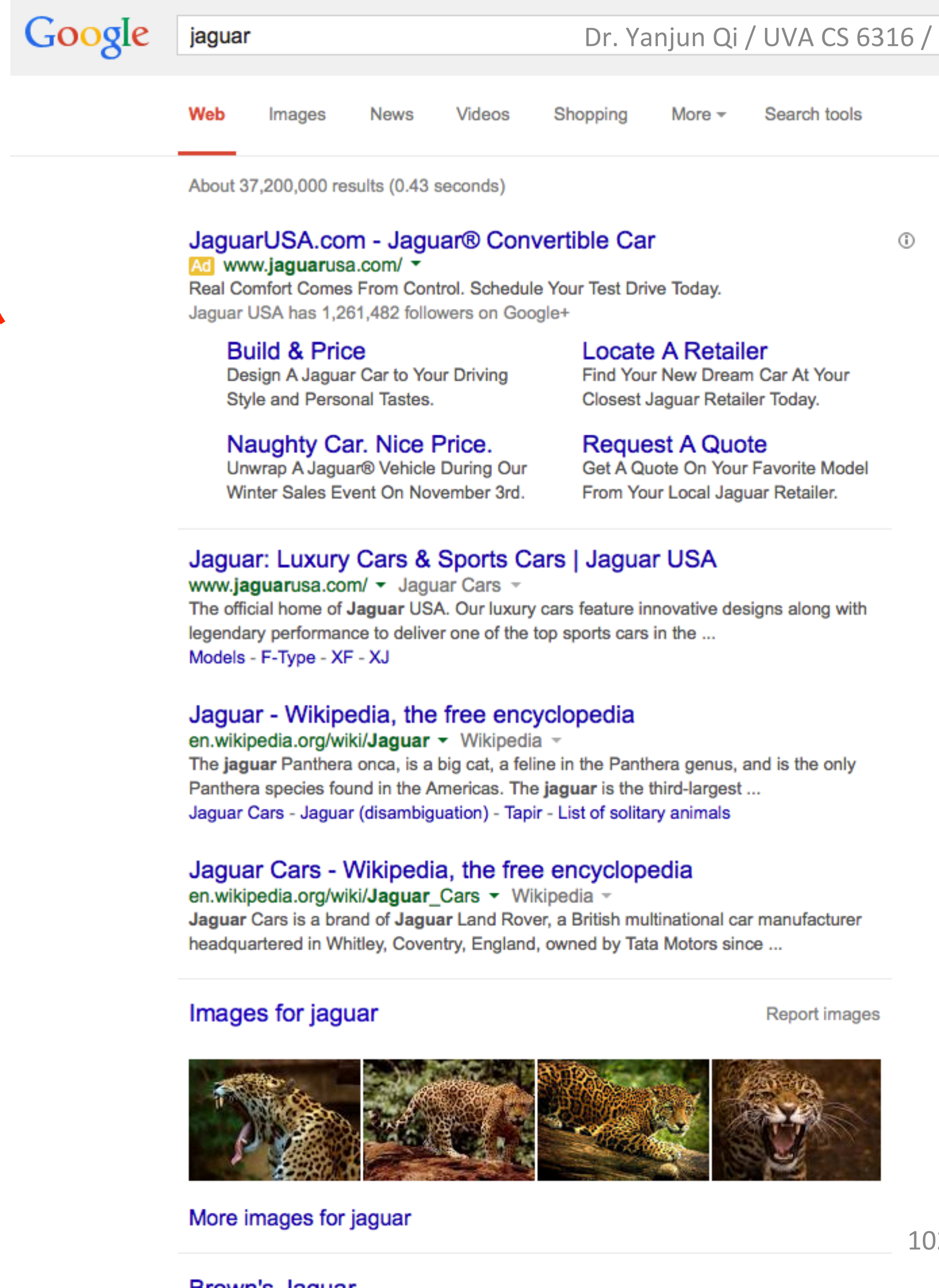


Cluster I: Purity = 1/6 (max(5, 1, 0)) = 5/6

