# Clustering

Instructor: Jesse Davis

Slides from: Colin Dewey, Pedro Domingos, Ray Mooney, David Page, Sofus Macskassy, Dan Weld

# Announcements

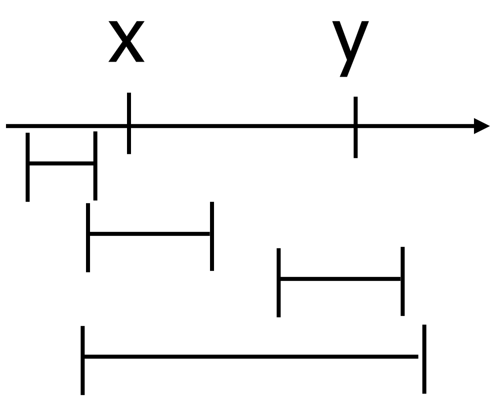
- No class final week
  - Office hours June 1st from 5:30-7:30 or 8
  - Homework 4 will be due @ midnight June 1st
- Andrey is out of town
  - He has access to email at funny times
  - Email both of us
- Clustering reading (Chapters 16+17): http://nlp.stanford.edu/IR-book/
- Lecture notes are available online

# Outline

- Homework 4: VC-Dimension problem
- Clustering

# Definition: Shattering

- A hypothesis space is said to shatter a set of instances iff for every partition of the instances into positive and negative, there is a hypothesis that produces that partition
- Example: Consider 2 instances with a single real-valued feature being shattered by intervals




| + | − |
|---|---|
| | x,y |
| x | y |
| y | x |
| x,y | |

# VC Dimensions

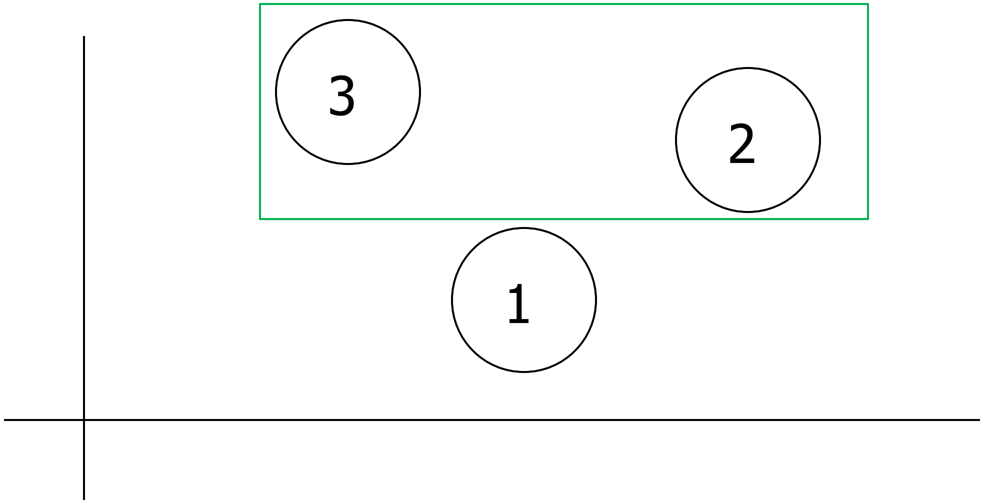
The Vapnik-Chervonenkis dimension, VC($H$). of hypothesis space $H$ defined over instance space $X$ is the size of the largest finite subset of $X$ shattered by $H$. If arbitrarily large finite subsets of $X$ can be shattered then VC($H$) = $\infty$

# Mitchell 7.5a

VC-Dim of rectangles in 2-D space

Part 1: For VC-dim, show ONE configuration of examples that can be separated regardless of labels

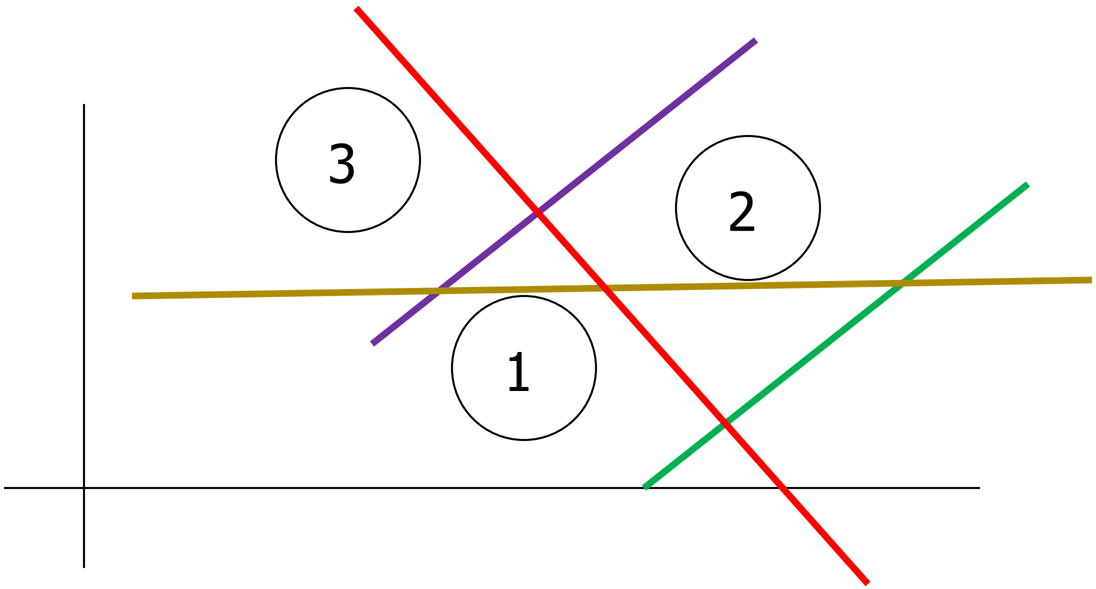
Part 2: For VC-dim+1, show that for ANY configuration of examples, there exists a labeling of the examples that can't be separated



# Example Justification

VC-dim of points in 2-D space, separated by single line

Part 1: Can classify <u>3 ex's</u> no matter how labeled




1,2 are same class

1,2,3 are same class

1,3 are same class

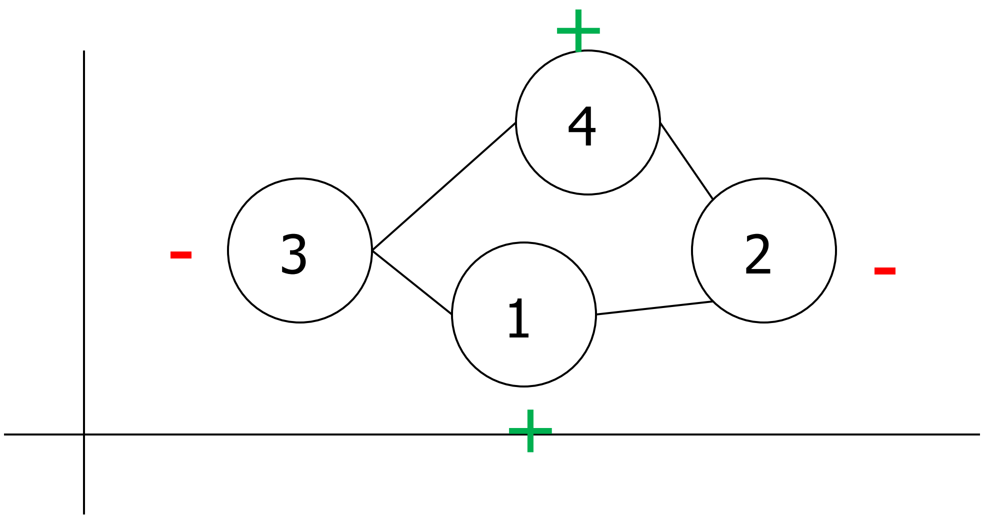
2,3 are same class

Therefore VC-dim at least 3

# Example Justification

Case 1: 3 or more points co-linear
Obviously can't label

Case 2: Other allignments
Form a regular polygon with points
Examples not connected get same label
Single line won't be able to separate (XOR)



# Outline

- Homework 4: VC-Dimension problem
- Clustering
  - Unsupervised learning, clustering intro
  - Hierarchical clustering
  - Partitional clustering
  - Model-based clustering
  - Applications

# Unsupervised Learning

- In supervised learning, we have data in the form of pairs <**x,y>, where y=f(x). The goal is to approximate f**
- In **unsupervised learning, the data just contains x!**
- The main goal is to find structure in the data
- The definition of ground truth is often missing (no clear error function, like in supervised learning)

# Uses of Unsupervised Learning

- Visualization of the data
- Data compression
- Density estimation: what distribution generated the data?
- Pre-processing step for supervised learning
- Partition data
- Novelty detection

