UVA CS 6316 – Fall 2015 Graduate: Machine Learning

Lecture 21: Unsupervised Clustering (II)

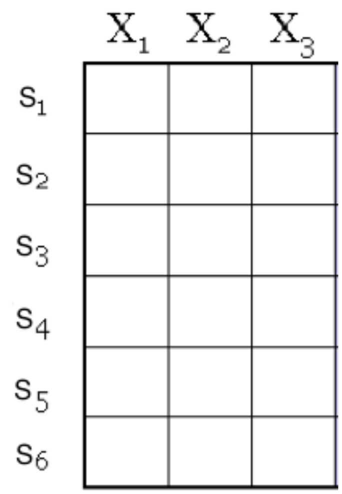
Dr. Yanjun Qi

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

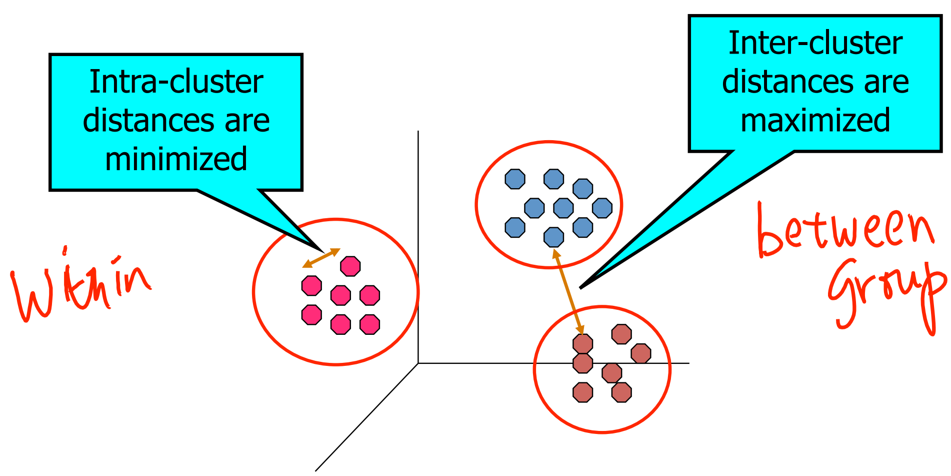
a data matrix of n observations on p variables $x_1, x_2, \ldots x_p$

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]

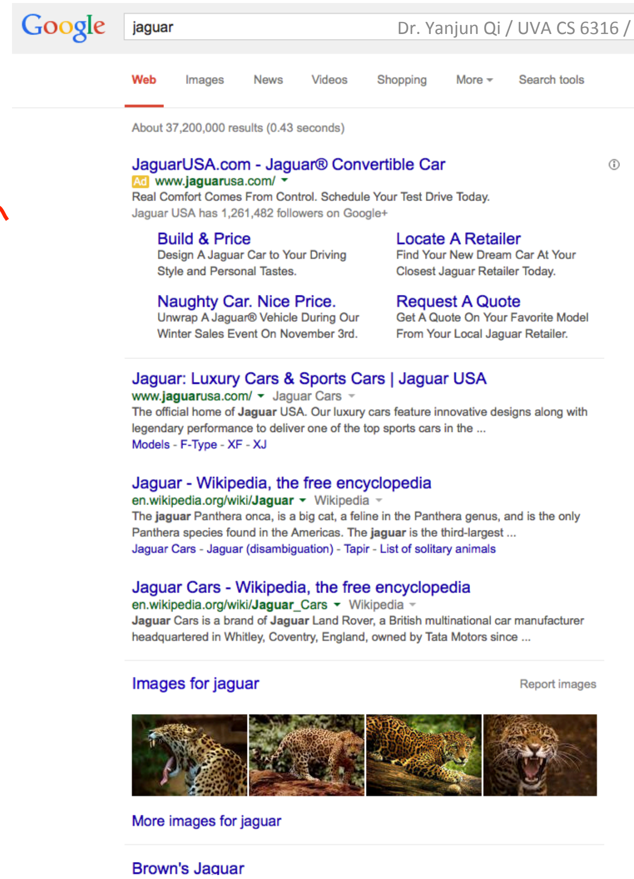
What is clustering?

- 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

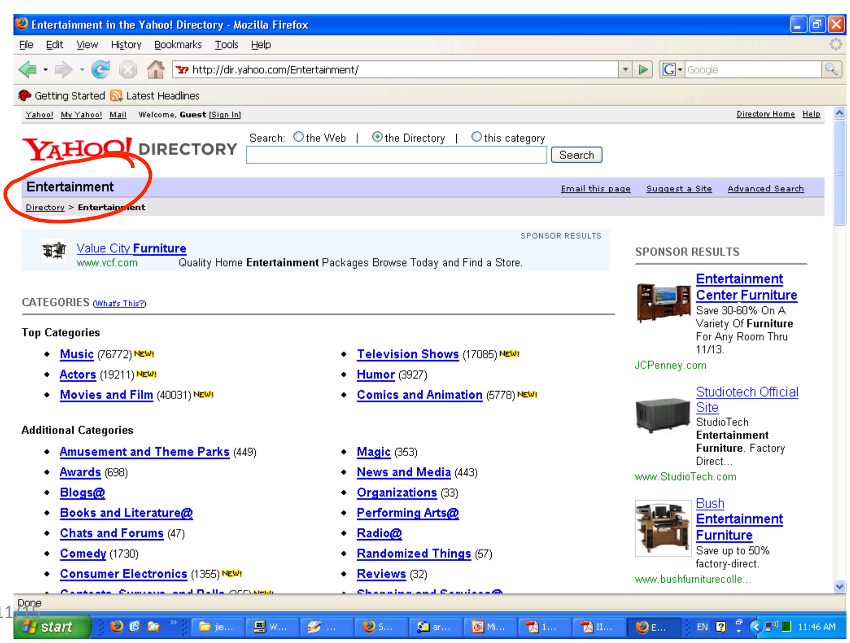


Application (I): Search Result Clustering

Partition



Application (II): Navigation

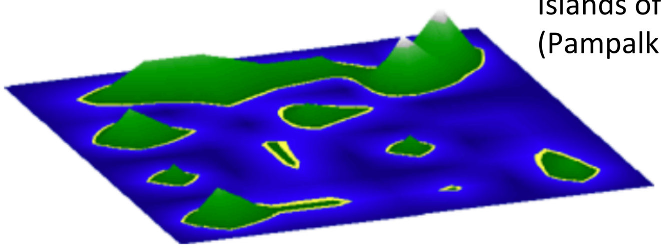


Hierachy

Application (III): Visualization

Islands of Music

Analysis, Organization, and Visualization of Music Archives



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

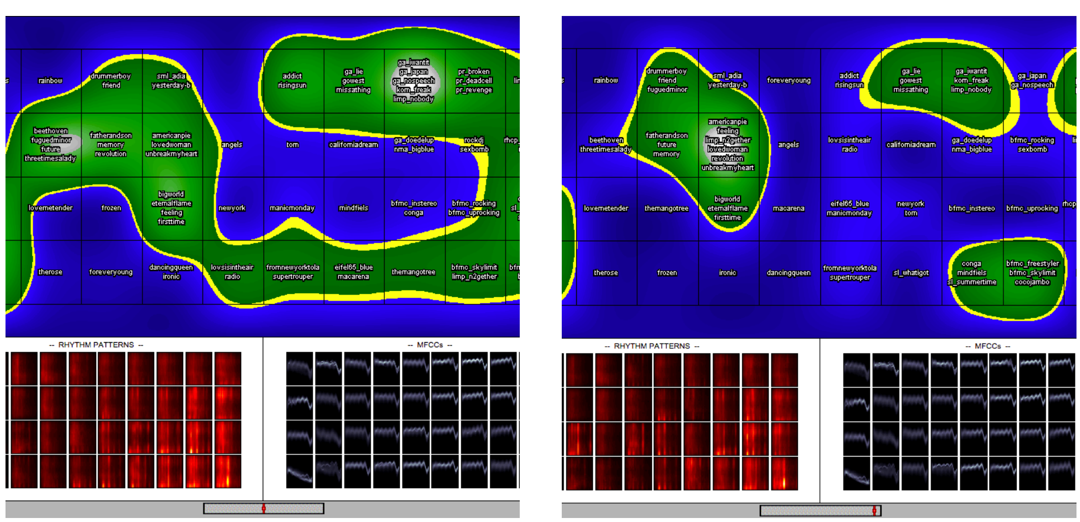
piece of music: member of a *music collection* and inhabitant of *islands of music*. Groups of similar pieces of music (also known as *genres*) like to gather around large mountains or small hills depending on the size of the group. Groups which are similar to each other like to live close together. 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)



