Project Proposals Due 2/24

- 1 page maximum
- Project title
- Data set (existing)
- Project Idea: approximately two paragraphs.
- Software you will need to write.
- 1-3 relevant papers.
- Teammate: 1-2 students. We expect projects done in a group to be more substantial than projects done individually.
- March 31st milestone: Some experimental results expected.

  Describe what portion of the project each partner will be doing.

Model Selection Recap…

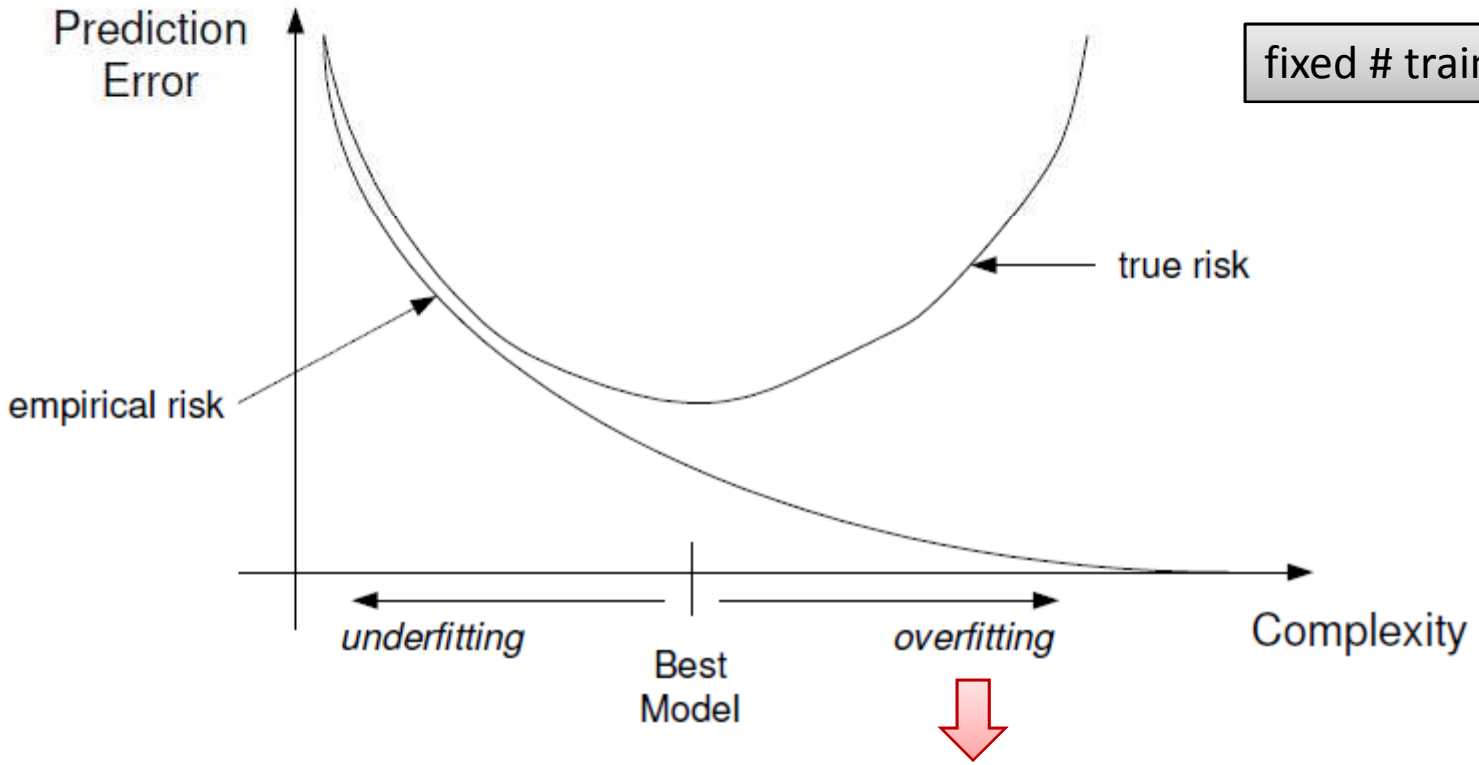
Aarti Singh

Machine Learning 10-701/15-781
Feb 8, 2010

MACHINE LEARNING DEPARTMENT
Carnegie Mellon.
School of Computer Science

Effect of Model Complexity

If we allow very complicated predictors, we could overfit the training data.

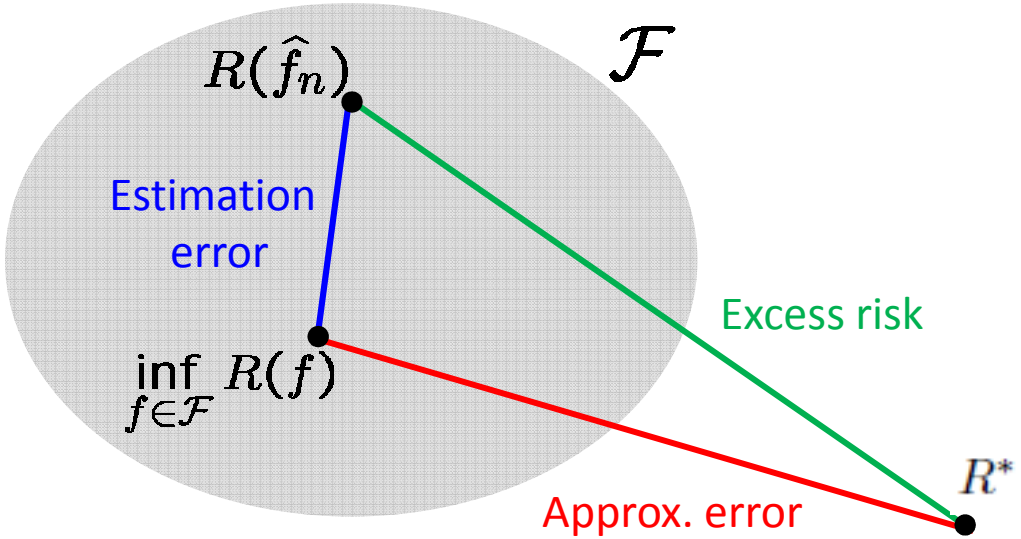


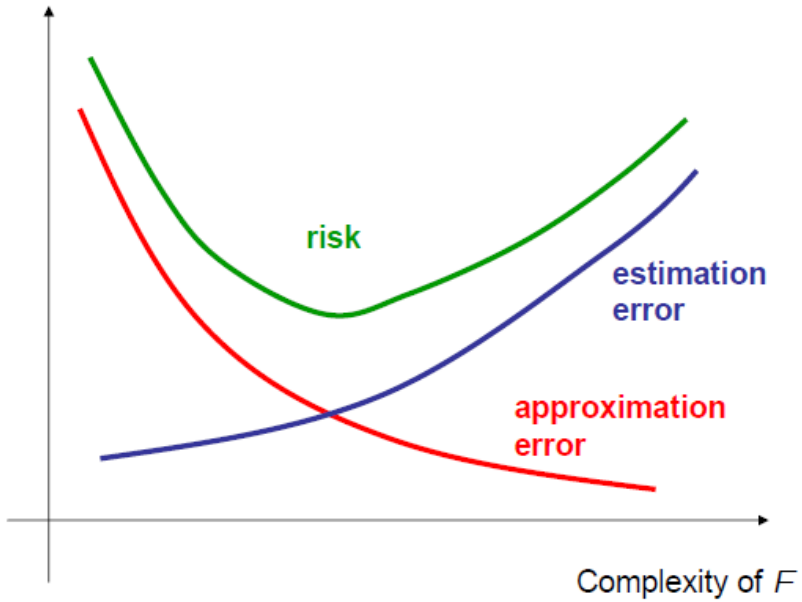
Empirical risk is no longer a good indicator of true risk

Behavior of True Risk

Training data ⇨ **Learning algorithm** ⇨ Prediction rule $\hat{f}_n$

Want predictor based on training data $\hat{f}_n$ to be as good as optimal predictor f^*





Model Selection

Setup:

Model Classes $\{\mathcal{F}_\lambda\}_{\lambda \in \Lambda}$ of increasing complexity $\mathcal{F}_1 \prec \mathcal{F}_2 \prec \ldots$

$$\min_{\lambda} \min_{f \in \mathcal{F}_\lambda} J(f, \lambda)$$

Ideally, right complexity model minimizes true risk (unknown).
But we can select the right complexity model in a data-driven/adaptive way:

- **Cross-validation** – select model with smallest validation error
- **Structural Risk Minimization** – select model that minimizes an upper bound on true risk (empirical risk + deviation bound)
- **Complexity Regularization** – select model that minimizes empirical risk + cost (- log prior)
- **Information Criteria** – select model that minimizes empirical risk + # bits needed to describe model

Information Criteria

Penalize complex models based on their **information content**.