# Unsupervised Learning: Clustering

- In many problems there are no class labels
- Humans: How do we form categories of objects?
- Humans are good at creating groups/categories/clusters from data
- Image analysis finding groups in data is very useful
  - e.g., can find pixels with similar intensities
  - e.g., can find images that are similar -> can automatically find classes/clusters of images

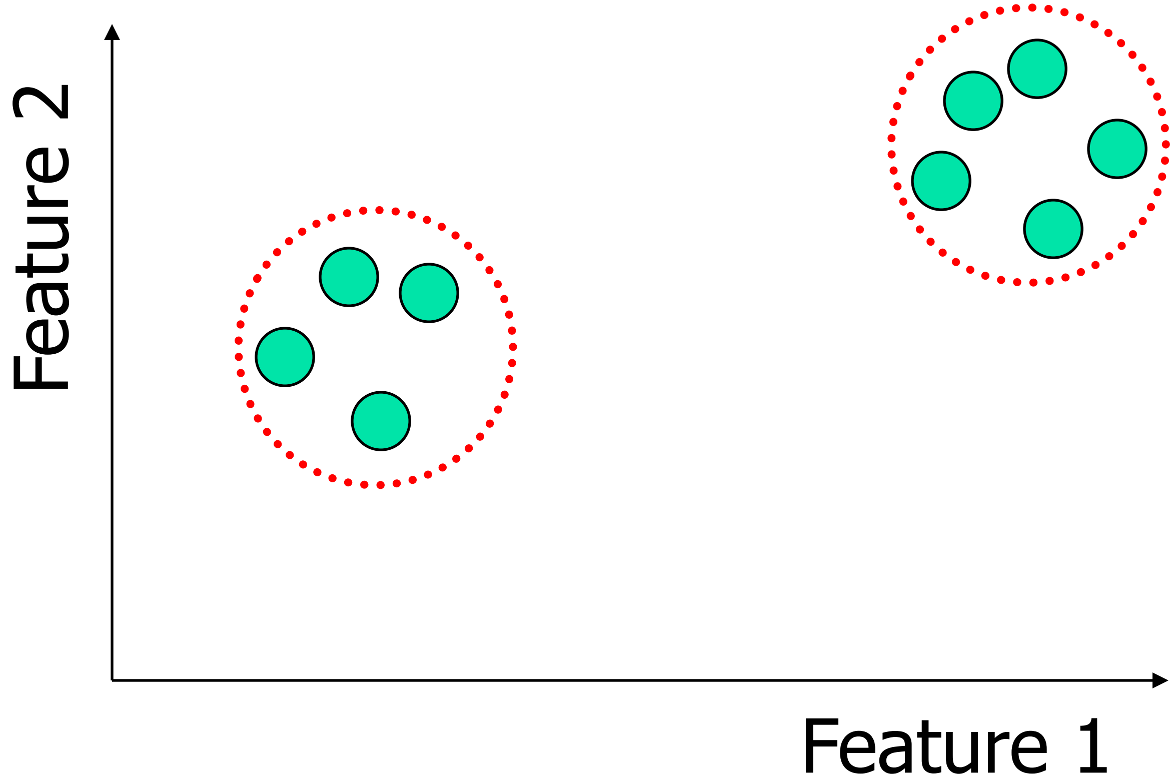
# What is Clustering

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis: Grouping objects into clusters
- Clustering is unsupervised classification
- Clusterings are usually not right or wrong
  - Different clusterings can reveal different things about the data
  - More direct measure of goodness if it is a first step towards supervised learning, or data compression

# How is Clustering Used

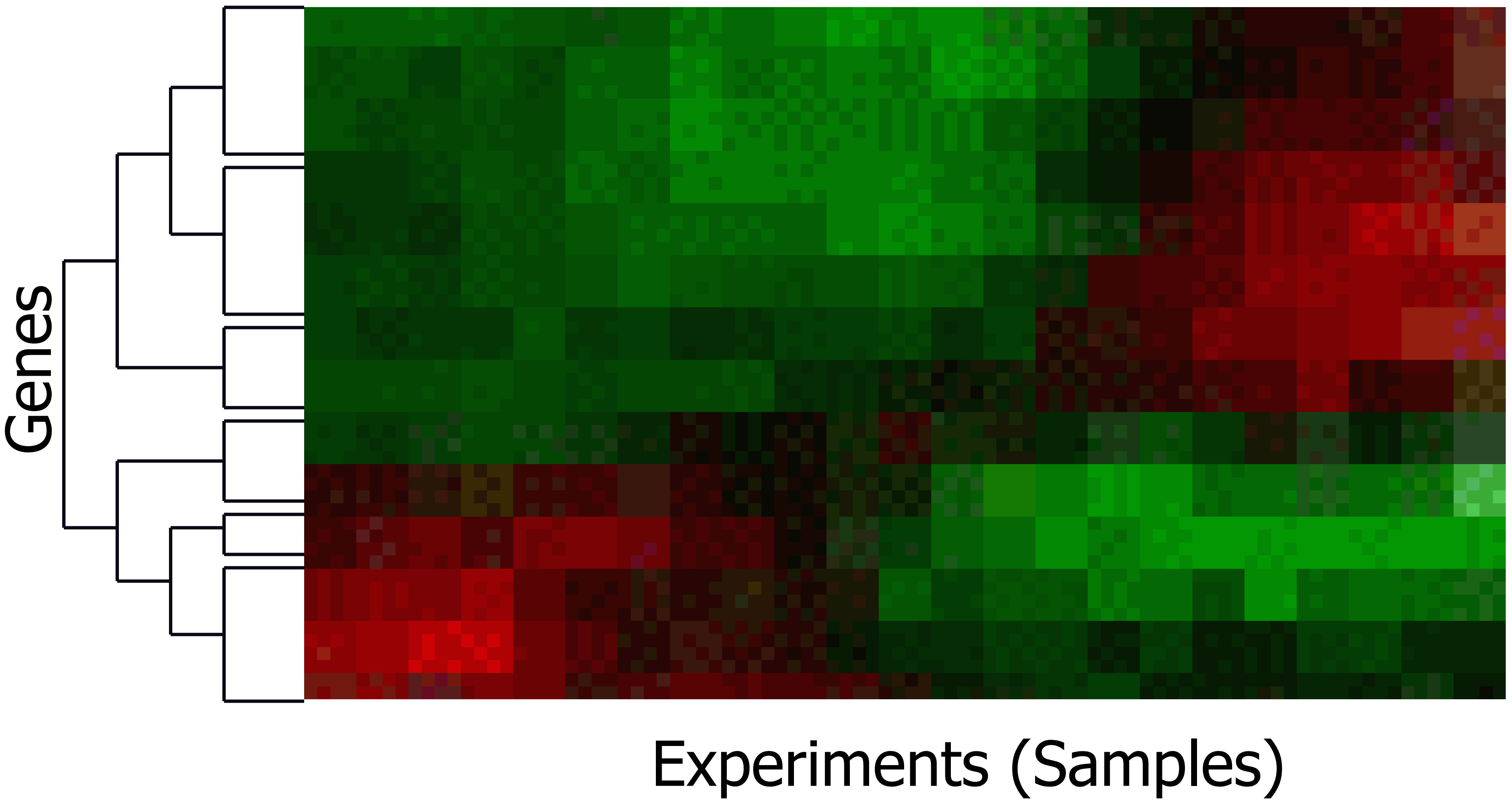
- Clustering is grouping similar objects together
  - To establish prototypes or detect outliers
  - To simplify data for further analysis/learning
  - To visualize data
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

# Example: Two Clusters



# Example: Gene Expression

(Green = up-regulated, Red = down-regulated)



# Clustering Applications

- <u>Marketing:</u> Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- <u>Land use:</u> Identification of areas of similar land use in an earth observation database
- <u>Insurance:</u> Identifying groups of motor insurance policy holders with a high average claim cost
- <u>Urban planning:</u> Identifying groups of houses according to their house type, value, and geographical location
- <u>Seismology:</u> Observed earth quake epicenters should be clustered along continent faults

# What Is a Good Clustering?

- A good clustering method will produce clusters with
  - High <u>intra-class</u> similarity
  - Low <u>inter-class</u> similarity
- Precise definition of clustering quality is difficult
  - Application-dependent
  - Ultimately subjective

# Requirements for Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal domain knowledge required to determine input parameters
- Ability to deal with noise and outliers
- Insensitivity to order of input records
- Robustness wrt high dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

# The Clustering Problem

- Let $\underline{x} = (x_1, x_2, \ldots, x_d,)$ be a d-dimensional feature vector
- Let D be a set of $\underline{x}$ vectors,
  - $D = \{ \underline{x}_1, \underline{x}_2, \ldots. \underline{x}_N \}$
- Given data D, group the N vectors into K groups such that the grouping is "optimal"