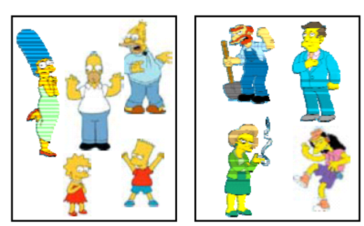
Roadmap: clustering

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

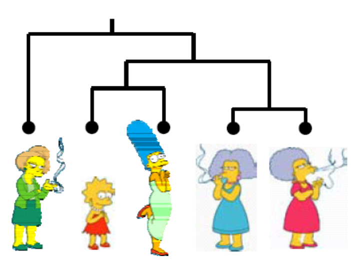
Dr. Yanjun Qi / UVA CS 6316 / f15

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



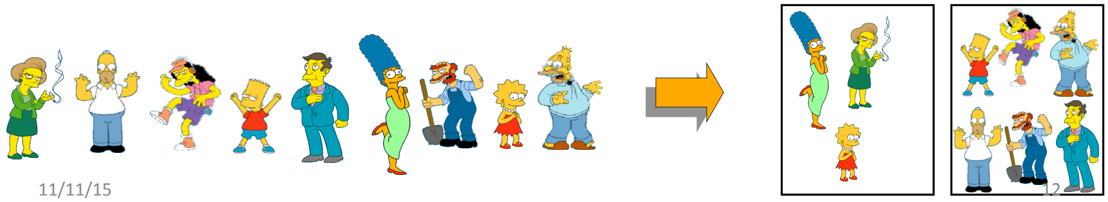
(1) Hierarchical Clustering

Task → Representation → Score Function → Search/Optimization → Models, Parameters

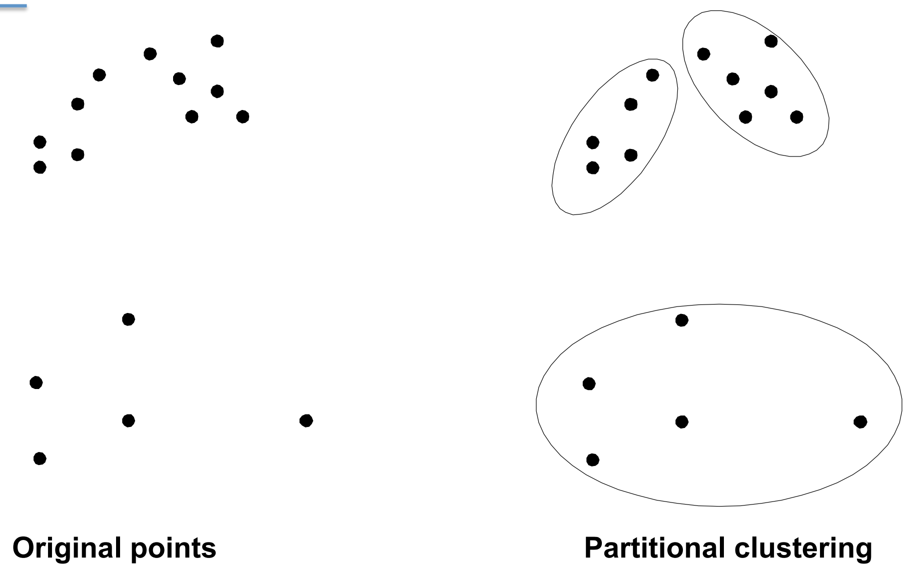
Clustering → n/a → No clearly defined loss → greedy bottom-up (or top-down) → Dendrogram (tree)

(2) Partitional Clustering

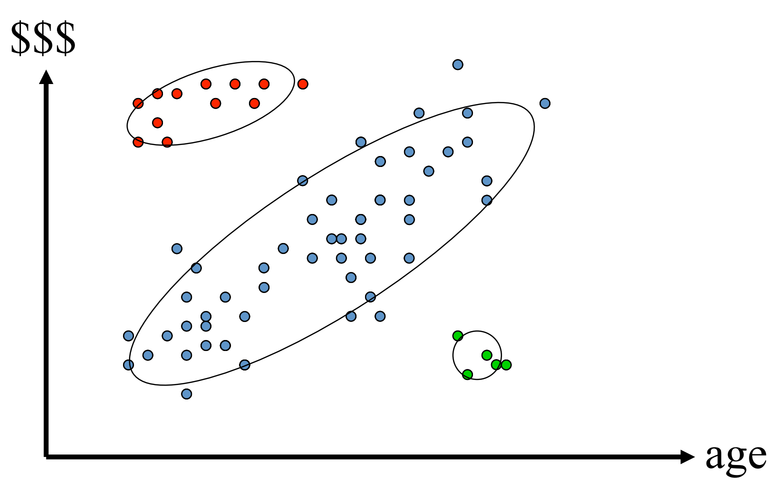
- Nonhierarchical
- Construct a partition of *n* objects into a set of *K* clusters
- User has to specify the desired number of clusters K.



Partitional clustering (e.g. K=3)

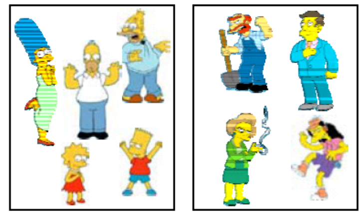


Partitional clustering (e.g. K=3)



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

    too expensive
  - Effective heuristic methods: K-means and K-medoids algorithms

K-Means

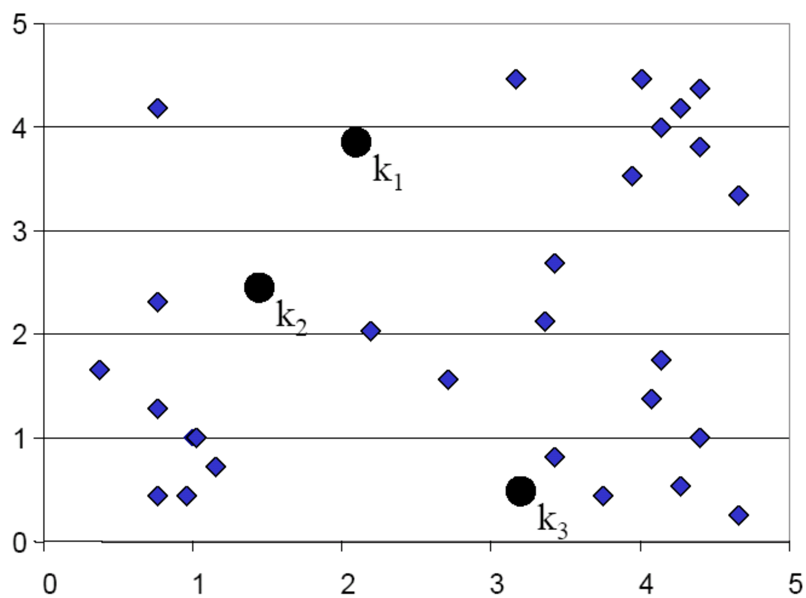
Algorithm

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)

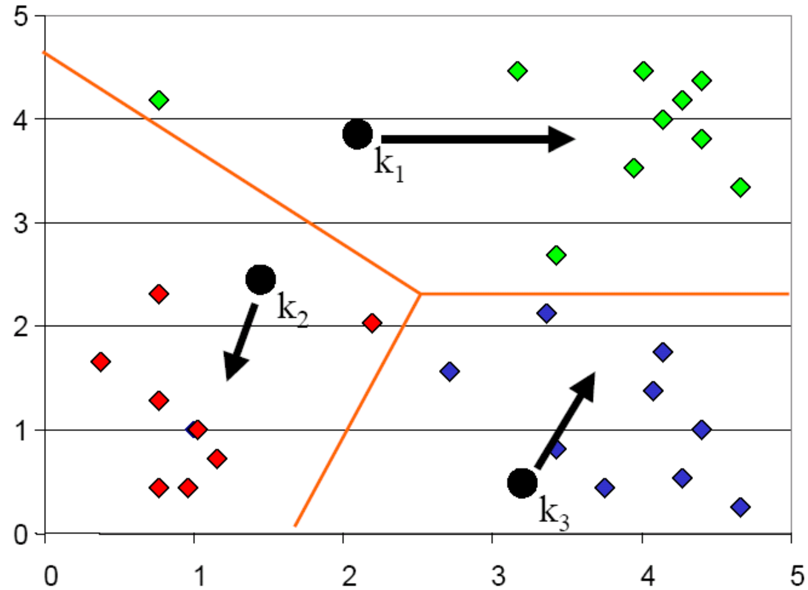
$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$

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.