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + C(f) \right\}$$

$C(f)$: # bits needed to describe f (description length)

MDL (Minimum Description Length)

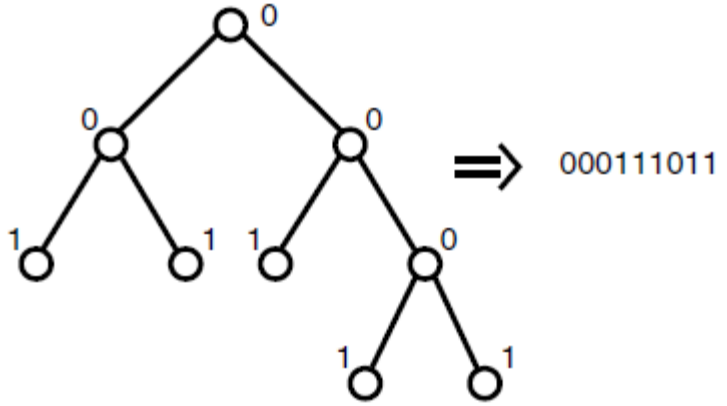
Example: Binary Decision trees $\mathcal{F}_k^T = \{\text{tree classifiers with } k \text{ leafs}\}$

$\mathcal{F}^T = \bigcup_{k \geq 1} \mathcal{F}_k^T$ prefix encode each element f of $\mathcal{F}^T$

$$C(f) = 3k - 1 \text{ bits}$$

k leaves => 2k – 1 nodes

2k – 1 bits to encode tree structure
+ k bits to encode label of each leaf (0/1)



5 leaves => 9 bits to encode structure

Information Criteria

Penalize complex models based on their **information content**.

$$\widehat{f}_n = \arg\min_{f \in \mathcal{F}} \left\{ \widehat{R}_n(f) + C(f) \right\}$$

$C(f)$ → # bits needed to describe *f* (description length)

MDL (Minimum Description Length)

Other Information Criteria:

AIC (Akiake IC) C(*f*) = # parameters

Allows # parameters to be infinite as # training data n become large

BIC (Bayesian IC) C(*f*) = # parameters * log n

Penalizes complex models more heavily – limits complexity of models as # training data n become large

Clustering

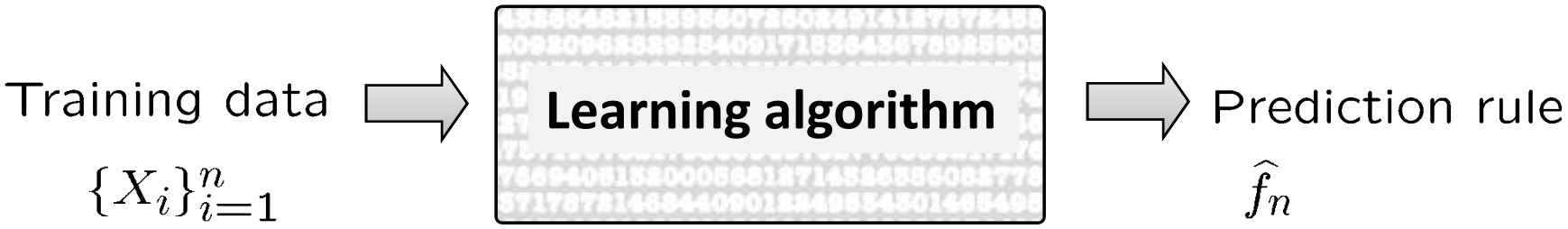
Aarti Singh

Slides courtesy: Eric Xing

Machine Learning 10-701/15-781
Feb 8, 2010

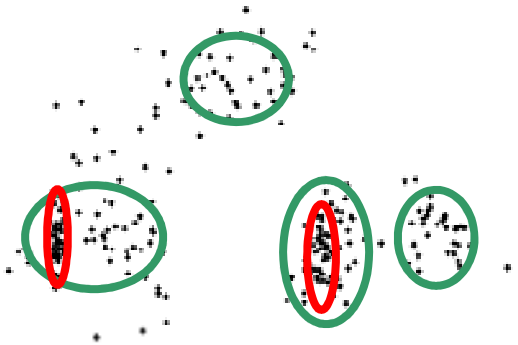
Unsupervised Learning

"Learning from unlabeled/unannotated data" (without supervision)



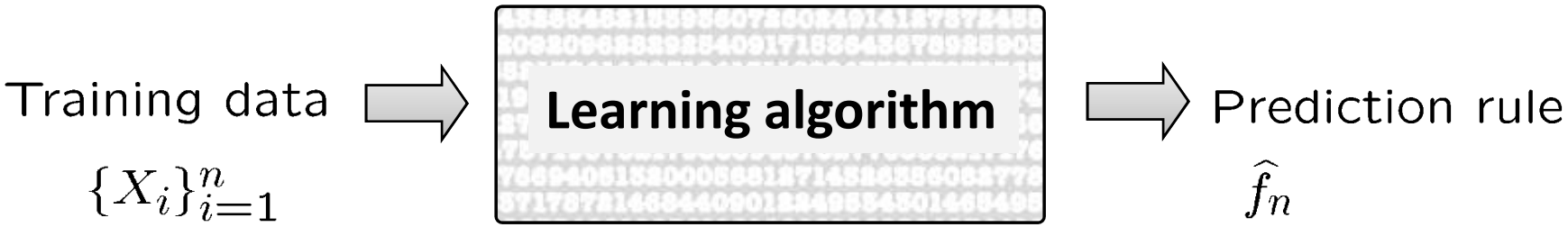
What can we predict from unlabeled data?

- Groups or clusters in the data



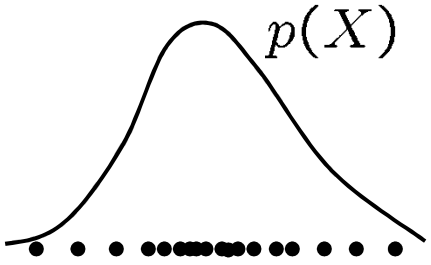
Unsupervised Learning

"Learning from unlabeled/unannotated data" (without supervision)



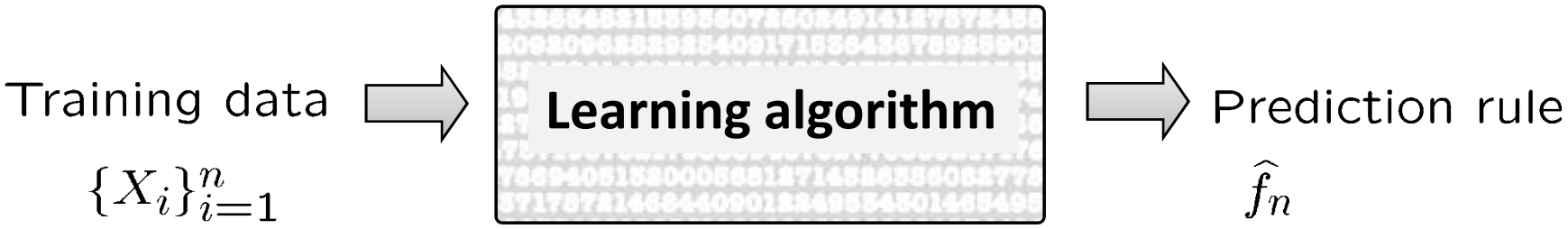
What can we predict from unlabeled data?

- Groups or clusters in the data
- Density estimation



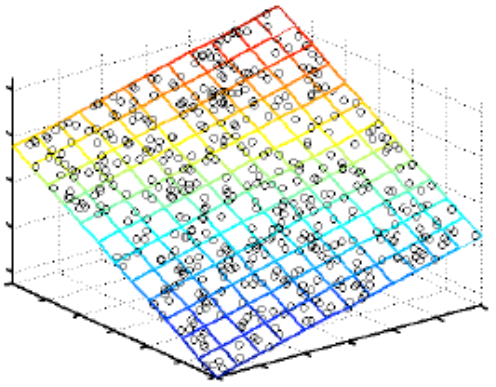
Unsupervised Learning

"Learning from unlabeled/unannotated data" (without supervision)



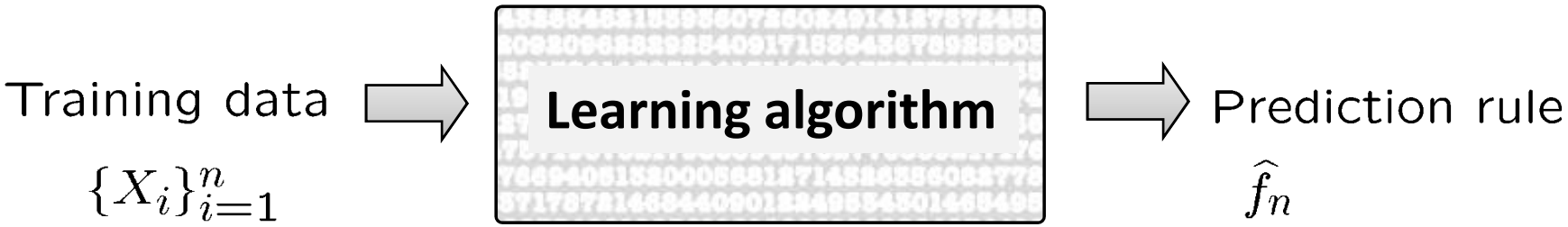
What can we predict from unlabeled data?