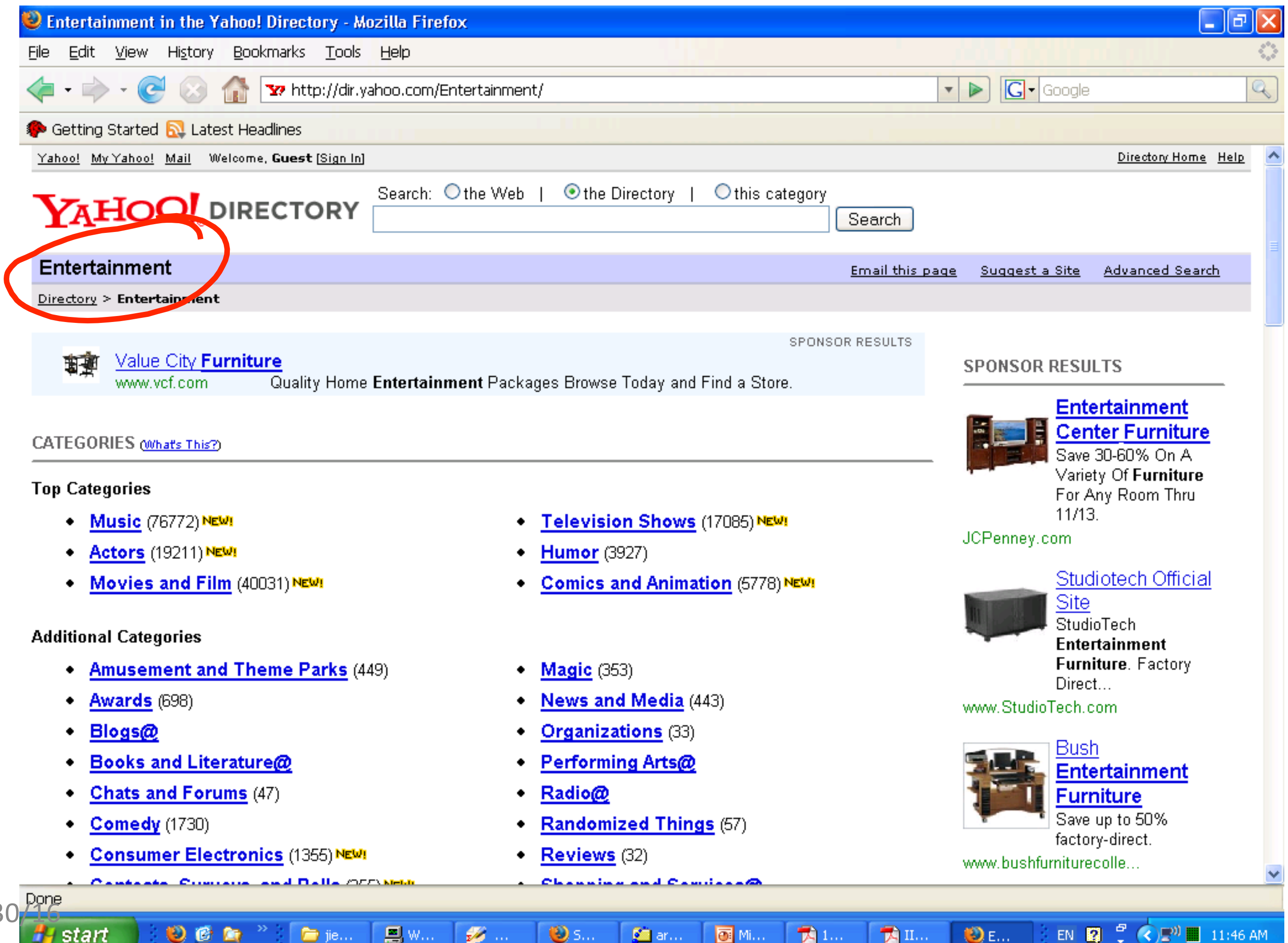
Cluster II: Purity = 1/6 (max(1, 4, 1)) = 4/6