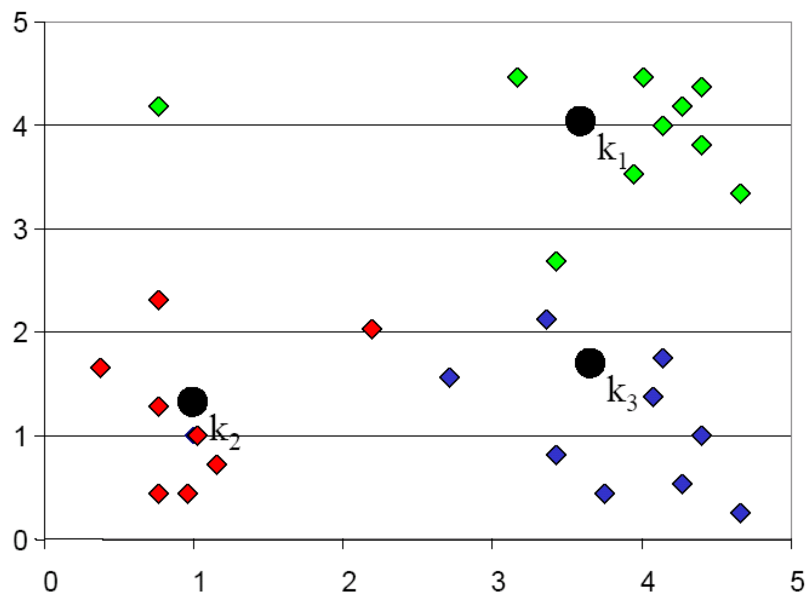
K-means Clustering: Step 1 - random guess of cluster centers



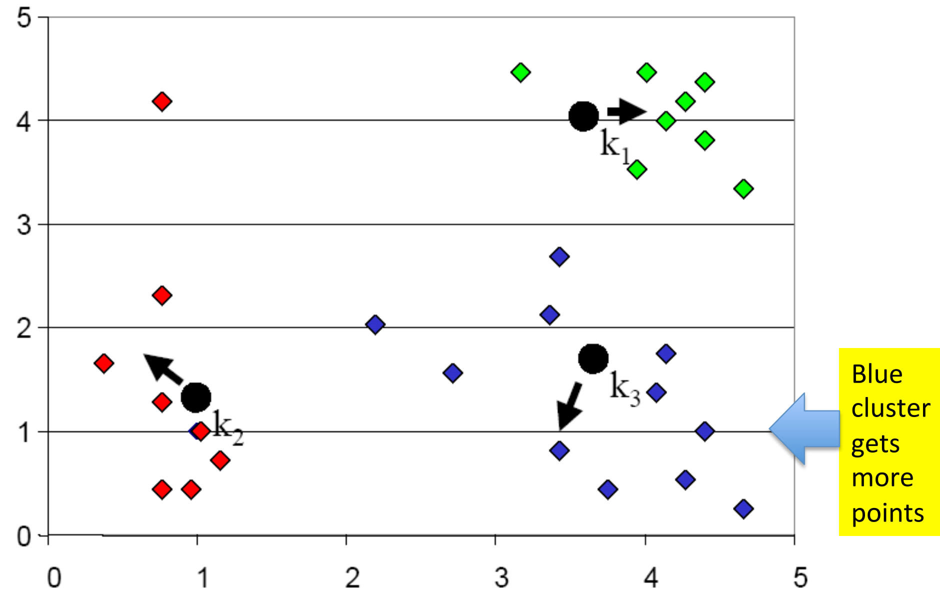
K-means Clustering: Step 2
- Determine the membership of each data points



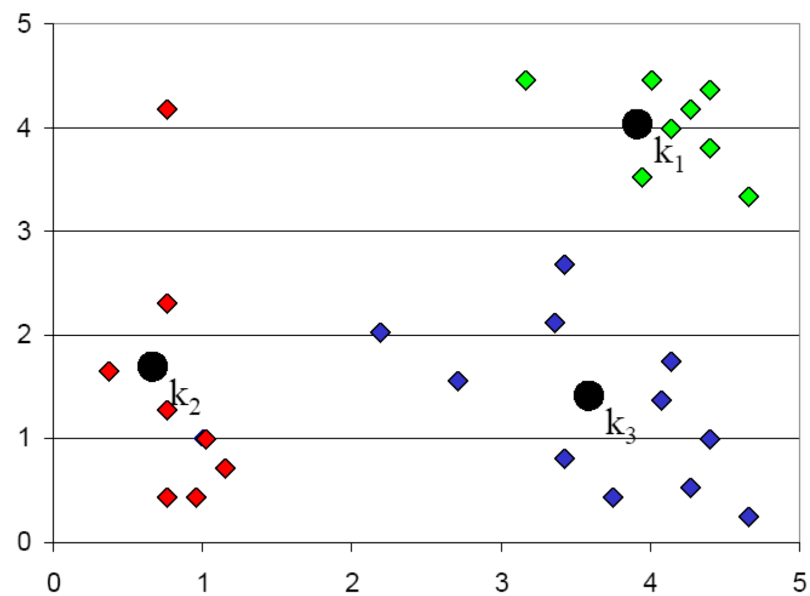
K-means Clustering: Step 3
- Adjust the cluster centers



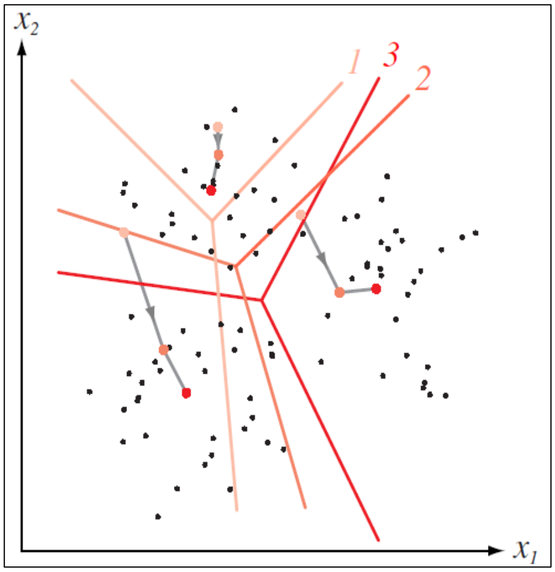
K-means Clustering: Step 4 - redetermine membership



K-means Clustering: Step 5 - readjust cluster centers



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.

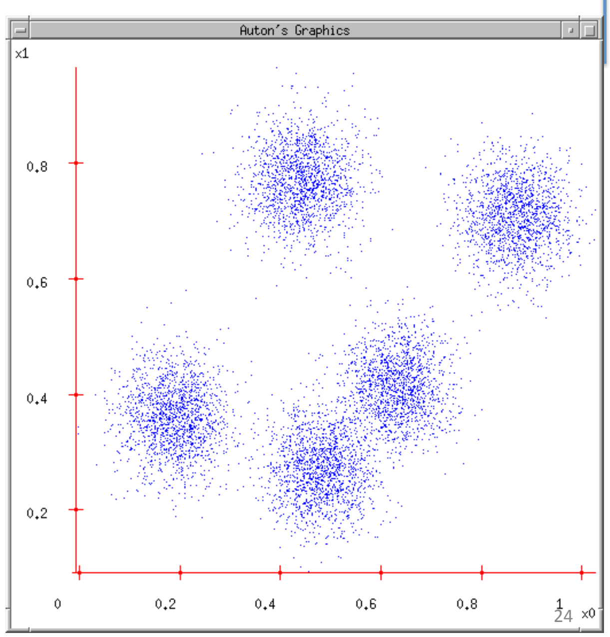
A partition amounts to a

[Voronoi Diagram]

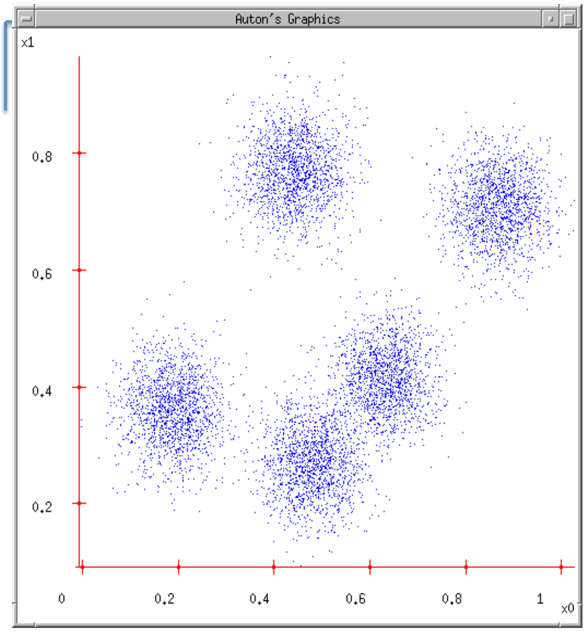
Changing positions of centroids leads to a new partitioning.