- Groups or clusters in the data
- Density estimation
- Low-dimensional structure
  - Principal Component Analysis (PCA) (linear)



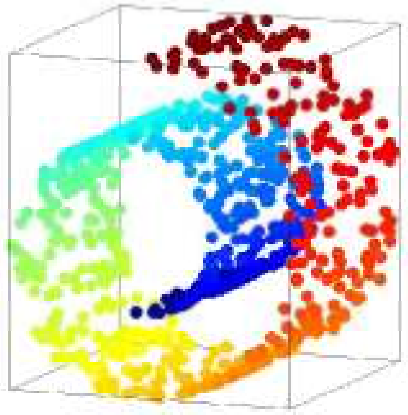
Unsupervised Learning

"Learning from unlabeled/unannotated data" (without supervision)

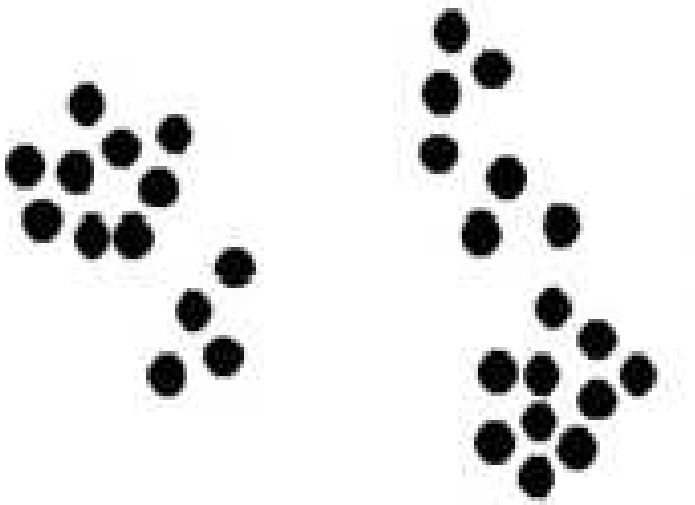


What can we predict from unlabeled data?

- Groups or clusters in the data
- Density estimation
- Low-dimensional structure
  - Principal Component Analysis (PCA) (linear)
  - Manifold learning (non-linear)



Clustering



- Are there any “groups” in the data ?
- What is each group ?
- How many ?
- How to identify them?

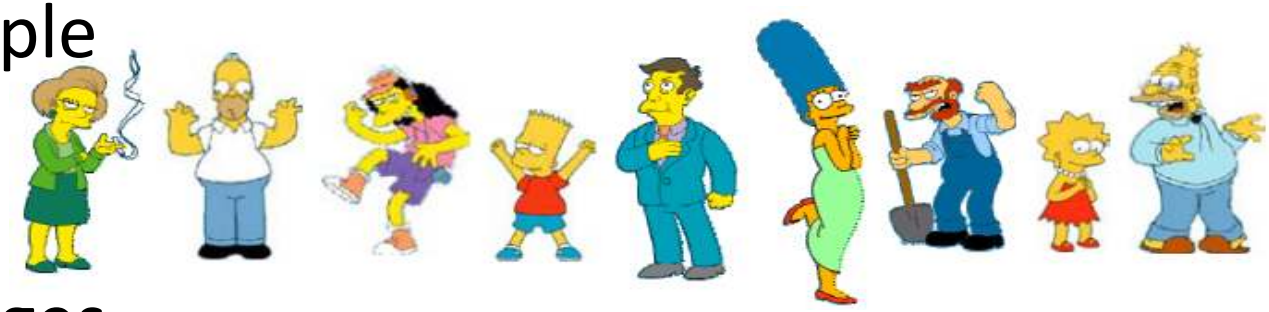
What is clustering?

- Clustering: the process of grouping a set of objects into classes of similar objects
  - high intra-class similarity
  - low inter-class similarity
  - It is the commonest form of unsupervised learning

- A common and important task that finds many applications in Science, Engineering, information Science, and other places
    - Group genes that perform the same function
    - Group individuals that has similar political view
    - Categorize documents of similar topics
    - Identify similar objects from pictures

Examples

- People



- Images

- Language

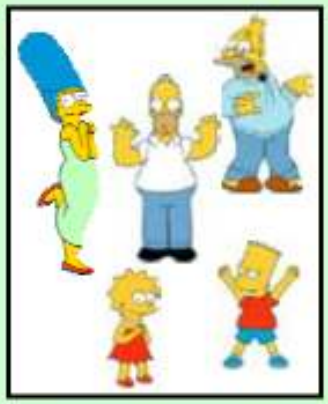
- species

Issues for clustering

- What is a natural grouping among these objects?
  - Definition of "groupness"
- What makes objects “related”?
  - Definition of "similarity/distance"
- Representation for objects
  - Vector space? Normalization?
- How many clusters?
  - Fixed a priori?
  - Completely data driven?
    - Avoid “trivial” clusters - too large or small
- Clustering Algorithms
  - Partition algorithms
  - Hierarchical algorithms
- Formal foundation and convergence

What is a natural grouping among these objects?

Clustering is subjective



Simpson's Family

School Employees

Females

Males

What is Similarity?

Hard to define!
But *we know it when we see it*

- The real meaning of similarity is a philosophical question. We will take a more pragmatic approach
- Depends on representation and algorithm. For many rep./alg., easier to think in terms of a distance (rather than similarity) between vectors.

What properties should a distance measure have?

- $D(A,B) = D(B,A)$ *Symmetry*
- $D(A,A) = 0$ *Constancy of Self-Similarity*
- $D(A,B) = 0$ IIf A= B *Positivity Separation*
- $D(A,B) \leq D(A,C) + D(B,C)$ *Triangular Inequality*

Intuitions behind desirable distance measure properties

- $D(A,B) = D(B,A)$ *Symmetry*
  - *Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex"*
- $D(A,A) = 0$ *Constancy of Self-Similarity*
  - *Otherwise you could claim "Alex looks more like Bob, than Bob does"*
- $D(A,B) = 0$ IIf A= B *Positivity Separation*
  - *Otherwise there are objects in your world that are different, but you cannot tell apart.*
- $D(A,B) \leq D(A,C) + D(B,C)$ *Triangular Inequality*
  - *Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl"*

Distance Measures: Minkowski Metric

- Suppose two object *x* and *y* both have *p* features

$$x = (x_1, x_2, \cdots, x_p)$$
$$y = (y_1, y_2, \cdots, y_p)$$

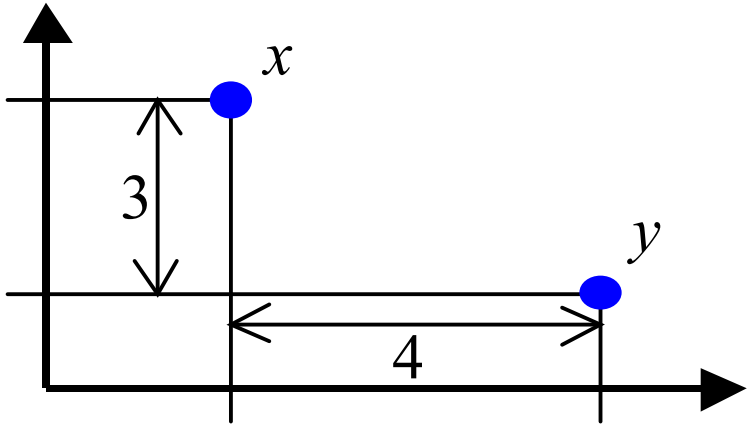
- The Minkowski metric of order *r* is defined by

$$d(x, y) = \sqrt[r]{\sum_{i=1}^{p} | x_i - y_i |^r}$$

- Most Common Minkowski Metrics

$1, r = 2$ (Euclidean distance)	$d(x, y) = \sqrt[2]{\sum_{i=1}^{p} | x_i - y_i |^2}$
$2, r = 1$ (Manhattan distance)	$d(x, y) = \sum_{i=1}^{p} | x_i - y_i |$
$3, r = +\infty$ ("sup" distance)	$d(x, y) = \max_{1 \le i \le p} | x_i - y_i |$

An Example



1 : Euclidean distance : $\sqrt[2]{4^2+3^2}=5.$

2 : Manhattan distance : $4+3=7.$

3 : "sup" distance : $\max\{4,3\}=4.$

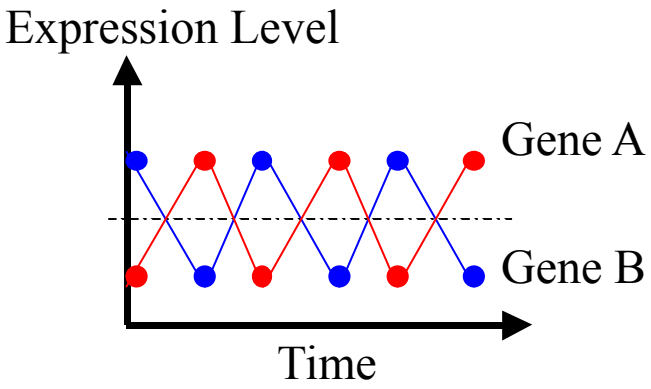
Hamming distance

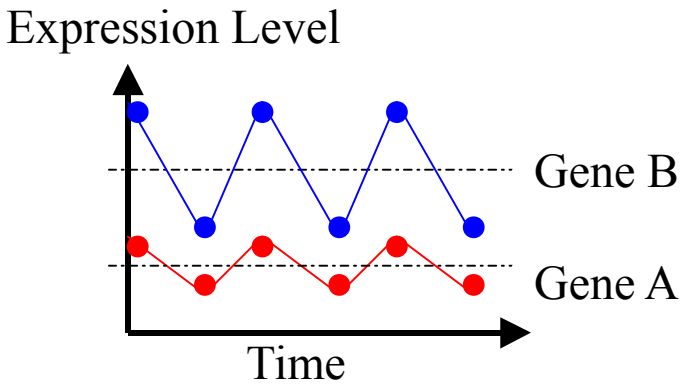
- Manhattan distance is called *Hamming distance* when all features are binary.
  - Gene Expression Levels Under 17 Conditions (1-High,0-Low)