# Basic Concept: Distances/Similarities

- Clustering methods use a distance (similarity) measure to assess the distance between
  - a pair of instances
  - a cluster and an instance
  - a pair of clusters
- Given a distance value, can convert it into a similarity value: sim(i,j) = 1/[1+dist(i,j)]
- Not always straightforward to go the other way
- We'll describe our algorithms in terms of distances

# Distances Between Instances

- Same we used for IBL (e.g, $L_p$ norm)
- Euclidean distance (p = 2):

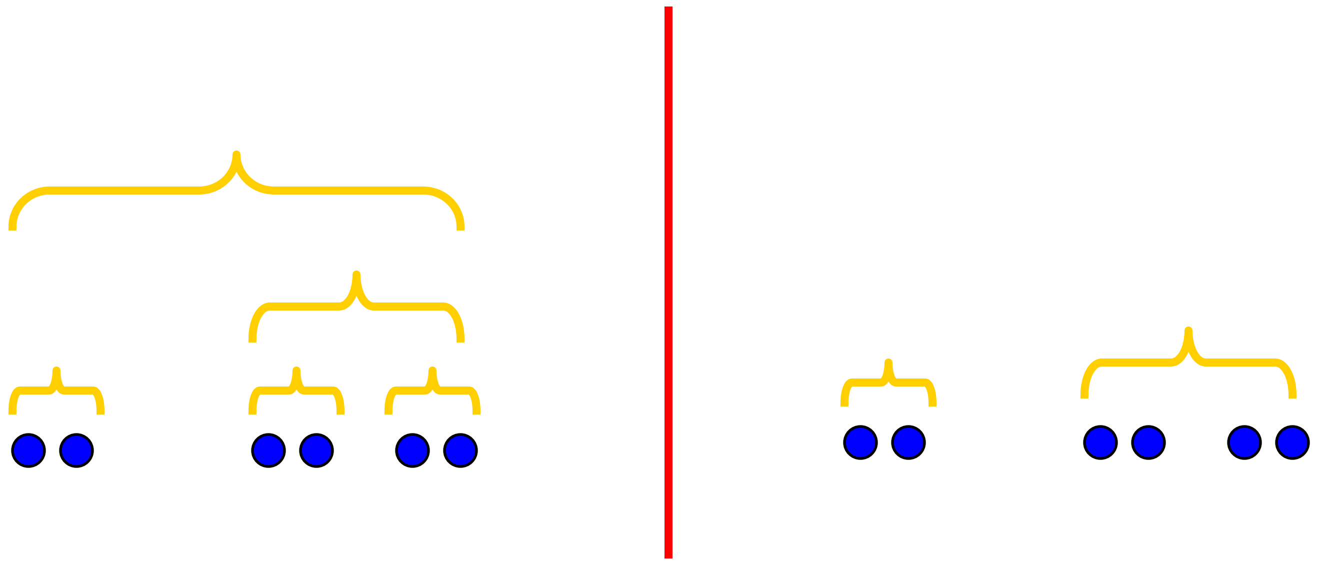
$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + ... + |x_{i_p} - x_{j_p}|^2)}$$

- Properties of a metric *d(i,j)*:
  - $d(i,j) \geq 0$
  - $d(i,i) = 0$
  - $d(i,j) = d(j,i)$
  - $d(i,j) \leq d(i,k) + d(k,j)$

# Basic Concept: Clusters Structure

Hierarchical

Flat



# Basic Concept: Cluster Assignment

- Hard clustering:
  - Each item in only one cluster
- Soft clustering:
  - Each item has a probability of membership in each cluster
- Disjunctive / overlapping clustering:
  - An item can be in more than one cluster

# Major Clustering Approaches

- Hierarchical: Create a hierarchical decomposition of the set of objects using some criterion
- Partitioning: Construct various partitions and then evaluate them by some criterion
- Model-based: Hypothesize a model for each cluster and find best fit of models to data
- Density-based: Guided by connectivity and density functions

# Outline

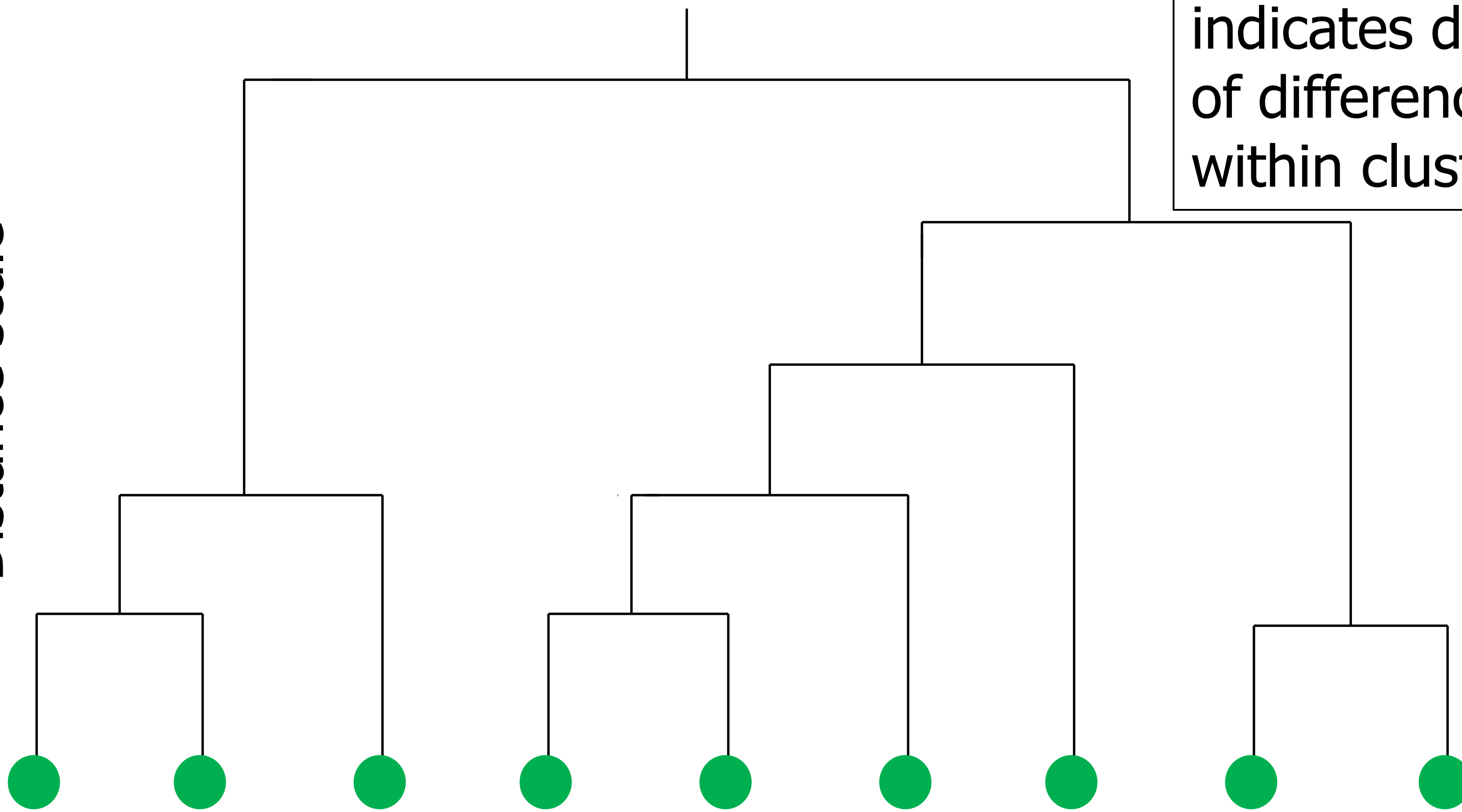
- Homework 4: VC-Dimension problem

- Clustering
  - Unsupervised learning, clustering intro
  - Hierarchical clustering
  - Partitional clustering
  - Model-based clustering
  - Applications

# Hierarchical Clustering

- Can do top-down (divisive) or bottom-up (agglomerative)
- In either case, we maintain a matrix of distance (or similarity) scores for all pairs of
  - Instances
  - Clusters (formed so far)
  - Instances and clusters