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

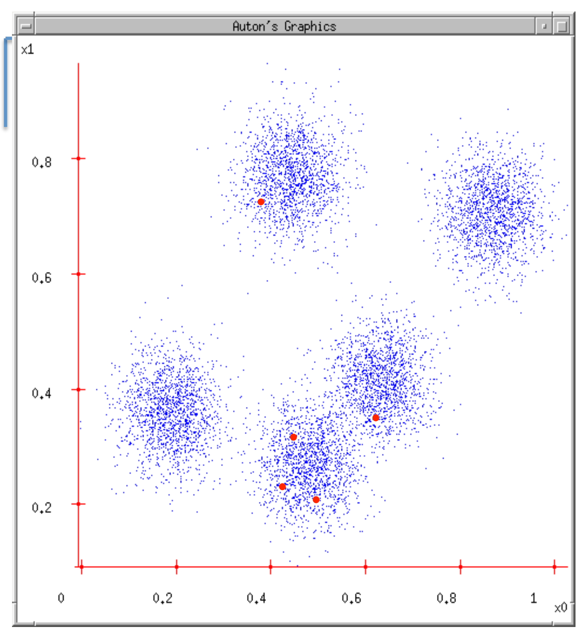


K-means: another Demo



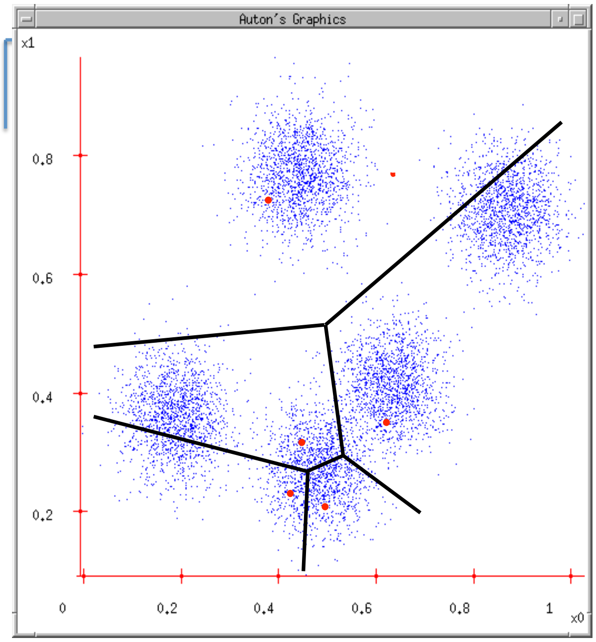
1. User set up the number of clusters they'd like. *(e.g. k=5)*

K-means: another Demo



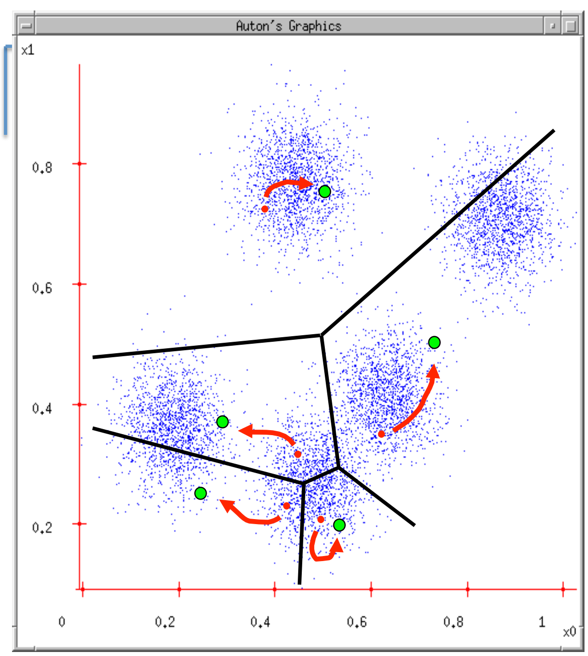
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess K cluster Center locations

K-means: another Demo



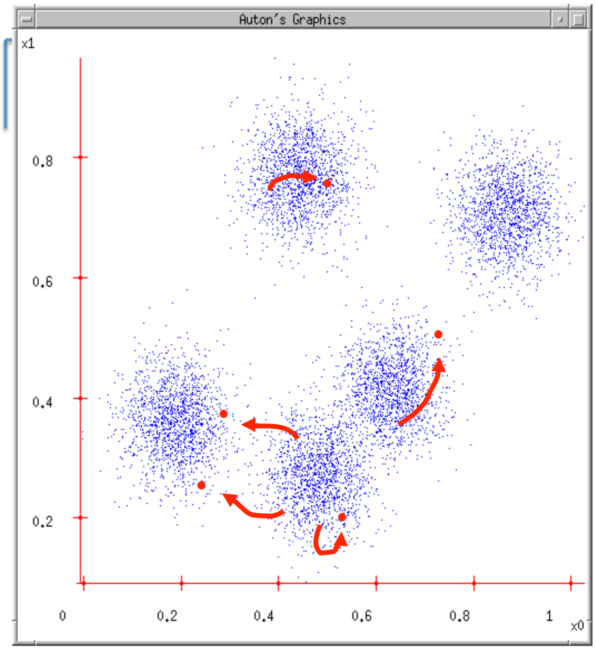
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess *K* cluster Center locations
3. Each data point finds out which Center it's closest to. (Thus each Center "owns" a set of data points)

K-means: another Demo



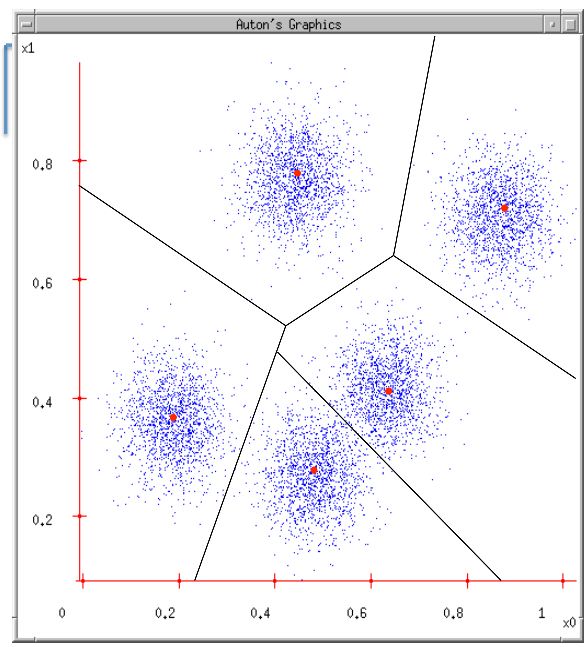
1. User set up the number of clusters they'd like. *(e.g. K=5)*
2. Randomly guess *K* cluster centre locations
3. Each data point finds out which centre it's closest to. (Thus each Center "owns" a set of data points)
4. Each centre finds the centroid of the points it owns

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!