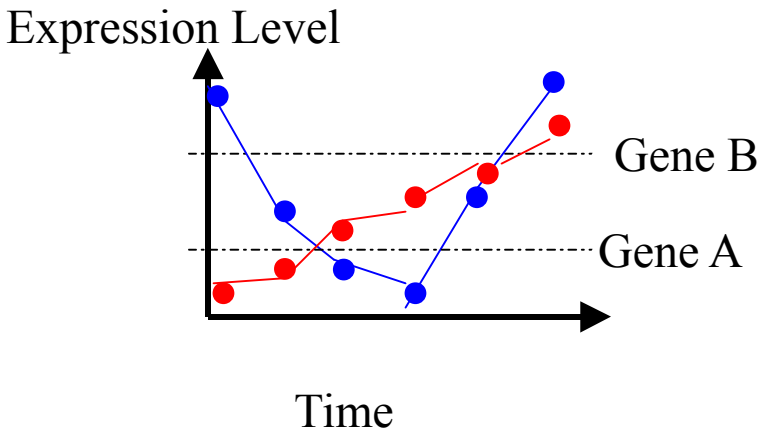
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
GeneA	0	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	1
GeneB	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	1	1

$$\text{Hamming Distance}: \#(01)+\#(10)=4+1=5.$$

Similarity Measures: Correlation Coefficient







Similarity Measures: Correlation Coefficient

- Pearson correlation coefficient

$$s(x,y) = \frac{\sum_{i=1}^{p}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{p}(x_i - \bar{x})^2 \times \sum_{i=1}^{p}(y_i - \bar{y})^2}} \qquad |s(x,y)| \leq 1$$

$$\text{where } \bar{x} = \frac{1}{p}\sum_{i=1}^{p} x_i \text{ and } \bar{y} = \frac{1}{p}\sum_{i=1}^{p} y_i .$$

Edit Distance:

A generic technique for measuring similarity

- To measure the similarity between two objects, transform one of the objects into the other, and measure how much effort it took. The measure of effort becomes the distance measure.

The distance between Patty and Selma.

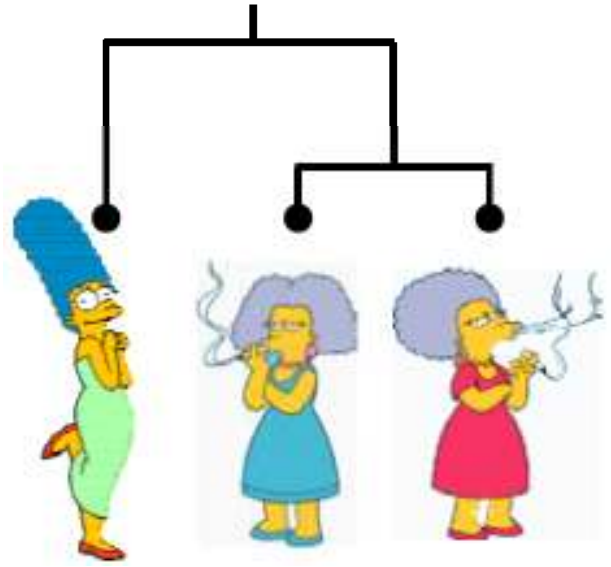
Change dress color, 1 point
Change earring shape, 1 point
Change hair part, 1 point

D(Patty,Selma) = **3**

The distance between Marge and Selma.

Change dress color, 1 point
Add earrings, 1 point
Decrease height, 1 point
Take up smoking, 1 point
Lose weight, 1 point

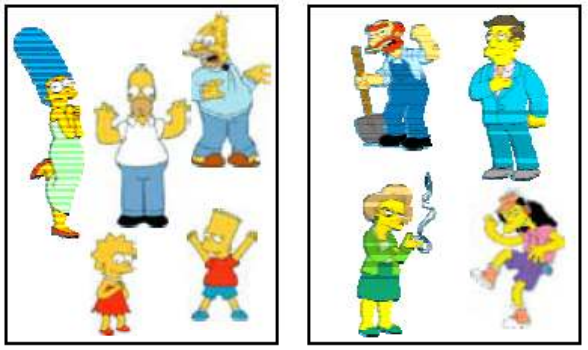
D(Marge,Selma) = **5**



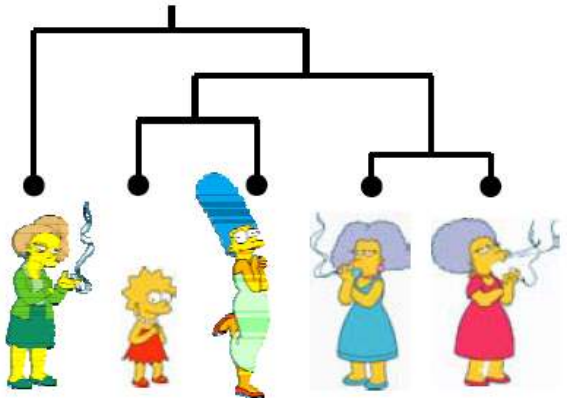
This is called the
Edit distance
or the
Transformation distance

Clustering Algorithms

- Partition algorithms
  - Usually start with a random partitioning
  - Refine it iteratively
    - K means clustering
    - Mixture-Model based clustering

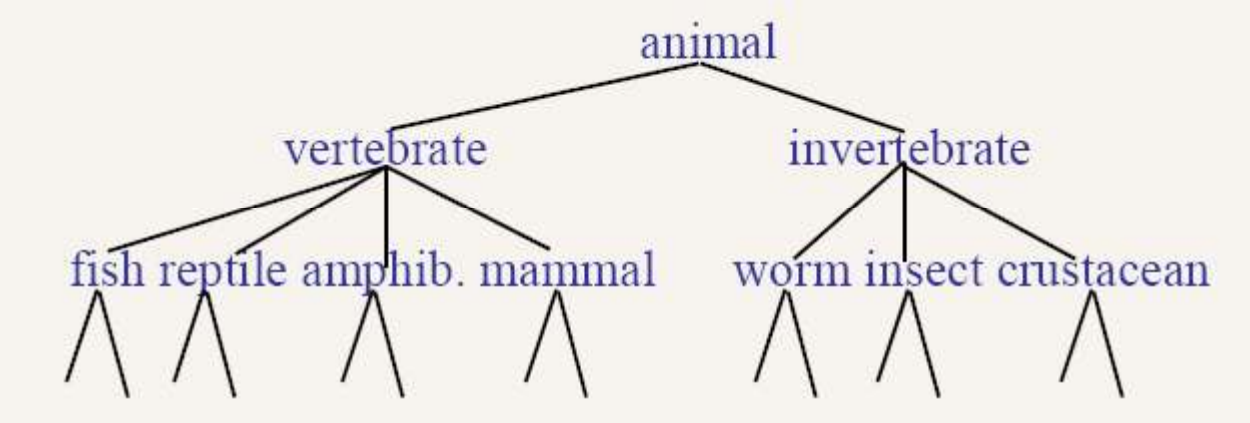


- Hierarchical algorithms
  - Bottom-up, agglomerative
  - Top-down, divisive



Hierarchical Clustering

- Build a tree-based hierarchical taxonomy (dendrogram) from a set of documents.