Cluster III: Purity = 1/5 (max(2, 0, 3)) = 3/5 partition

Application (I): Search Result Clustering



Application (II): Navigation



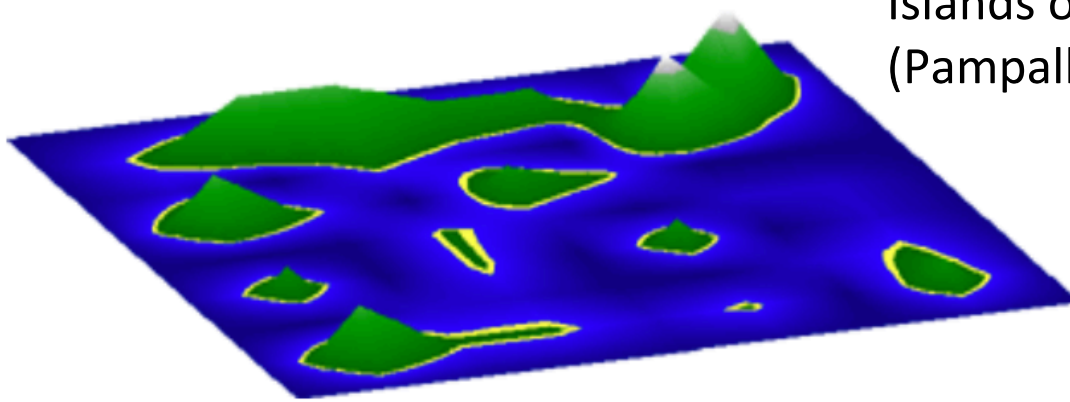
Hierachy

Application (III): Visualization

Islands of Music

Analysis, Organization, and Visualization of Music Archives

Islands of music
(Pampalk et al., KDD' 03)



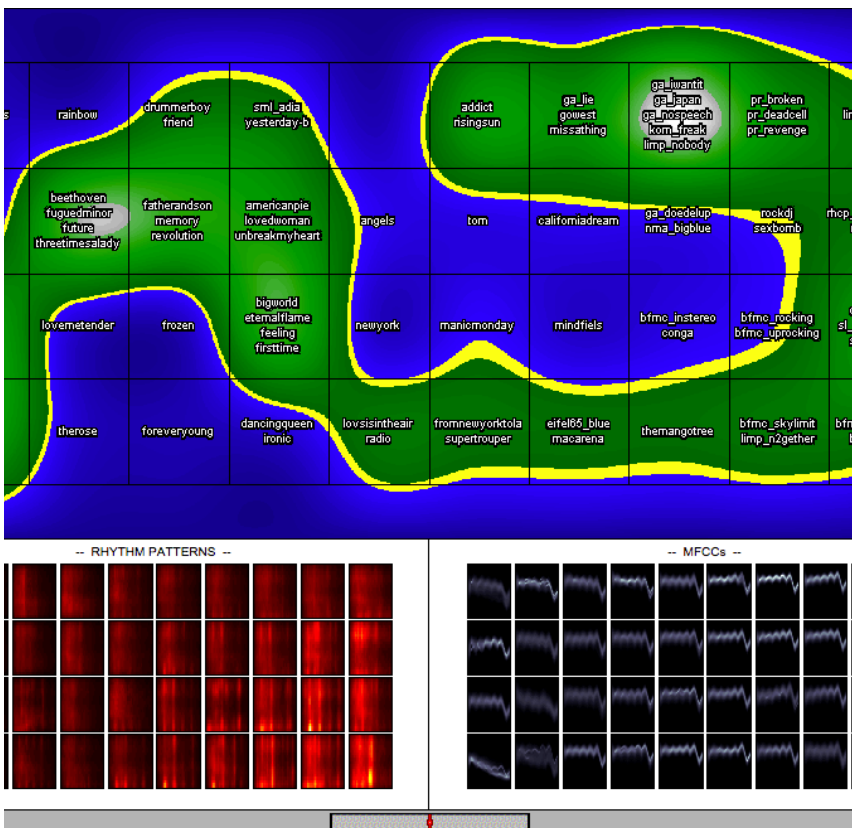
piece of music: member of a *music collection* and inhabitant of *islands of music*. Groups of similar pieces of music (also known as *genres*) like to gather around large mountains or small hills depending on the size of the group. Groups which are similar to each other like to live close together. Individuals which are not members of specific groups usually live near the beach and some very individualistic pieces might be found swimming in deep water.