# Hierarchical Clustering: Dendogram



# Bottom-Up Hierarchical Clustering

Given: instances $x_1,\ldots,x_n$

For i = 1 to n, $c_i = \{x_i\}$

$C = \{c_1,\ldots,c_n\}$

$j = n$

While $|C| > 1$

  $j = j+1$

  $(c_a,c_b) = \text{argmin dist}(c_u,c_v)$

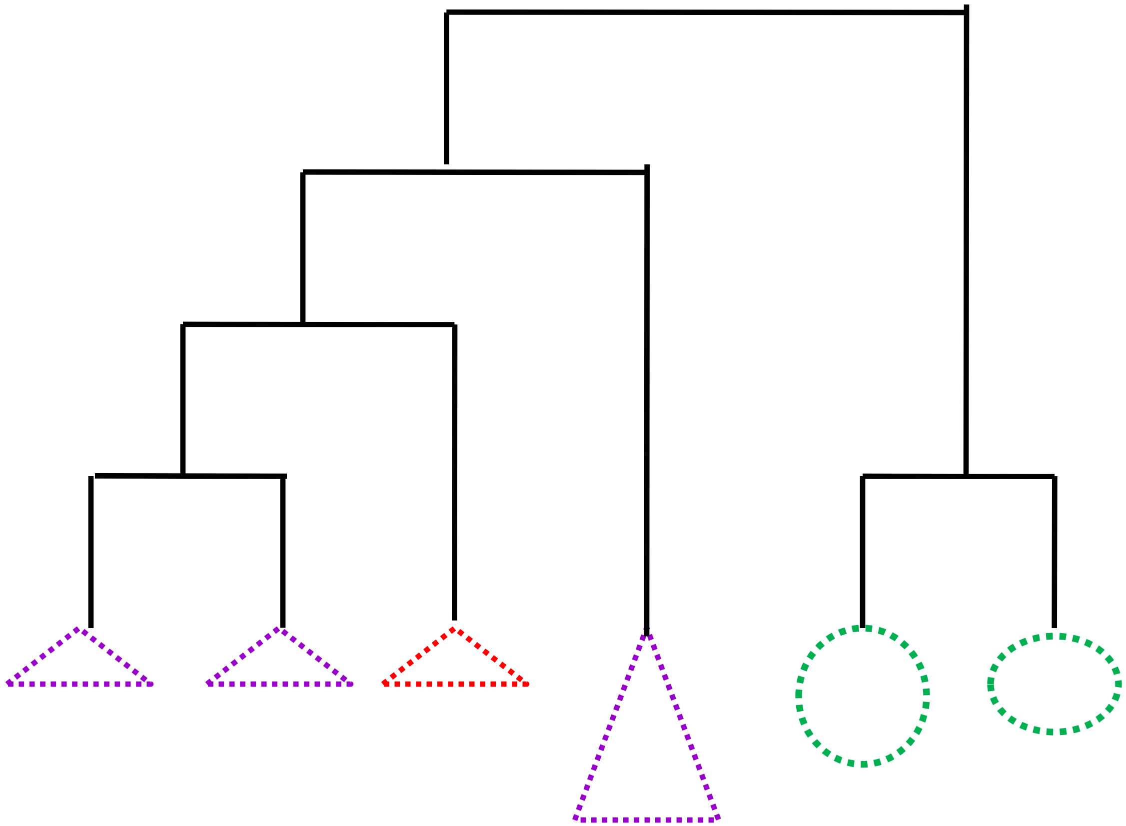
  $c_j = c_a \cup c_v$

  add node to tree joining a and b

  $C = C - \{c_a,c_b\} \cup c_j$

Return tree with root node j

# Bottom-Up Example

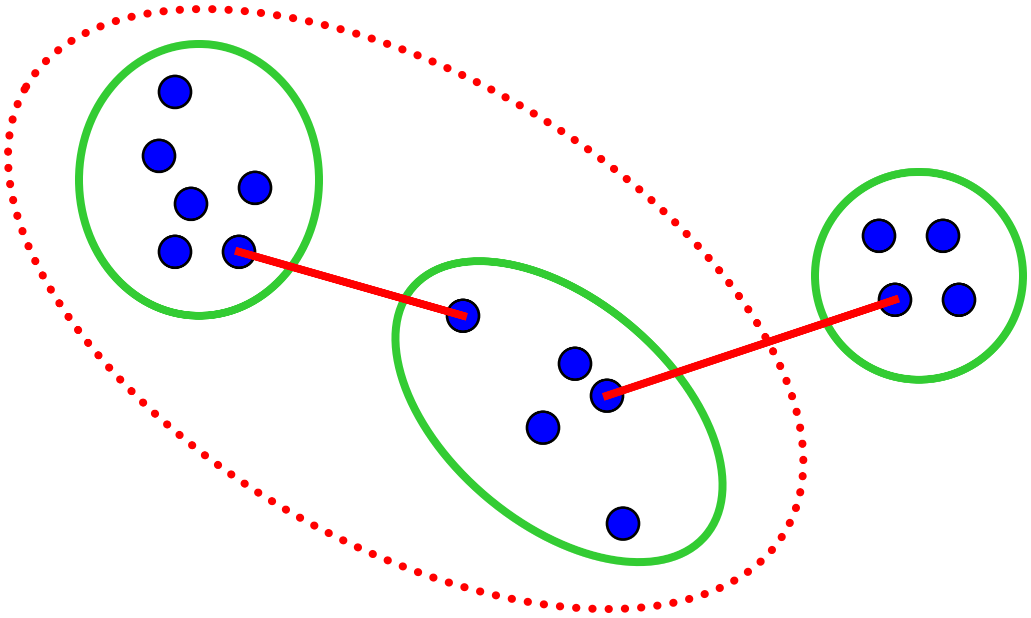


# Distance Between Two Clusters

- The distance between two clusters can be determined in several ways
  - Single link: distance of two most similar instances: $dist(c_u, c_v) = \min\{dist(a, b) \mid a \in c_u, b \in c_v\}$
  - Complete link: distance of two least similar instances: $dist(c_u, c_v) = \max\{dist(a, b) \mid a \in c_u, b \in c_v\}$
  - Average link: average distance between instances: $dist(c_u, c_v) = avg\{dist(a, b) \mid a \in c_u, b \in c_v\}$