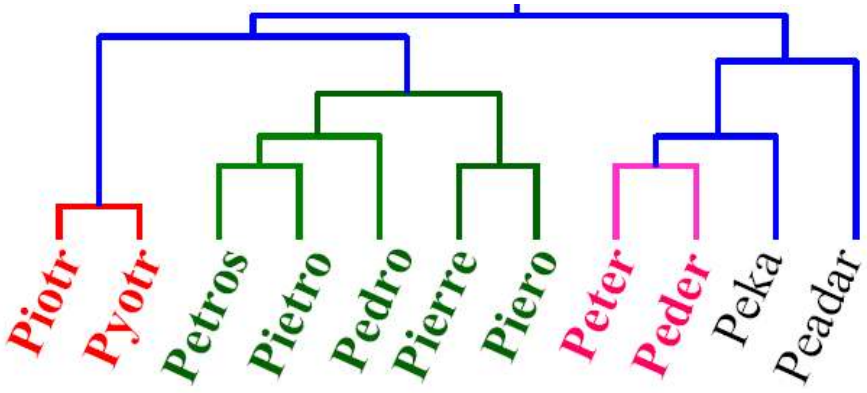


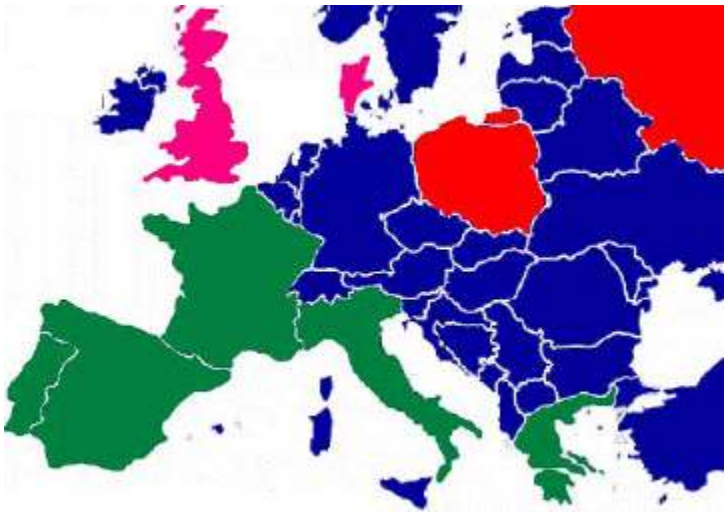
- Note that hierarchies are commonly used to organize information, for example in a web portal.
  - Yahoo! is hierarchy is manually created, we will focus on automatic creation of hierarchies in data mining.

Dendrogram

- A Useful Tool for Summarizing Similarity Measurement
  - The similarity between two objects in a dendrogram is represented as the height of the lowest internal node they share.

- Clustering obtained by cutting the dendrogram at a desired level: each connected component forms a cluster.





Hierarchical Clustering

- Bottom-Up Agglomerative Clustering
  - Starts with each obj in a separate cluster
  - then repeatedly joins the closest pair of clusters,
  - until there is only one cluster.

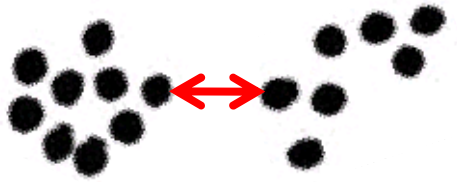
  The history of merging forms a binary tree or hierarchy.

- Top-Down divisive
  - Starting with all the data in a single cluster,
  - Consider every possible way to divide the cluster into two. Choose the best division
  - And recursively operate on both sides.

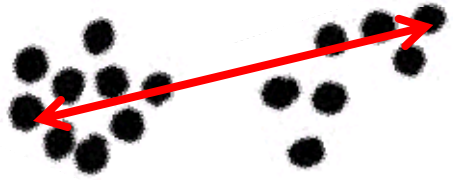
Closest pair of clusters

The distance between two clusters is defined as the distance between

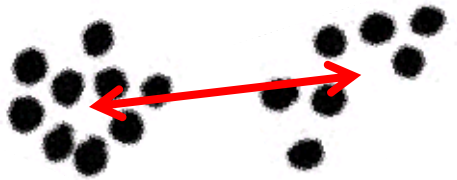
- Single-Link
  - Nearest Neighbor: their closest members.



- Complete-Link
  - Furthest Neighbor: their furthest members.



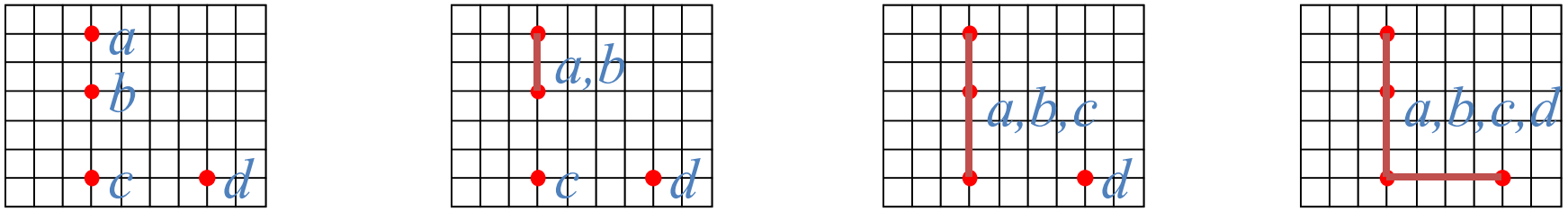
- Centroid
  - Centers of gravity



- Average-Link
  - Average of all cross-cluster pairs.

Single-Link Method

Euclidean Distance



(1) (2) (3)

	b	c	d
a	2	5	6
b		3	5
c			4

	b	c	d
a	2	5	6
b		3	5
c			4

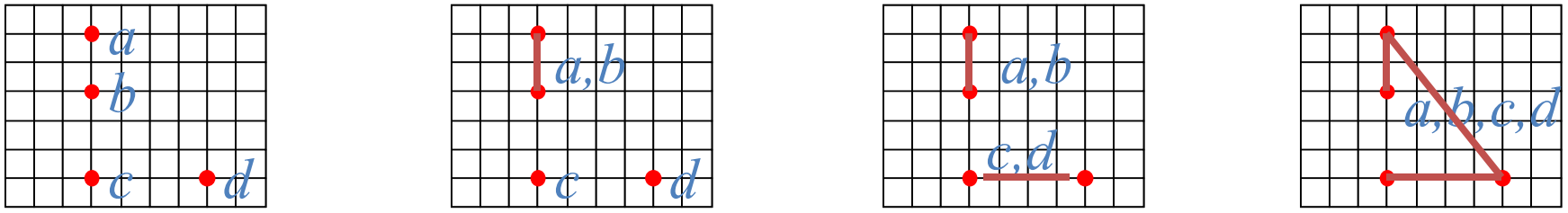
	c	d
a,b	3	5
c		4

	d
a,b,c	4

Distance Matrix

Complete-Link Method

Euclidean Distance



(1) (2) (3)