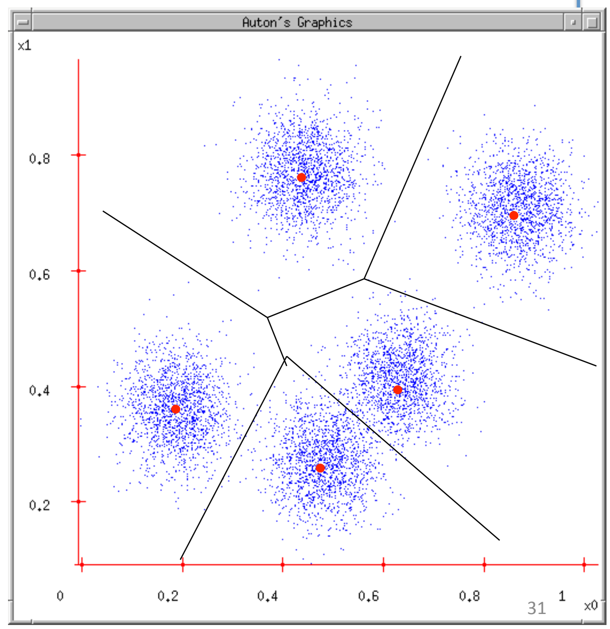
K-means

1. Ask user how many clusters

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*).

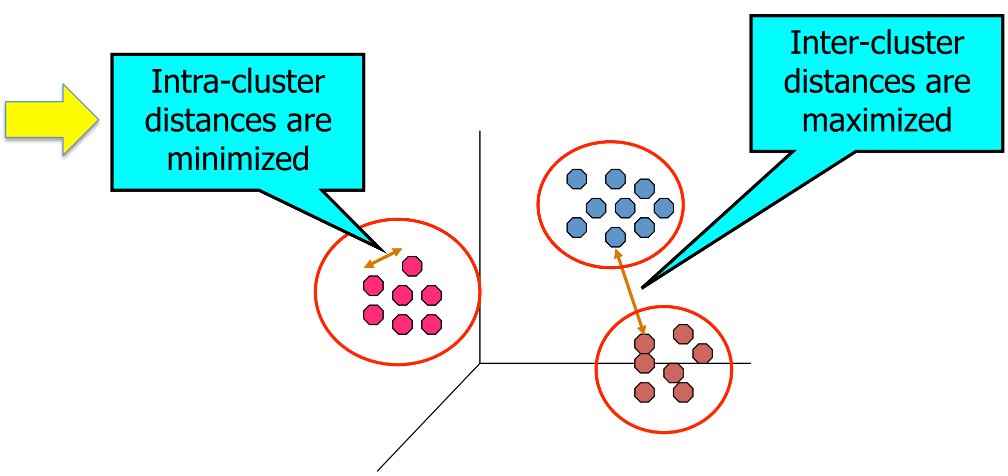
$O(n^3)$ Hierarchical

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.

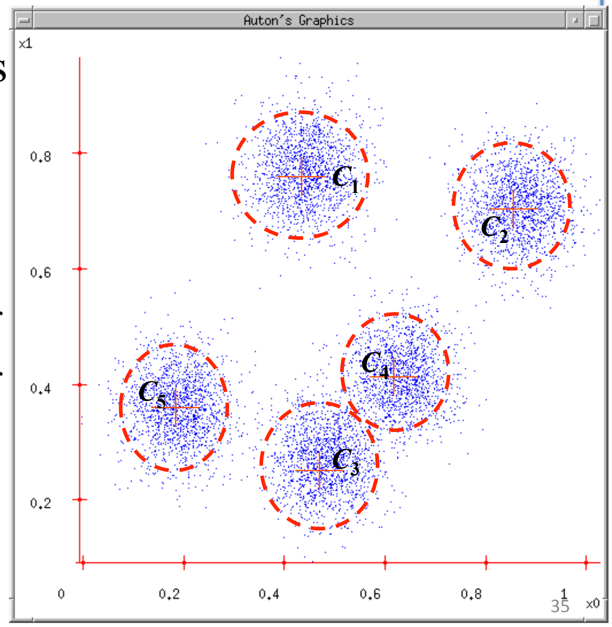
- Minimize the sum of distance within clusters

$$\arg\min_{\{\vec{C}_j, m_{i,j}\}} \sum_{j=1}^{K=6} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j\right)^2$$

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

$$\sum_{j=1}^{6} m_{i,j} = 1$$

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



How to Efficiently Cluster Data?

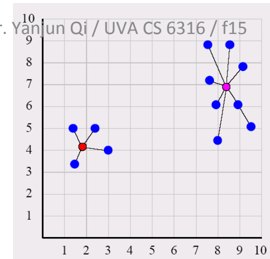
$$\arg\min_{\{\vec{C}_j, m_{i,j}\}} \sum_{j=1}^{6} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j\right)^2$$

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

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

$$\text{Given memberships } \{m_{i,j}\},\ \vec{C}_j = \frac{\sum_{i=1}^{n} m_{i,j}\vec{x}_i}{\sum_{i=1}^{n} m_{i,j}}$$

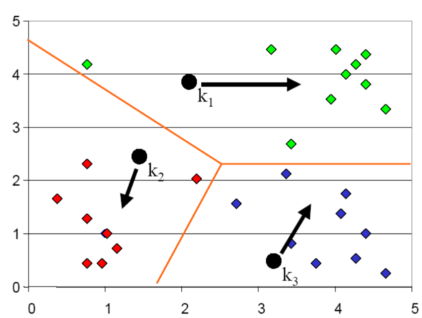
Convergence



- 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.

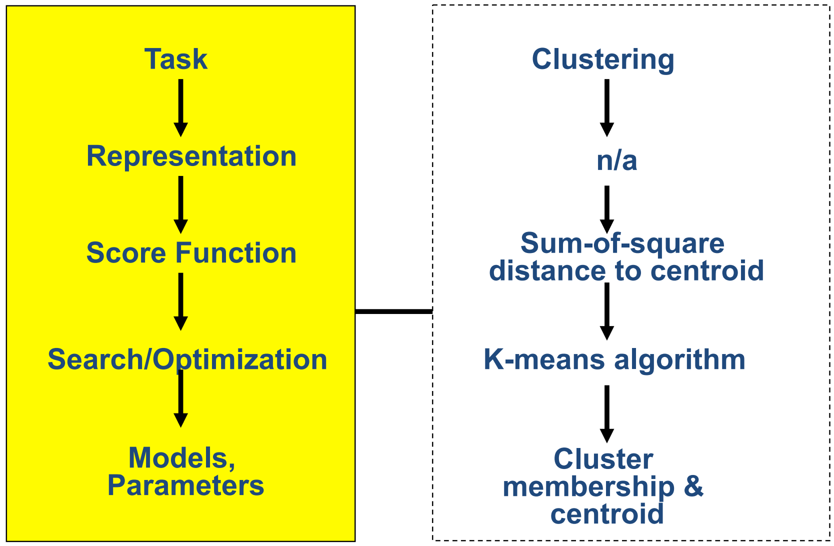
Seed Choice

- Results can vary based on random seed selection.



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

(2) K-means Clustering



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

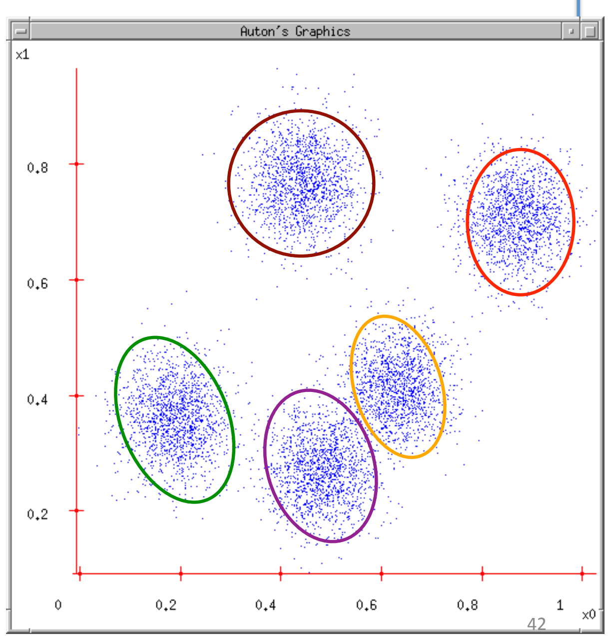
Other partitioning Methods

- 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)

A Gaussian Mixture Model for Clustering

- Assume that data are generated from a mixture of Gaussian distributions
- For each Gaussian distribution
  - Center: μ_i
  - Variance: $\sum_i$ (ignored in the following for simplified equations)
- For each data point
  - Determine membership

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



Learning a Gaussian Mixture

(with known covariance)

- Probability $p(x = x_i)$ j=1,…,K

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

Total low of probability

Learning a Gaussian Mixture

(with known covariance)

- Probability $p(x = x_i)$

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

$$= \sum_{\mu_j} p(\mu = \mu_j) \frac{1}{\left(2\pi\sigma^2\right)^{d/2}} \exp\left(-\frac{\left\|x_i - \mu_j\right\|_2}{2\sigma^2}\right)$$

?? ??