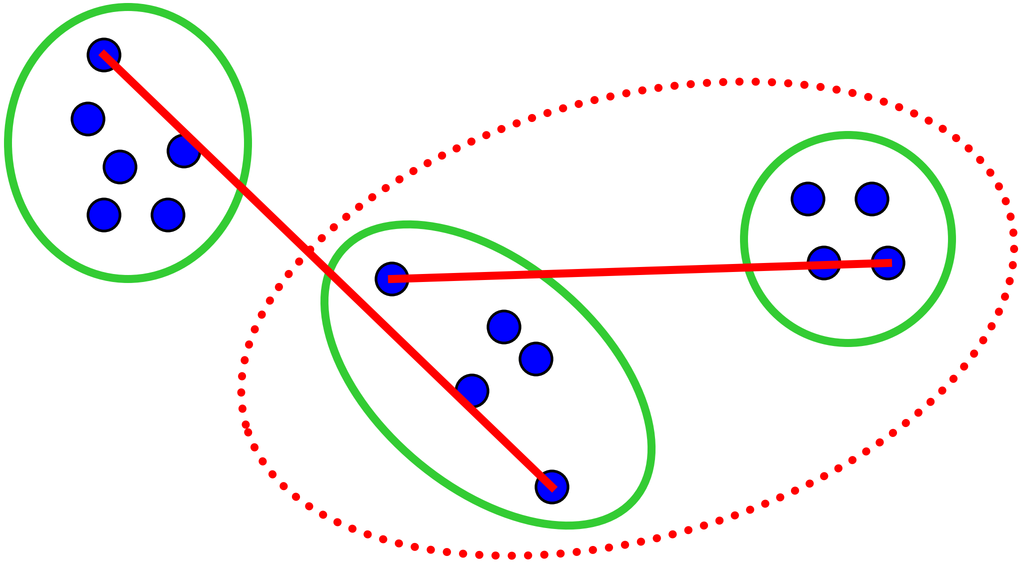
# Single Link

Cluster similarity = similarity of two most similar members



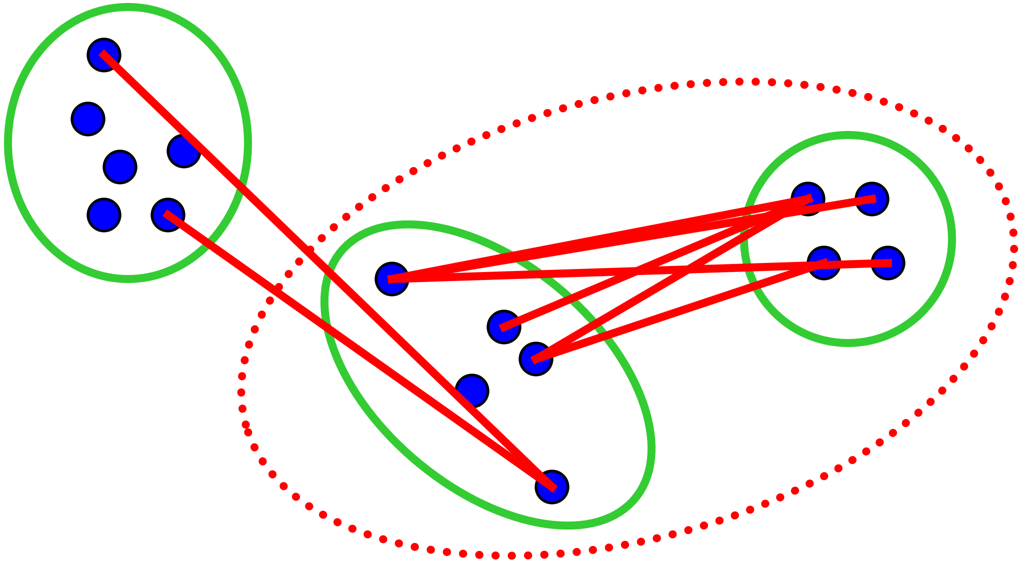
# Complete Link

Cluster similarity = similarity of two least similar members



# Average Link

Cluster similarity = average similarity of all pairs



Note: Picture doesn't show all connections

# Efficient Distance Updates

- If we merged and $c_u$ and $c_v$ into $c_j$, we can determine distance to each other cluster:
  - Single link:
    $\text{dist}(c_j, c_k) = \min\{\text{dist}(c_u, c_k) , \text{dist}(c_v, c_k)\}$
  - Complete link:
    $\text{dist}(c_j, c_k) = \max\{\text{dist}(c_u, c_k) , \text{dist}(c_v, c_k)\}$
  - Average link:
    $$\text{dist}(c_j, c_k) = \frac{|c_u| * \text{dist}(c_u, c_k) + |c_v| * \text{dist}(c_v, c_k)\}}{|c_u| + |c_v|}$$

# Computational Complexity

Naïve implementation has $O(n^3)$ time complexity, where n is the number of instances

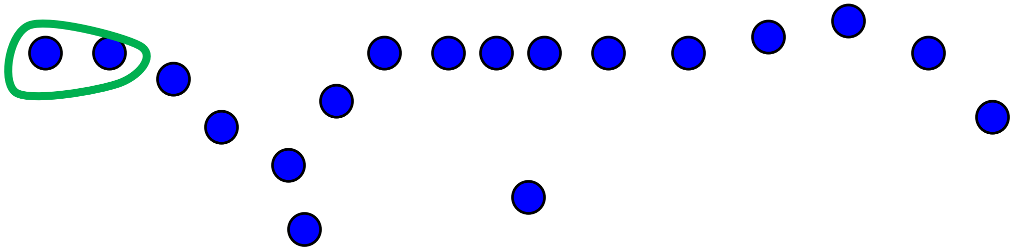
- Compute initial distances: $O(n^2)$
- Merge steps: $O(n)$, each step
  - Update distance matrix: $O(n)$
  - Select next pair of clusters: $O(n^2)$

# Computational Complexity

- Single link: Can update and pick pair in O(n), which results in $O(n^2)$ algorithm
- Complete and average link: Can do these steps in O(n log n), which yields an $O(n^2 \log n)$ algorithm

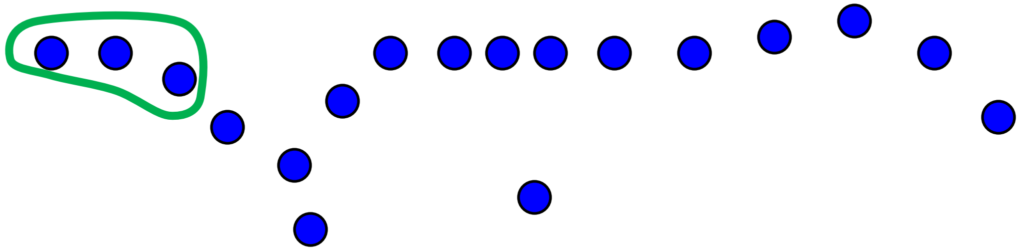
# Single Link

- Chaining:



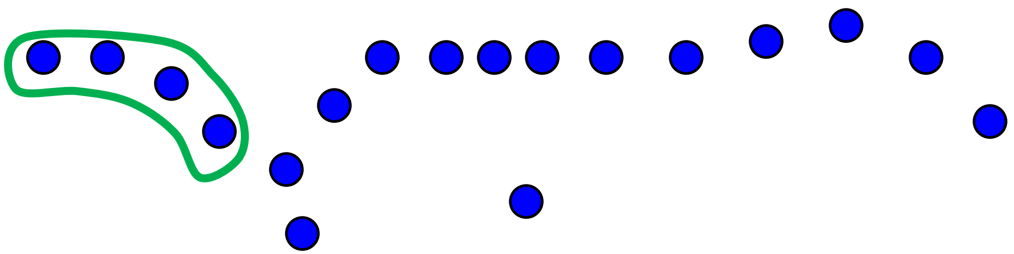
# Single Link

- Chaining:



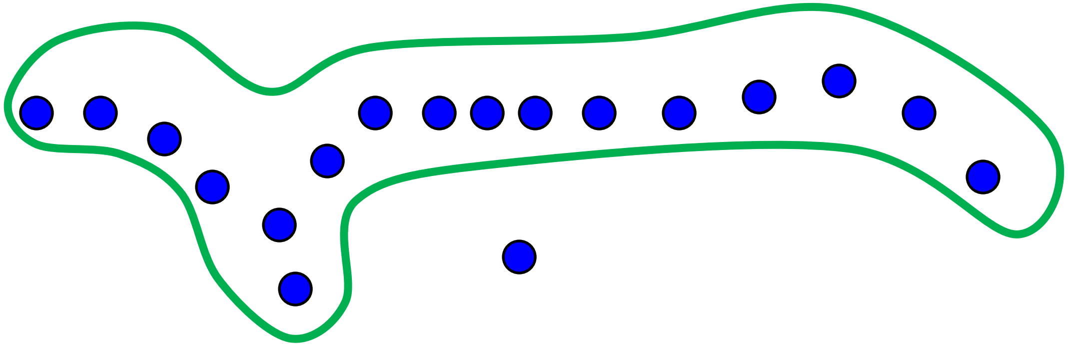
# Single Link

- Chaining:



# Single Link

- Chaining:



- Bottom line:
  - Simple, fast
  - Often low quality

# Complete Link

- Worst case $O(n^3)$
- Fast algorithm: Requires $O(n^2)$ space
- No chaining
- Bottom line:
  - Typically much faster than $O(n^3)$
  - Often good quality

# Divisive or Top-Down Clustering

Initialize: All items one cluster

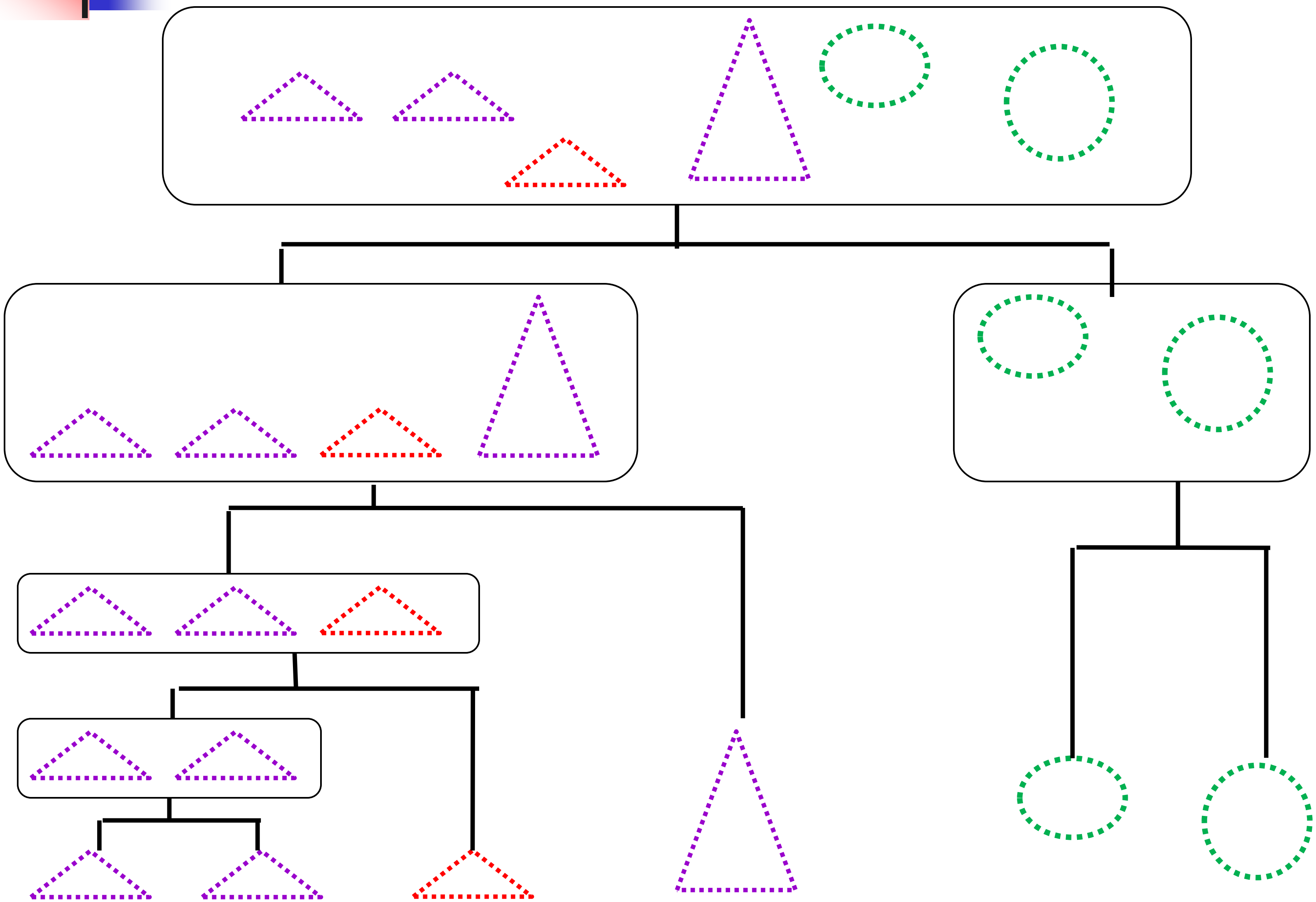
Iterate:

1. select a cluster $c_j$ (least coherent)
2. divide $c_j$ into two clusters

Halt: When have required # of clusters

Note: Step 2 requires another clustering algorithm!

# Top-Down Example



# Other Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - Do not scale well: time complexity of at least $O(n^2)$, where $n$ is the number of total objects
  - Can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - BIRCH: uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CURE: selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction

# BIRCH

- BIRCH: Balanced Iterative Reducing and Clustering using Hierarchies (Zhang, Ramakrishnan & Livny, 1996)
- Incrementally construct a CF (Clustering Feature) tree
  - Parameters: max diameter, max children
  - Phase 1: scan DB to build an initial in-memory CF tree (each node: #points, sum, sum of squares)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan
- *Weaknesses:* handles only numeric data, sensitive to order of data records

# Definitions

- **Centroid:** $\vec{X}0 = \frac{\sum_{i=1}^{N} \vec{X}_i}{N}$

- **Radius:** average distance from member points to cluster centroid $R = (\frac{\sum_{i=1}^{N} (\vec{X}_i - \vec{X}0)^2}{N})^{\frac{1}{2}}$

# Cluster Feature Vector

- Given: $X_1,\ldots,X_n$, data points in a cluster where each with d-dimensions
- We define CF = ($N$, $LS$, $SS$), where
  - $N$: Number of data points
  - $LS$: $\sum_{i=1}^{N} X_i$
  - $SS$: $\sum_{i=1}^{N} X_i^2$
- Note: CFs are additive!
  - E.g., $CF_1 + CF_2 = (N_1+N_2, LS_1+LS_2, SS_1+SS_2)$

# Cluster Feature Example

$$CF = (5, (16,30),(54,190))$$

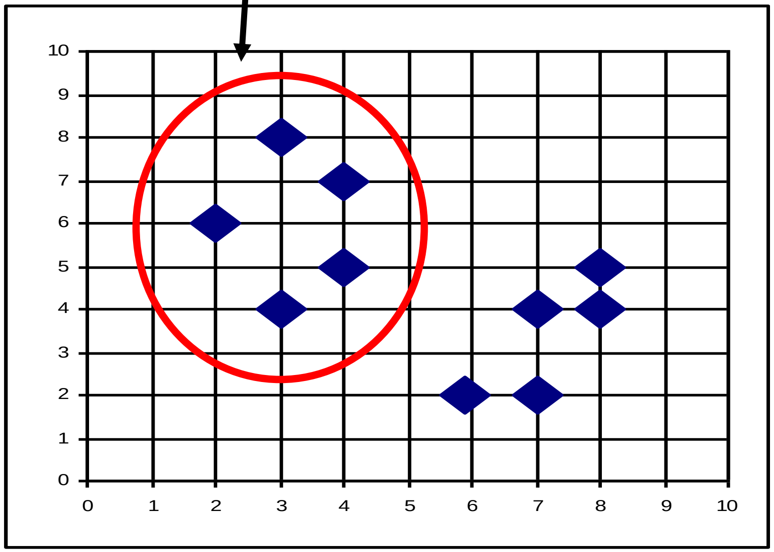
(3,4)

(2,6)

(4,5)

(4,7)

(3,8)




$$LS_x = 3 + 2 + 4 + 4 + 3 = 16$$

$$LS_y = 4 + 6 + 5 + 7 + 8 = 30$$

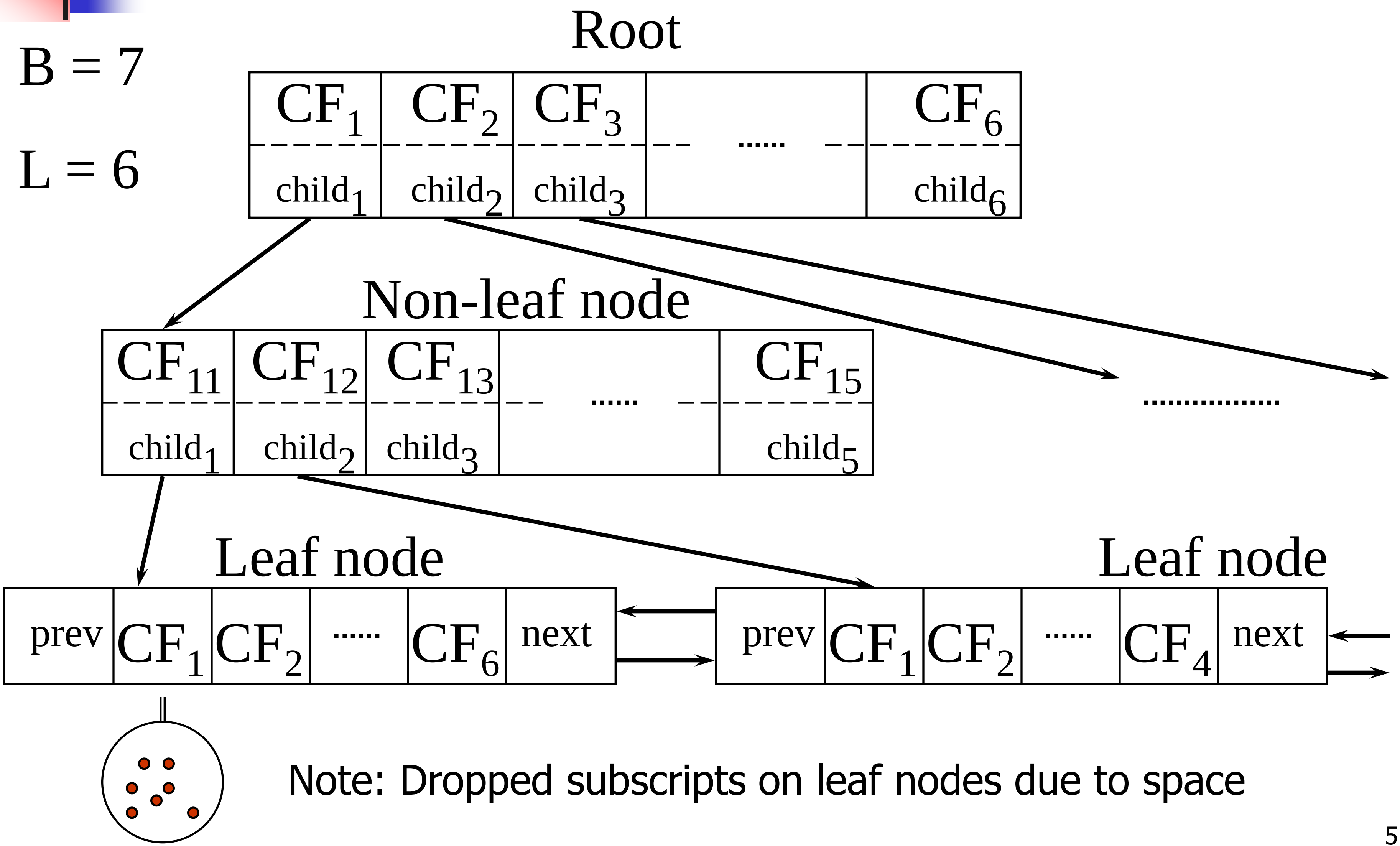
$$SS_x = 3^2 + 2^2 + 4^2 + 4^2 + 3^2 = 54$$

$$SS_y = 4^2 + 6^2 + 5^2 + 7^2 + 8^2 = 190$$

# Cluster Feature Tree

- A CF-tree is a height-balanced tree with two parameters:
  - Branching factor (non leaf nodes B, leaf nodes, L)
  - Threshold T
- Each non leaf node has the form [$CF_i$, $child_i$]
- Each leaf node has CF
  - Set of CFs
  - Two pointers: prev and next
- Diameter of a subcluster under a leaf node can not exceed the threshold T