	b	c	d
a	2	5	6
b		3	5
c			4

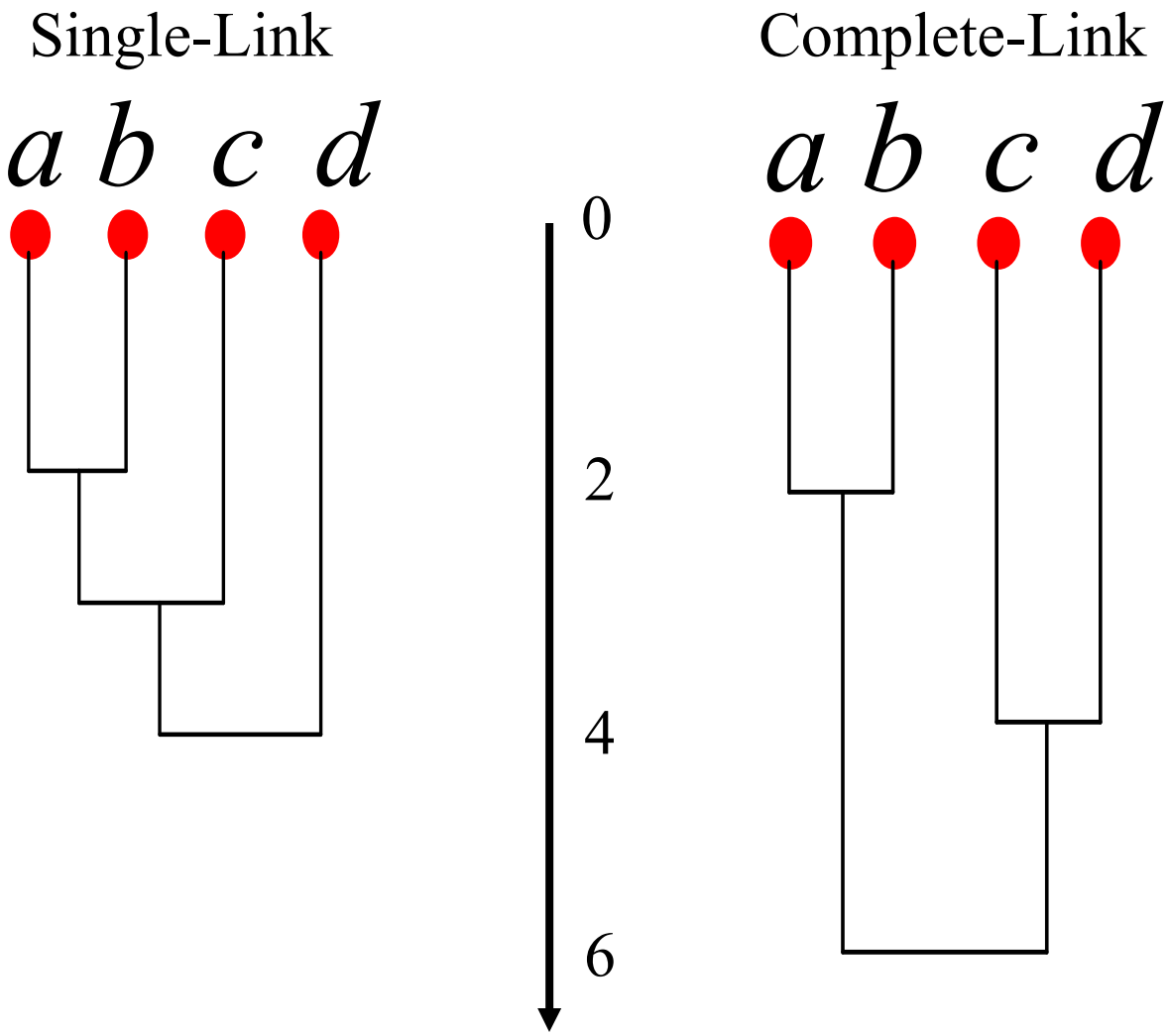
	b	c	d
a	2	5	6
b		3	5
c			4

	c	d
a,b	5	6
c		4

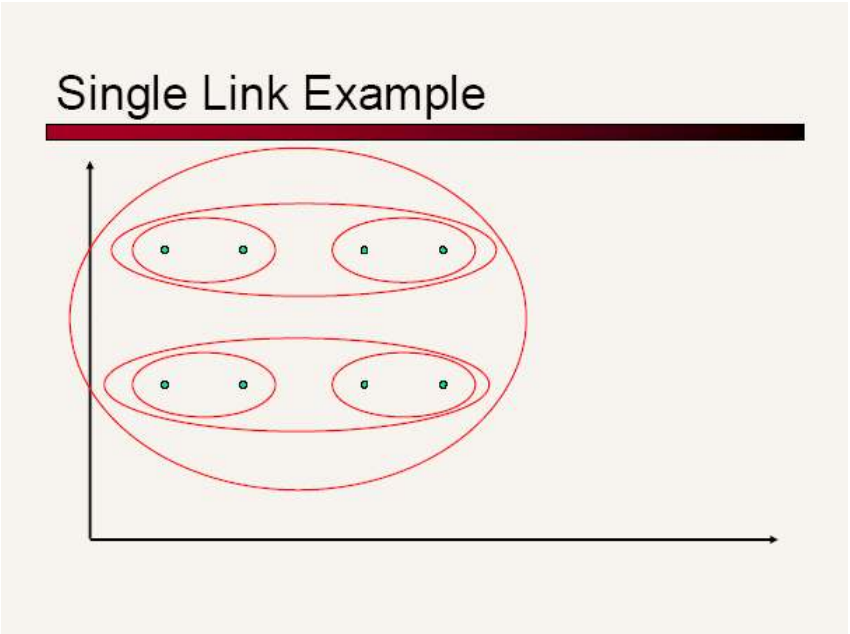
	c,d
a,b	6

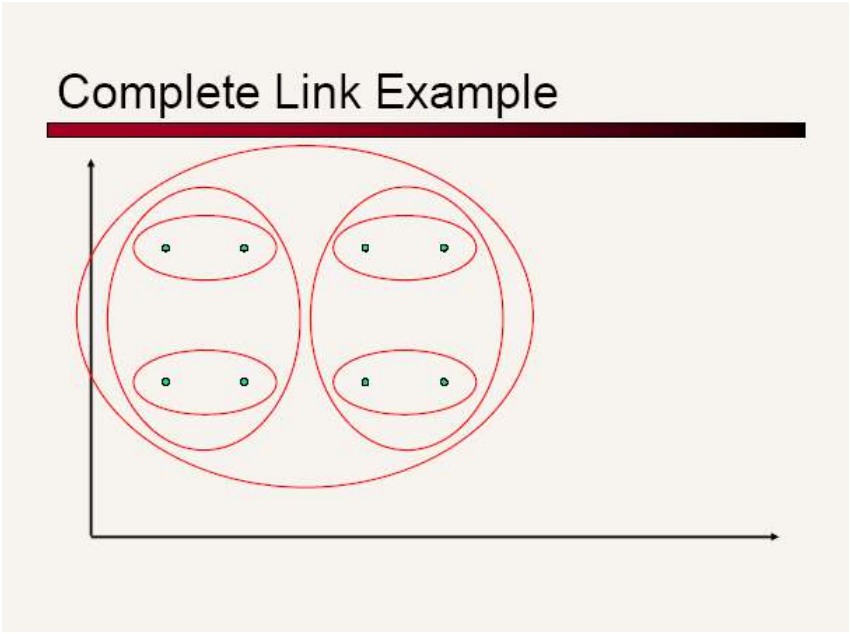
Distance Matrix

Dendrograms



Another Example





Computational Complexity

- All hierarchical clustering methods need to compute similarity of all pairs of *n* individual instances which is $O(n^2)$.
- At each iteration,
  - Sort similarities to find largest one $O(n^2 \log n)$.
  - Update similarity between merged cluster and other clusters.
- In order to maintain an overall $O(n^2)$ performance, computing similarity to each other cluster must be done in constant time.
- So we get $O(n^2 \log n)$ or $O(n^3)$ if done naively

Partitioning Algorithms

- Partitioning method: Construct a partition of *n* objects into a set of *K* clusters

- Given: a set of objects and the number *K*

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

K-Means

Algorithm

Input – Desired number of clusters, *k*

Initialize – the *k* cluster centers (randomly if necessary)

Iterate –

1. Decide the class memberships of the *N* objects by assigning them to the nearest cluster centroids (aka the center of gravity or mean)

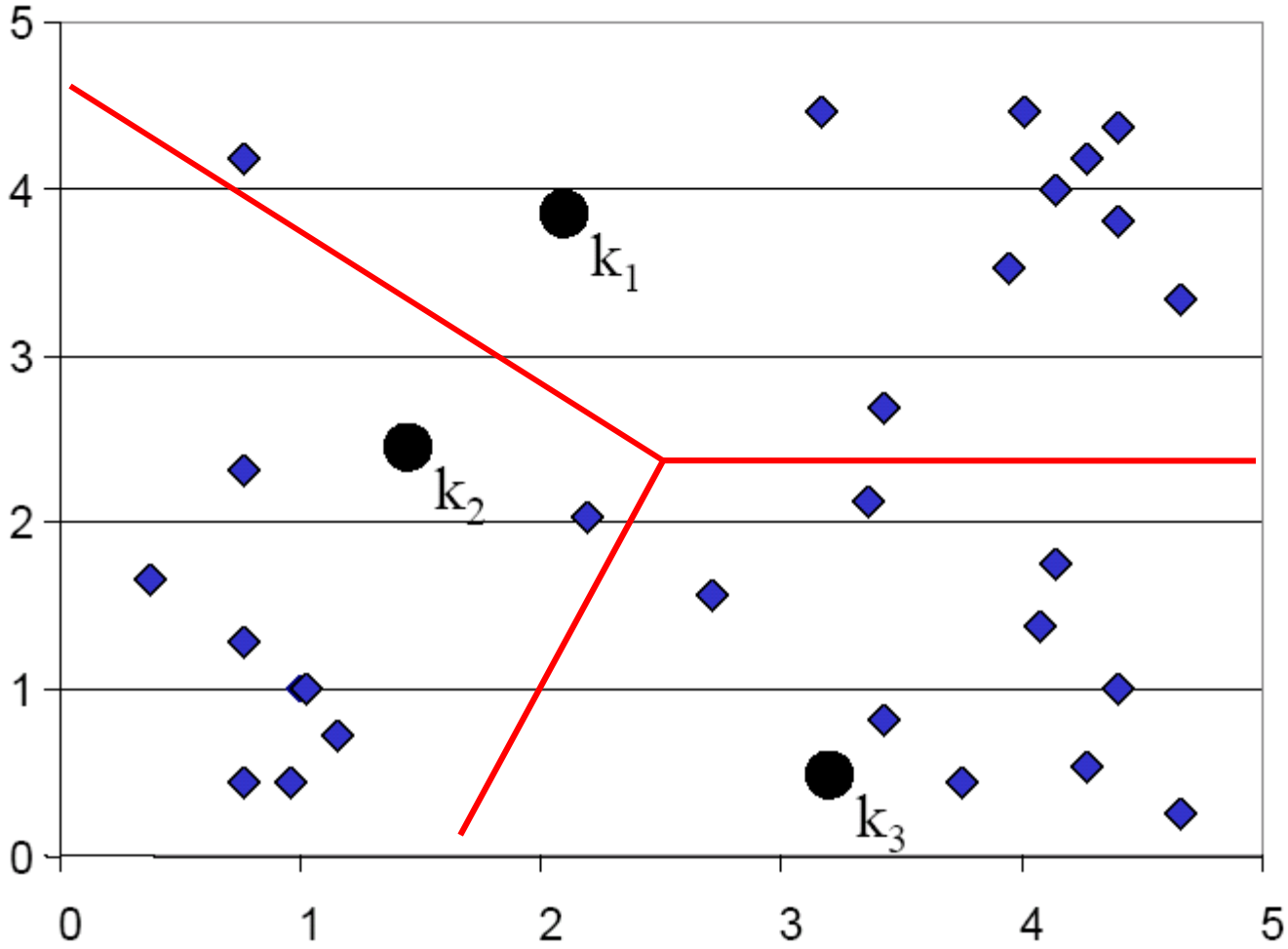
2. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.