Assuming

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

$$\sum_{i=1}^{n} \log p(x = x_i) = \sum_i \log\left[\sum_{\mu_j} p(\mu = \mu_j) \frac{1}{\left(2\pi\sigma^2\right)^{d/2}} \exp\left(-\frac{\left\|x_i - \mu_j\right\|_2}{2\sigma^2}\right)\right]$$

- Apply MLE to find optimal parameters $\left\{p(\mu = \mu_j), \mu_j\right\}_j$ j=1,…,K

Learning a Gaussian Mixture
(with known covariance)

m_{ij} – k means

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

membership of data x_i to cluster j

$$= \frac{p(x = x_i \mid \mu = \mu_j) p(\mu = \mu_j)}{\sum_{n=1}^{k} p(x = x_i \mid \mu = \mu_n) p(\mu = \mu_j)}$$

$$= \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2} p(\mu = \mu_j)}{\sum_{n=1}^{k} e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2} p(\mu = \mu_n)}$$

Learning a Gaussian Mixture
(with known covariance)

k mean => centroid $\frac{1}{N_j} \sum_{i=1} m_{ij} x_i$

M-Step

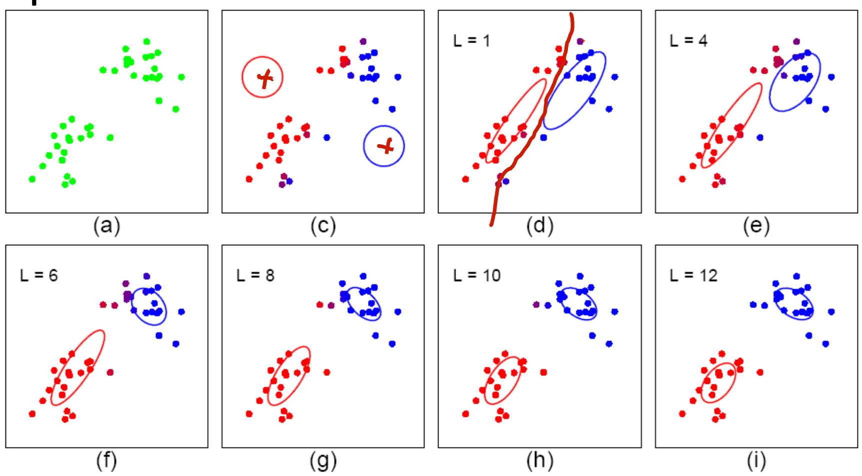
$$\mu_j \leftarrow \frac{1}{\sum_{i=1}^{n} E[z_{ij}]} \sum_{i=1}^{n} E[z_{ij}] x_i$$

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

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

Expectation-Maximization for training GMM

- Start:
  - "Guess" the centroid m_k and covariance S_k of each of the K clusters
- Loop each cluster, revising both the mean (centroid position) and covariance (shape)



Recap: K-means iterative learning

$$\arg\min_{\{\vec{C}_j, m_{i,j}\}} \sum_{j=1}^{6} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j \right)^2$$

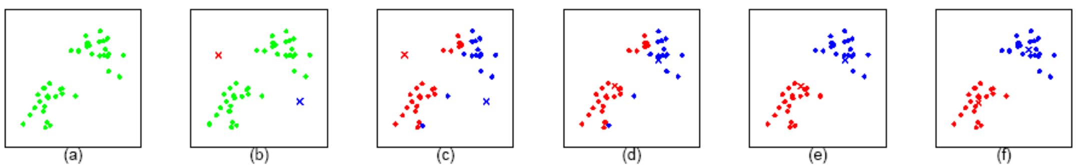
Memberships $\{m_{i,j}\}$ and centers $\{C_j\}$ are correlated.

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

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

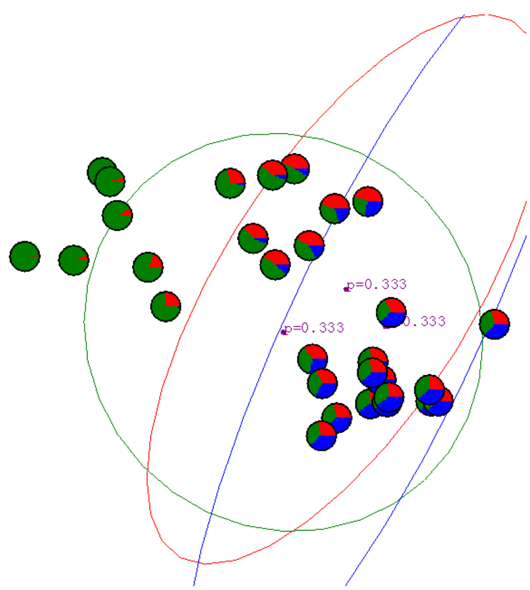
- The EM algorithm for mixtures of Gaussians is like a "soft version" of the K-means algorithm.
- In the K-means “E-step” we do hard assignment:
- In the K-means “M-step” we update the means as the weighted sum of the data, but now the weights are 0 or 1:



(a) (b) (c) (d) (e) (f)

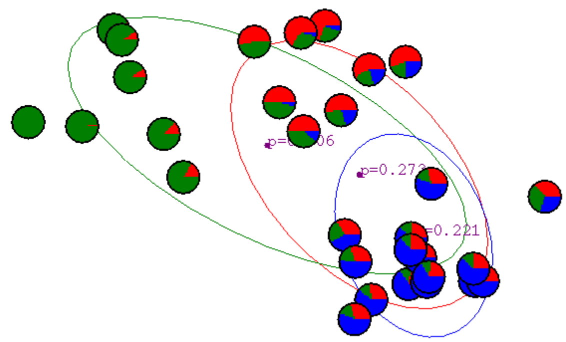
Gaussian Mixture Example: Start

$p(\mu_j|x_i)$



After First Iteration

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

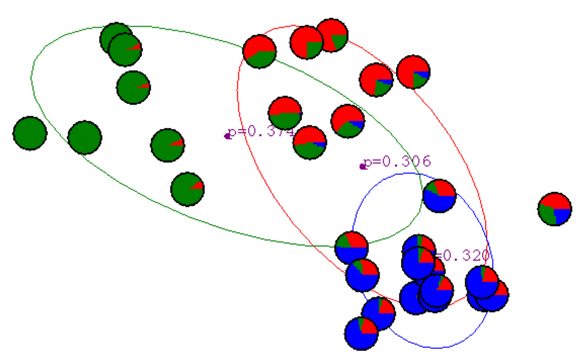


For each cluster, revising its mean (centroid position), covariance (shape) and proportion in the mixture

Dr. Yanjun Qi / UVA CS 6316 / f15

After 2nd Iteration

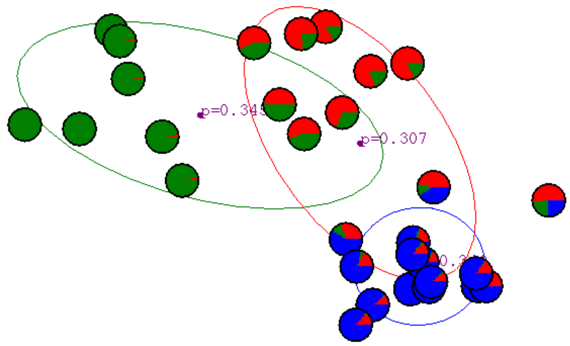
For each point, revising its proportions belonging to each of the K clusters