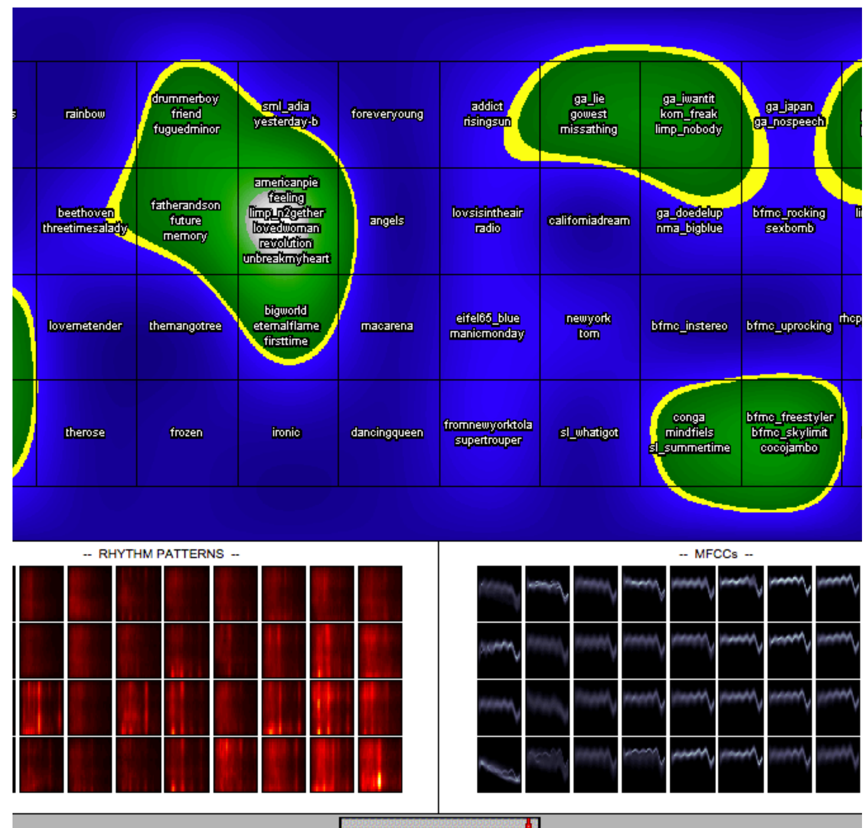
islands of music: serve as graphical *user interface* to a music collection and are intended to help the user explore vast amounts of music in an efficient way. Islands of music are generated automatically based on *psychoacoustics models* and *self-organizing maps*.

SOM

Application (III): Visualization (feature changes ➔ clusters' change)

Islands of music (Pampalk et al., KDD' 03, http://www.ofai.at/~elias.pampalk/kdd03/
Visualizing Changes in the Structure of Data for Exploratory Feature Selection)





References

- Hastie, Trevor, et al. *The elements of statistical learning*. Vol. 2. No. 1. New York: Springer, 2009.

- Big thanks to Prof. Eric Xing @ CMU for allowing me to reuse some of his slides
- Big thanks to Prof. Ziv Bar-Joseph @ CMU for allowing me to reuse some of his slides
- clustering slides from Prof. Rong Jin @ MSU