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

Termination –

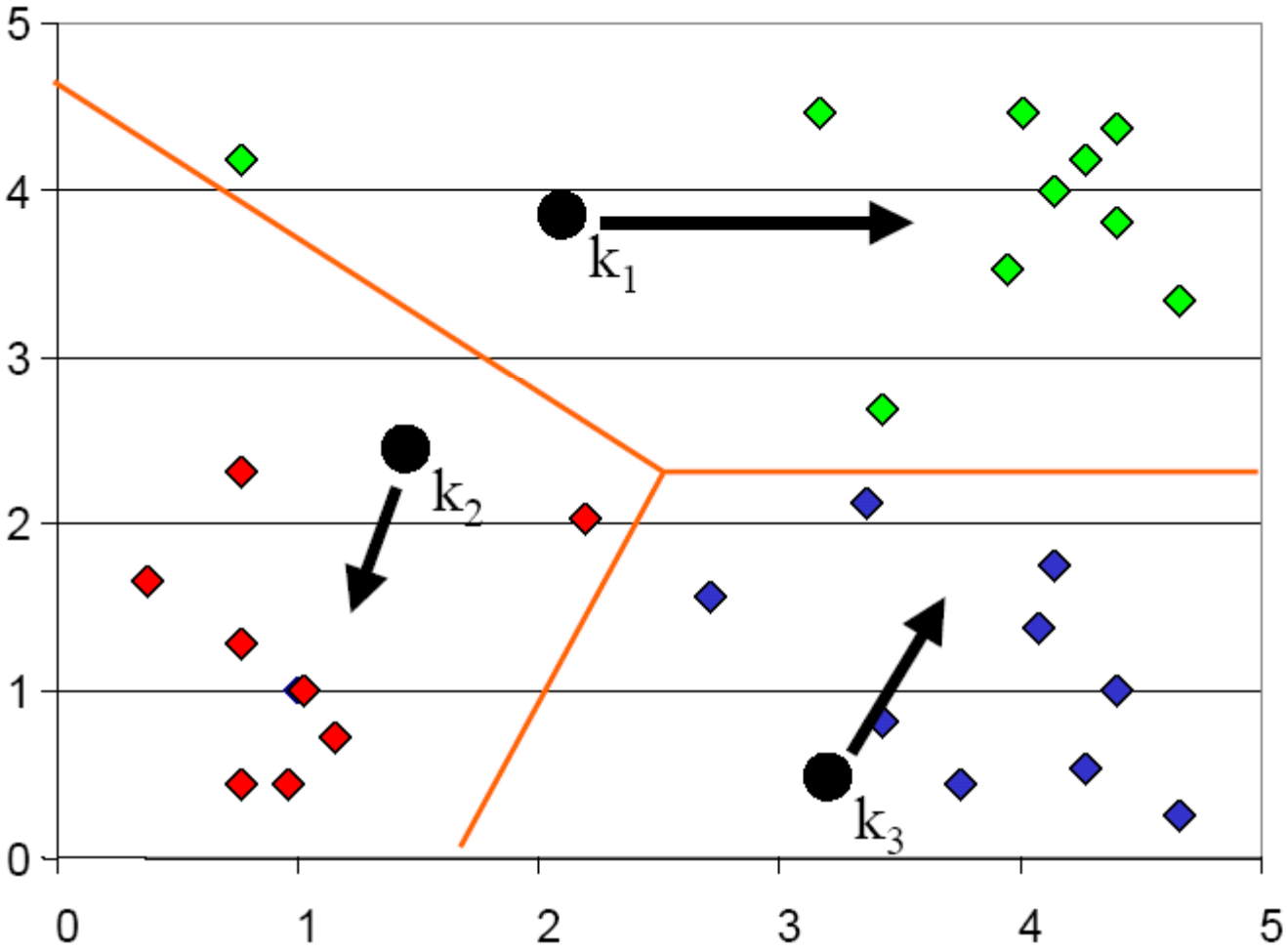
If none of the *N* objects changed membership in the last iteration, exit. Otherwise go to 3.