# CF Tree




Note: Dropped subscripts on leaf nodes due to space

# CF-Tree Construction

- Scan data set and insert the incoming data instances into the CF tree one by one
- Each instance is inserted into the closest subcluster under a leaf node
- If insertion causes subcluster diameter to exceed threshold, then create new subcluster

# CF-Tree Construction

- The new subcluster may cause its parent to exceed branching factor
- If so, split leaf node
  - Identifying the pair of subclusters with largest inter-cluster distance
  - Divide by proximity to these two subclusters
- If this split clause non-leaf node to exceed branching fact, then recursively split
- If the root node is split, then the height of the CF tree is increased by one

# Outline

- Homework 4: VC-Dimension problem

- Clustering
  - Unsupervised learning, clustering intro
  - Hierarchical clustering
  - Partitional clustering
  - Model-based clustering
  - Applications

# Partitioning Algorithms

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means*, *k-medoids* algorithms
  - *k-means* (MacQueen, 1967): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw, 1987): Each cluster is represented by one of the objects in the cluster

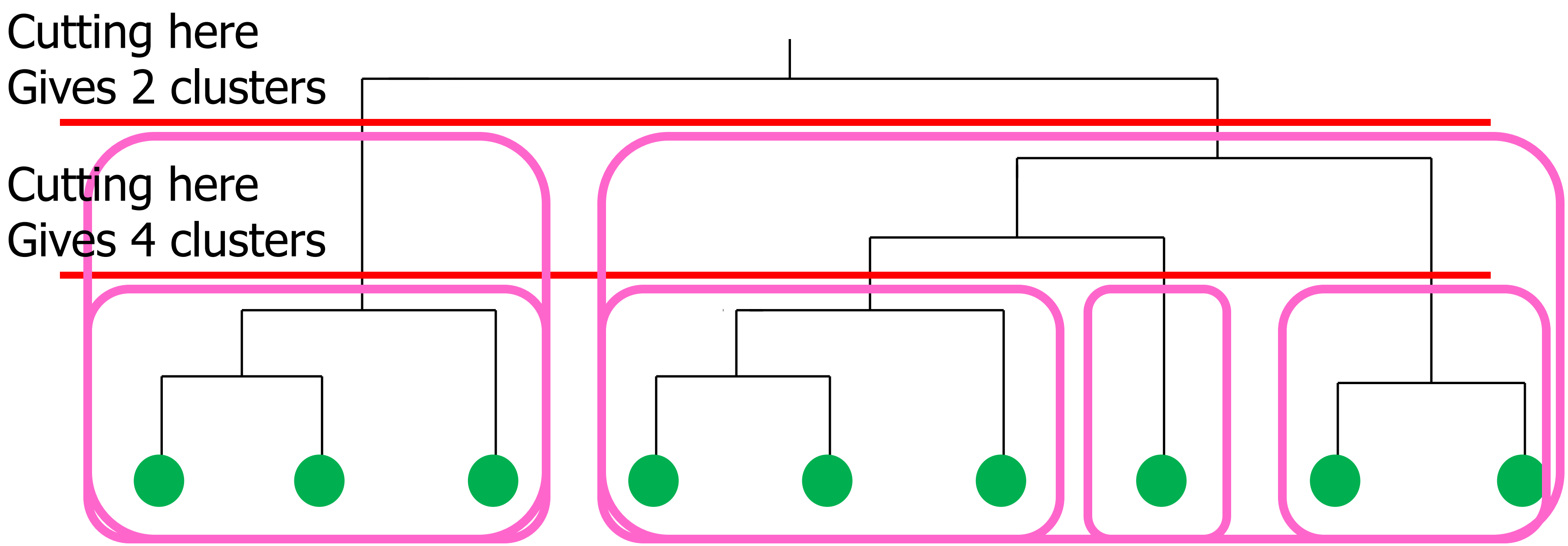
# Partitional Clusterings

- Divide instances into disjoint clusters
  - Flat vs. tree structure

- Key issues:
  - How many clusters should there be?
  - How should clusters be represented?

# Partitional Clustering from a Hierarchical Clustering

- Can generate a partitional clustering from a hierarchical clustering by "cutting" the tree at some level



# K-Means Clustering

- A commonly-used clustering algorithm
  - Easy to implement
  - Quick to run
- Assumes
  - Objects are n-dimensional vectors
  - Distance/similarity measure between these instances
- Goal: Partition the data in K disjoint subsets
- Ideally: Partition reflects the structure of the data

# K-Means Overview

- Inputs:
  - A set of n-dimensional real vectors {x1,…, xm}
  - K, the desired number of clusters
- Output: A mapping of the vectors into k clusters (disjoint subsets), C: {1,…,m} -> {1,…,k}
- The *k* cluster centers are in the same space as instances
- Each cluster is represented by a vector

# K-Means Algorithm

Let $d$ be the distance measure between instances
Pick $k$ random centroids, $s_1,\ldots,sj$

Until clustering converges or other stopping criterion:
For each instance $x_i$:
Assign $x_i$ to the cluster $c_j$ s.t. $d(x_i, s_j)$ is minimal

*Update the centroid of each cluster*
For each cluster $c_j$
$s_j = \mu(c_j)$

# Algorithmic Details

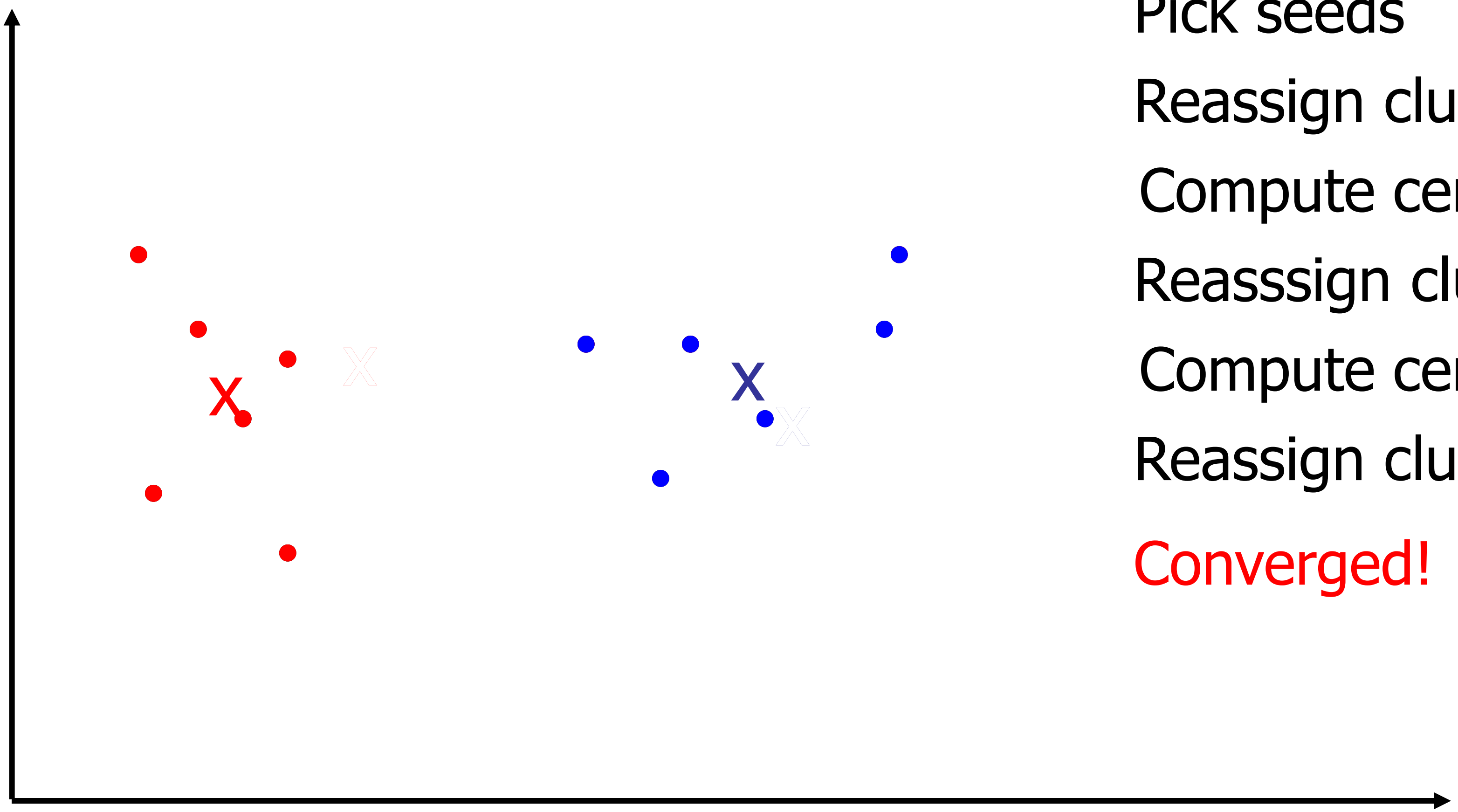
- Initializing the centroids
  - Pick points randomly
  - Pick points from data instances