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

After 3rd Iteration

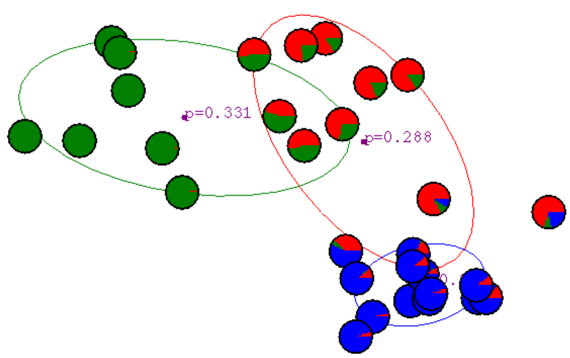
For each point, revising its proportions belonging to each of the K clusters



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

After 4th Iteration

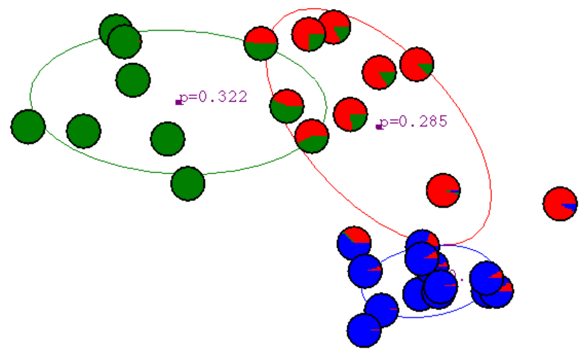
For each point, revising its proportions belonging to each of the K clusters



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

After 5th Iteration

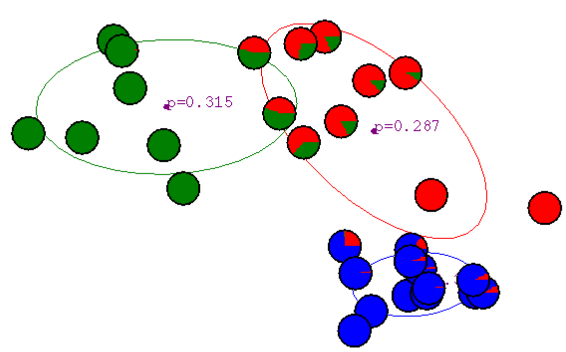
For each point, revising its proportions belonging to each of the K clusters



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

After 6th Iteration

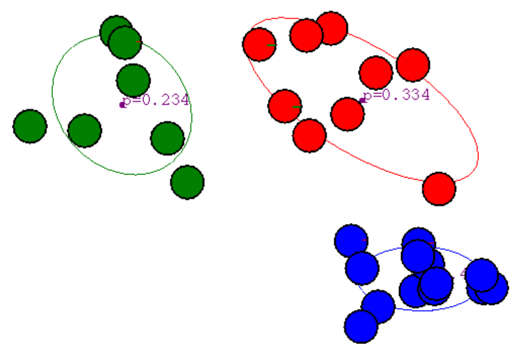
For each point, revising its proportions belonging to each of the K clusters



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 → Representation → Score Function → Search/Optimization → Models, Parameters

Clustering → Mixture of Gaussian → Likelihood → EM algorithm → Each point's soft membership & mean / covariance per cluster

$$\sum_i \log p(x = x_i) = \sum_i \log \left[\sum_{\mu_j} p(\mu = \mu_j) \frac{1}{\left(2\pi\sigma^2\right)^{d/2}} \exp\left(-\frac{\left\| x_i - \mu_j \right\|_2}{2\sigma^2} \right) \right]$$

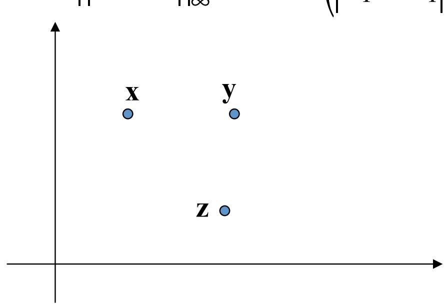
M-step (more in L23 EM lecture)

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

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

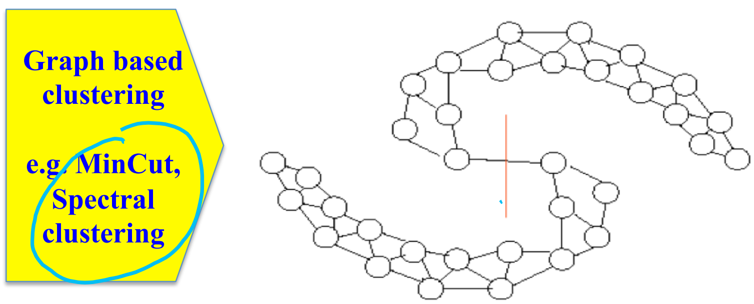


$$\|\mathbf{x} - \mathbf{y}\|_\infty = \|\mathbf{x} - \mathbf{z}\|_\infty$$

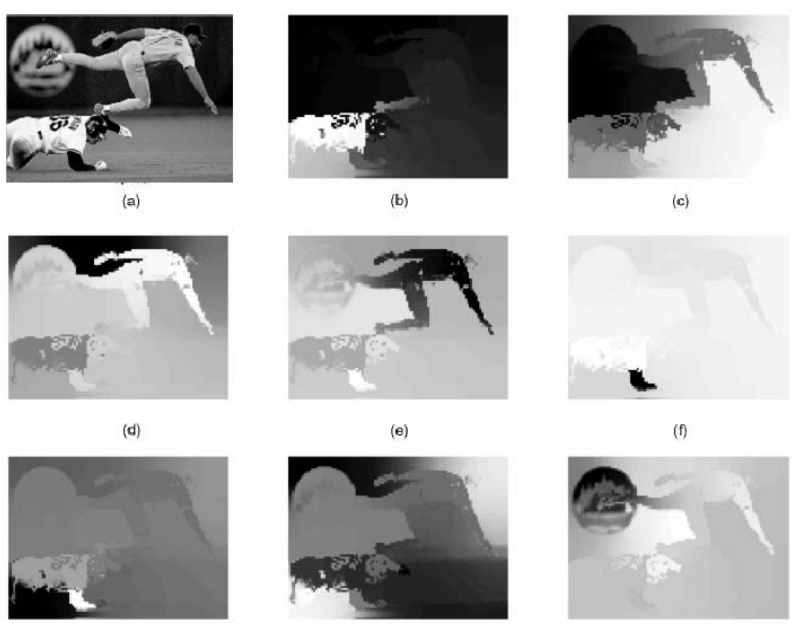
Problem (II)

tight

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



e.g. Image Segmentation through minCut

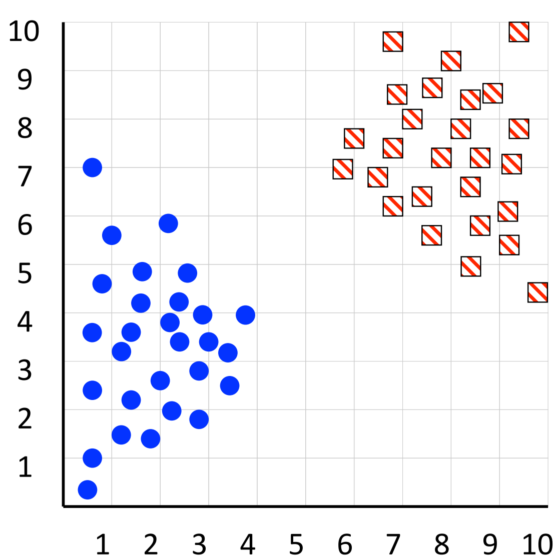


Roadmap: clustering

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

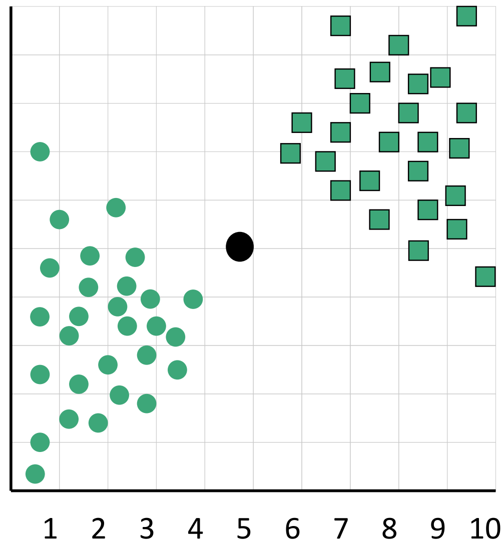
How can we tell the *right* number of clusters?