K-means Clustering: Step 1

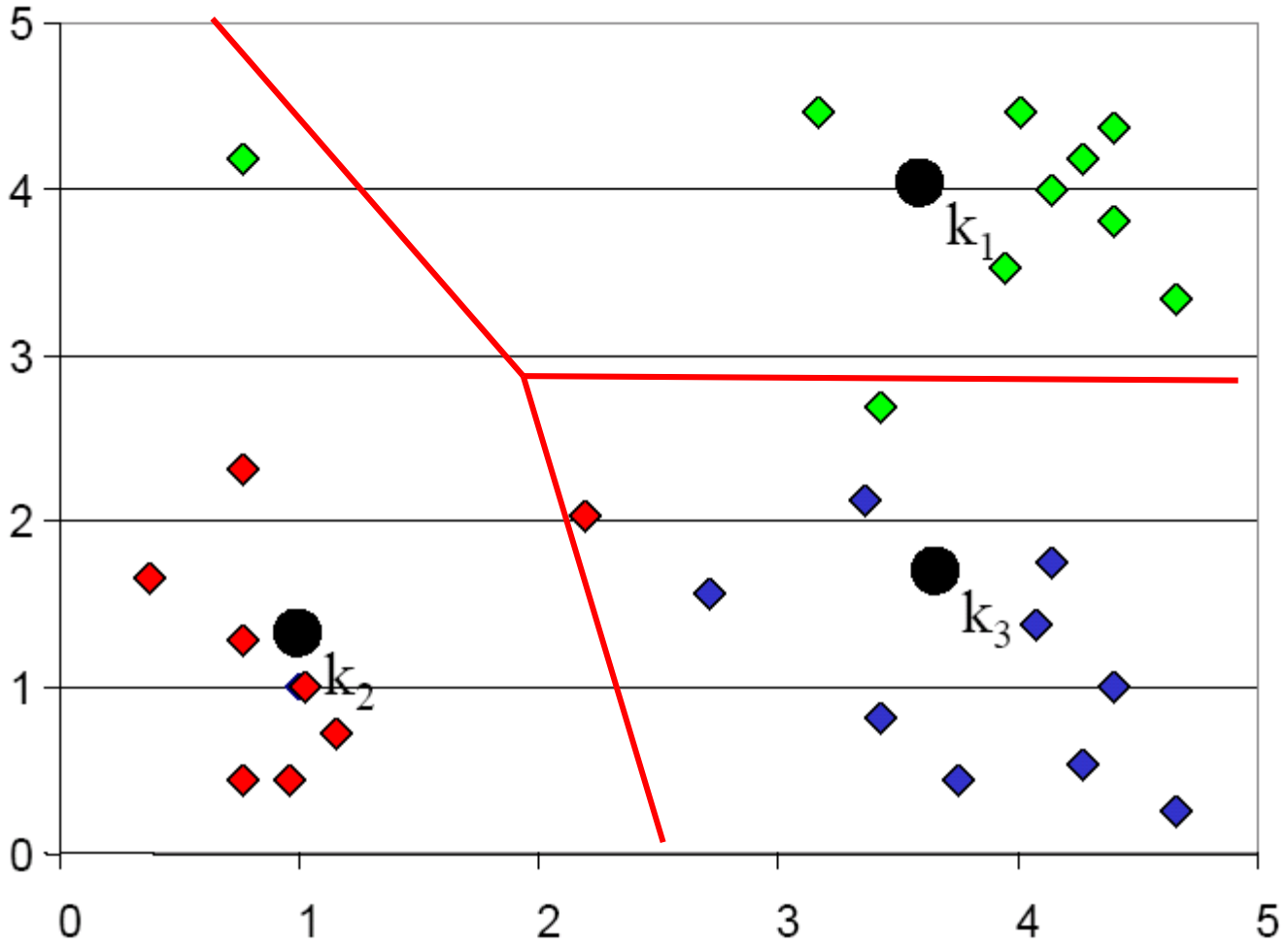


Voronoi diagram

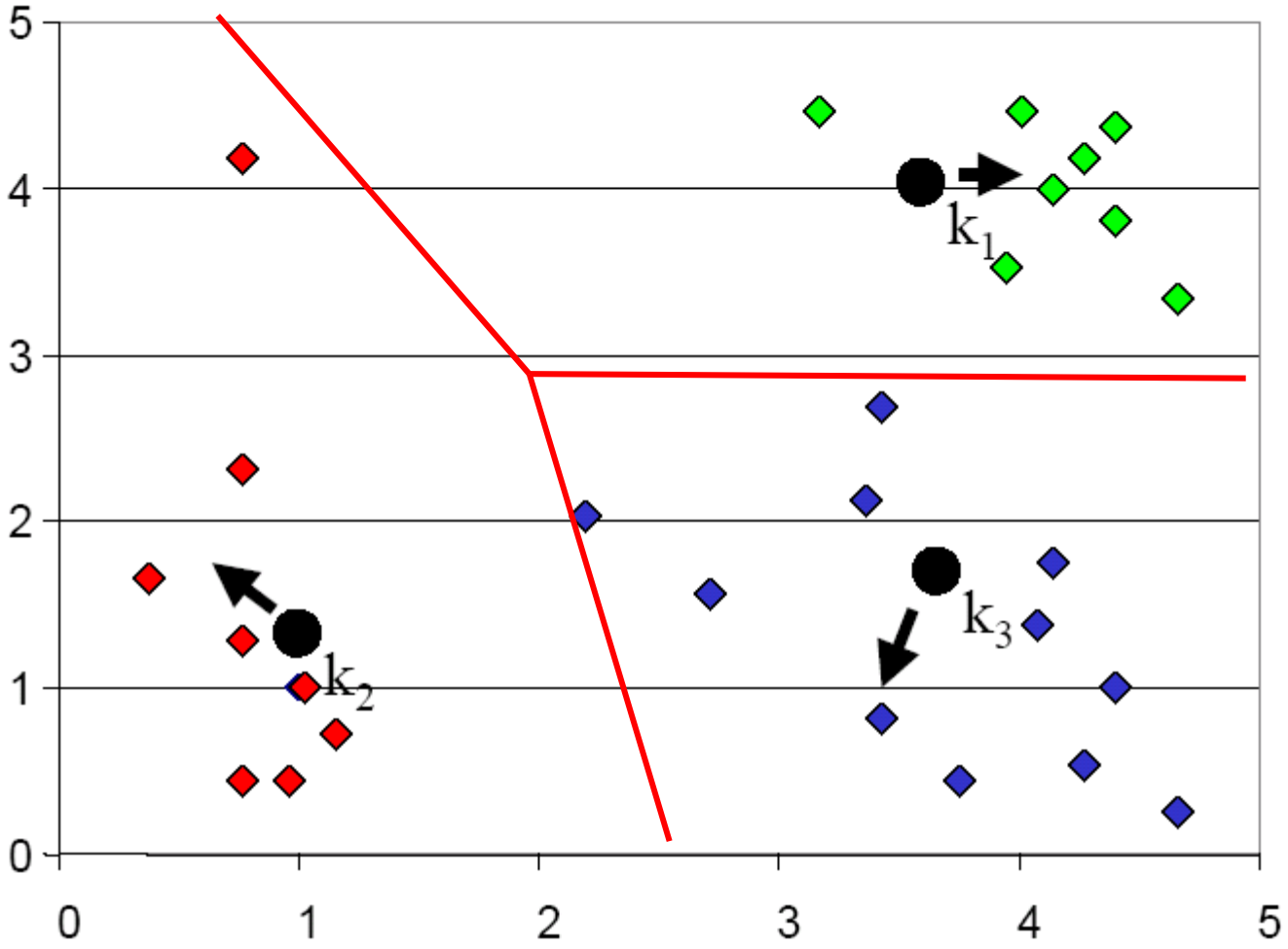
K-means Clustering: Step 2



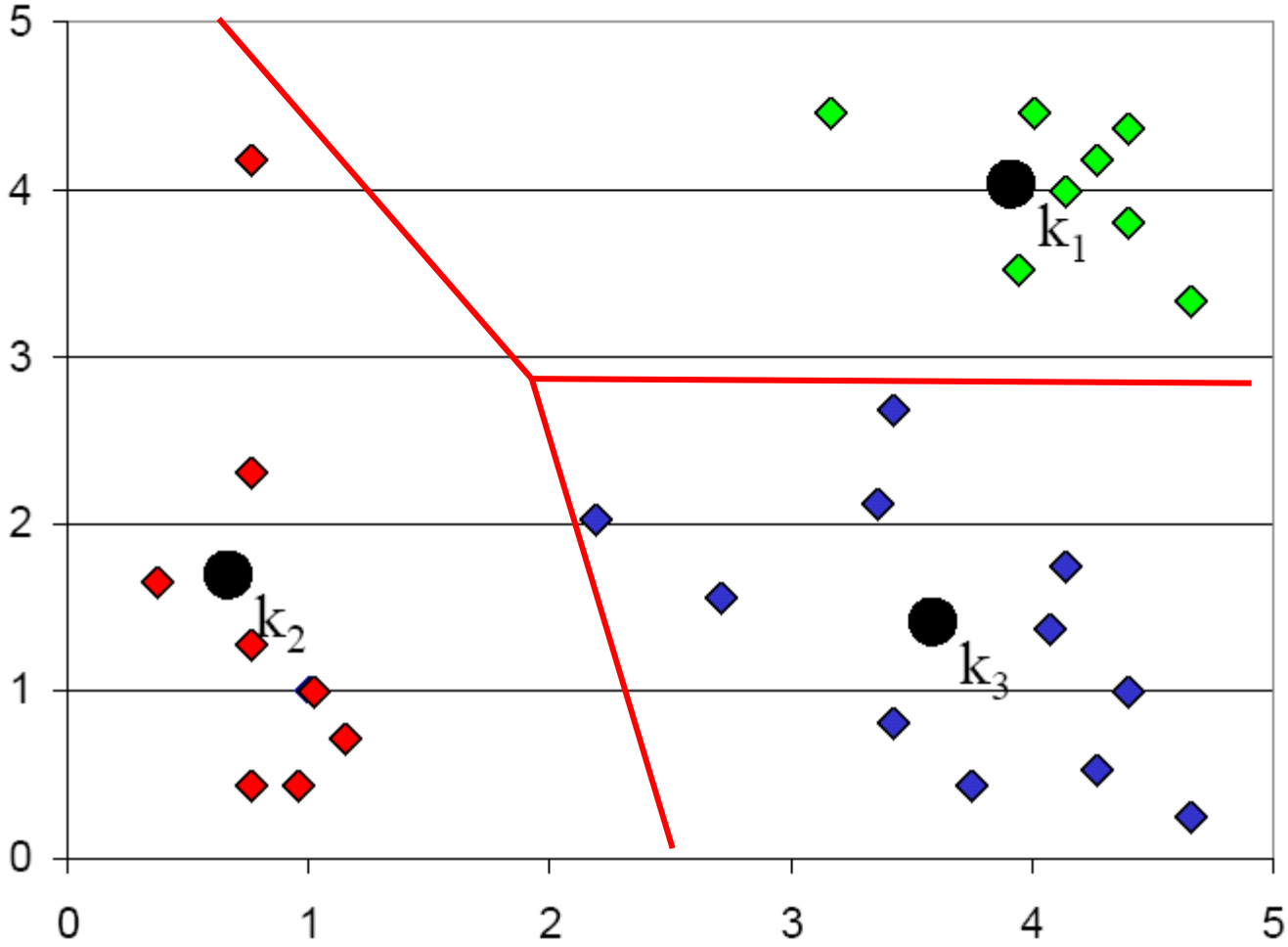
K-means Clustering: Step 3



K-means Clustering: Step 4



K-means Clustering: Step 5



Computational Complexity

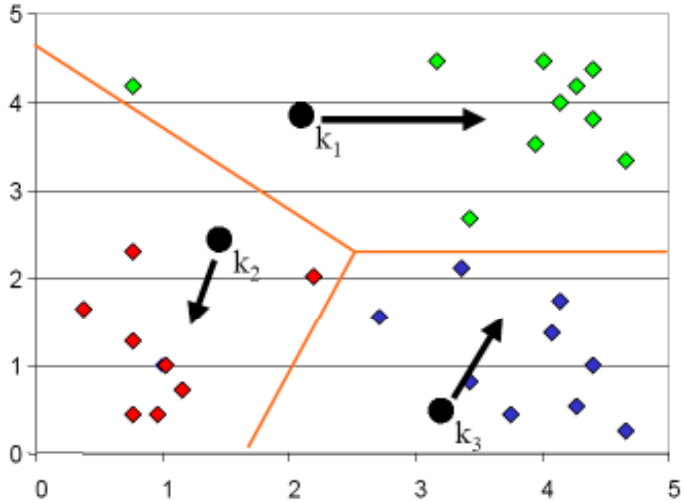
- At each iteration,
  - Computing distance between each of the n objects and the K cluster centers is O(*Kn*).
  - Computing centroids: Each object gets added once to some centroid: O(*n*).

- Assume these two steps are each done once for l iterations: O(lKn).

- Is K-means guaranteed to converge? Next class.

Seed Choice

- Results can vary based on random seed selection.



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

How Many Clusters?

- Number of clusters K is given
  - Partition n objects into predetermined number of clusters
- Finding the "right" number of clusters is part of the problem
  - Given objects, partition into an "appropriate" number of subsets.
- Solve an optimization problem: penalize having lots of clusters
  - application dependent, e.g., compressed summary of search results list.
  - Information theoretic approaches: model-based approach
- Tradeoff between having more clusters (better focus within each cluster) and having too many clusters
- Nonparametric approaches – let number of clusters scale with number of data points/objects

Clustering algorithms

- Hierarchical clustering
  - Single-linkage, complete-linkage, average-linkage
- K-means

What are the limitations of these algorithms? (Homework)

Shape of clusters

Number of clusters

Hard assignment of objects to clusters

- Mixture-based clustering
- Density-based clustering
- Soft/fuzzy clustering
- Graph-based approaches

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 object 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)

What Is A Good Clustering?

- Interesting read:

  An impossibility theorem for clustering – Kleinberg (NIPS'02)

  There exists no function mapping similarities to partition of the dataset s.t.

  - partition remains the same if all similarities are scaled by same amount (scale invariance)
  - it is capable of producing all partitions of the dataset (richness)
  - partition remains the same if all inter-cluster distances are compressed and intra-cluster distances are expanded (consistency)