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

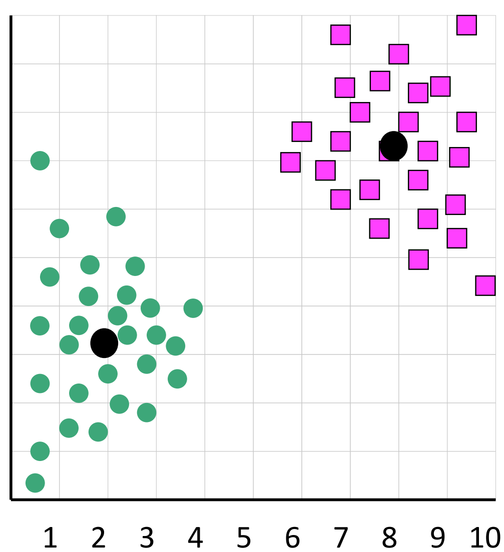


$$\underset{\{\vec{C}_j, m_{i,j}\}}{\arg\min} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \left(\vec{x}_i - \vec{C}_j \right)^2$$

When k = 1, the objective function is 873.0

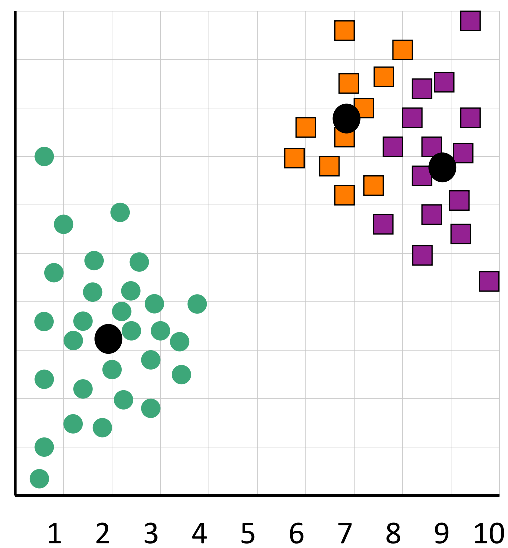


When k = 2, the objective function is 173.1



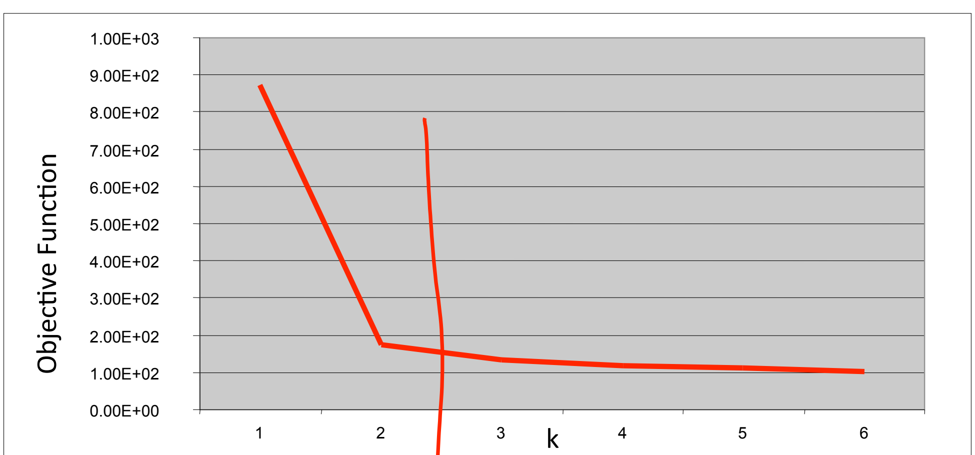
When k = 3, the objective function is 133.6

$K = n, obj = 0$



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 Within
  - the inter-class similarity is low between
  - 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)

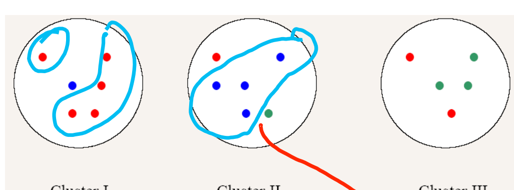
External Evaluation of Cluster Quality, e.g. using purity

- 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



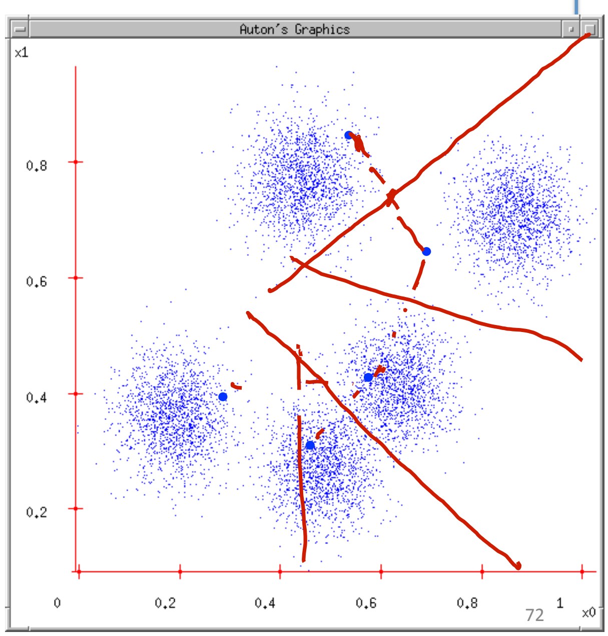
Cluster I: Purity = 1/6 (max(5, 1, 0)) = 5/6
Cluster II: Purity = 1/6 (max(1, 4, 1)) = 4/6
Cluster III: Purity = 1/5 (max(2, 0, 3)) = 3/5

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

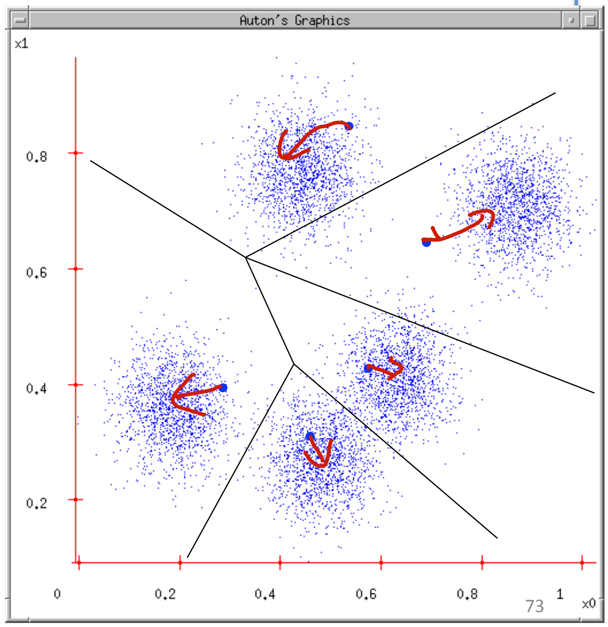
Extra practice: K-means

1. Ask user how many clusters they' d like. (e.g. k=5)
2. Randomly guess k cluster Center locations



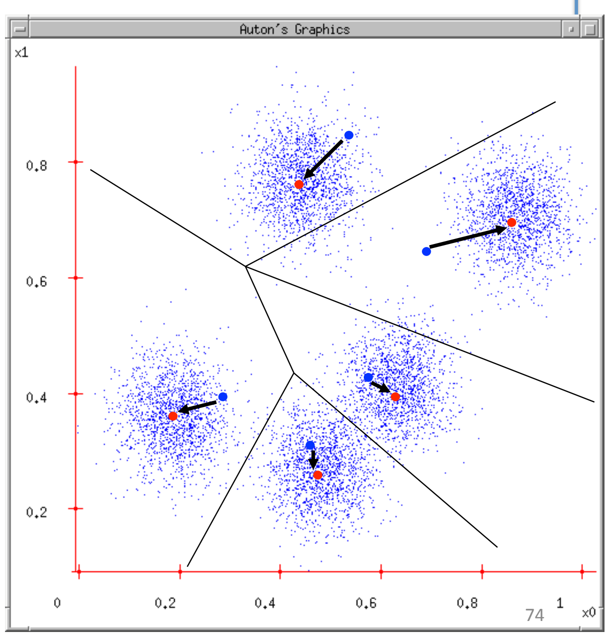
Extra practice: K-means

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. (Thus each Center "owns" a set of datapoints)



Extra practice: 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



K-means: extra practice

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