- $\mu(c_j) = [1/\ |c_j|] * [\ \Sigma_i\ x_i\ ]$
  - $|c_j|$ is number of examples assigned to cluster $c_j$
  - $i \in c_j$, i.e., examples that are assigned to cluster $c_j$
  - Note: This is a vector [calculate the mean along each dimension]

# Seed Choice

- Results vary based on seed selection
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings
- Select good seeds using a heuristic or the results of another method
- Do many runs of k-means, each from a different random start configuration

# K-Means W/K=2




Pick seeds

Reassign clusters

Compute centroids

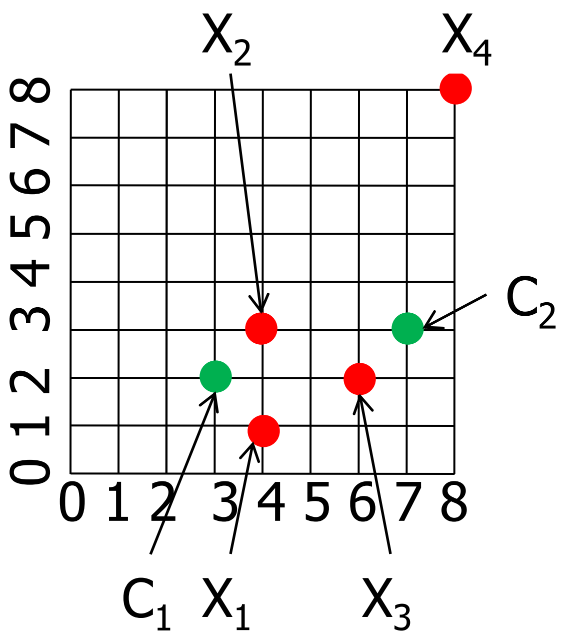
Reasssign clusters

Compute centroids

Reassign clusters

Converged!

# K-Means Example



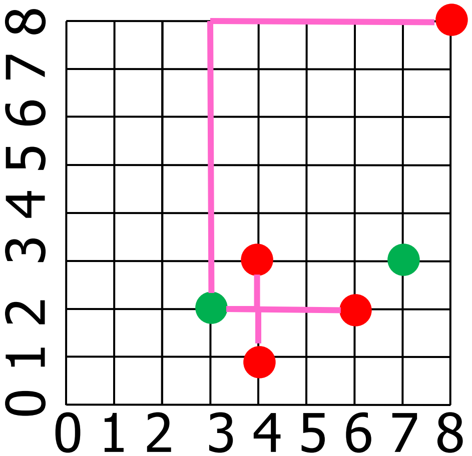

$X_1 = <4,1>$
$X_2 = <4,3>$
$X_3 = <6,2>$
$X_4 = <8,8>$

$C_1 = <3,2>$
$C_2 = <7,3>$

Distance function: Manhattan

# K-Means Example

## Step 1a




$X_1$= <4,1>
$X_2$= <4,3>
$X_3$= <6,2>
$X_4$= <8,8>

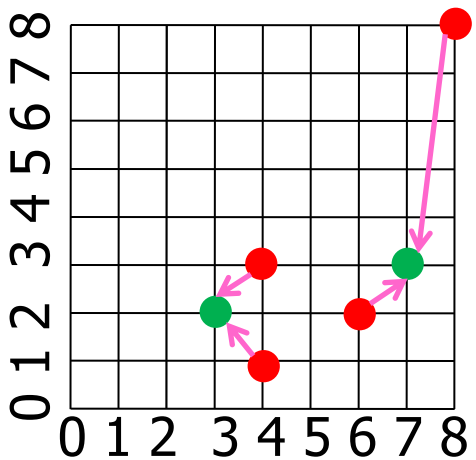
$C_1$= <3,2>
$C_2$= <7,3>

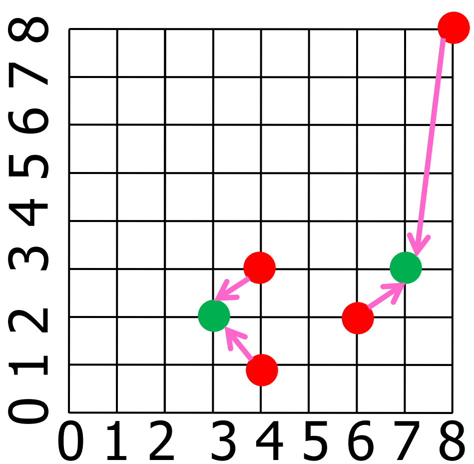
Dist($X_1$,$C_1$) = 2
Dist($X_2$,$C_1$) = 2
Dist($X_3$,$C_1$) = 3
Dist($X_4$,$C_1$) = 11

Dist($X_1$,$C_2$) = 5
Dist($X_2$,$C_2$) = 3
Dist($X_3$,$C_2$) = 2
Dist($X_4$,$C_2$) = 6

# K-Means Example

## Step 1b




$X_1 = <4,1>$ $\quad$ $Dist(X_1,C_1) = 2$

$X_2 = <4,3>$ $\quad$ $Dist(X_2,C_1) = 2$

$X_3 = <6,2>$ $\quad$ $Dist(X_3,C_1) = 3$

$X_4 = <8,8>$ $\quad$ $Dist(X_4,C_1) = 11$

$C_1 = <3,2>$

$C_2 = <7,3>$

$Dist(X_1,C_2) = 5$

$Dist(X_2,C_2) = 3$

$Dist(X_3,C_2) = 2$

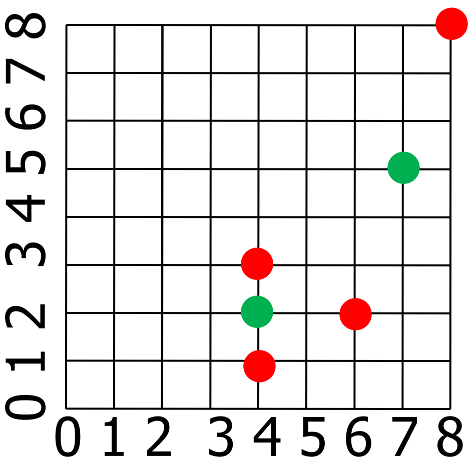
$Dist(X_4,C_2) = 6$

$$C_1 = \left< \frac{4+4}{2}, \frac{1+3}{2} \right> = <4,2>$$

$$C_2 = \left< \frac{6+8}{2}, \frac{2+8}{2} \right> = <7,5>$$

# K-Means Example

## Step 2a




$X_1 = <4,1>$
$X_2 = <4,3>$
$X_3 = <6,2>$
$X_4 = <8,8>$

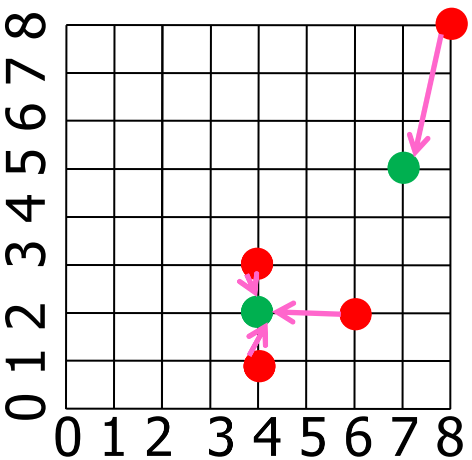
$C_1 = <4,2>$
$C_2 = <7,5>$

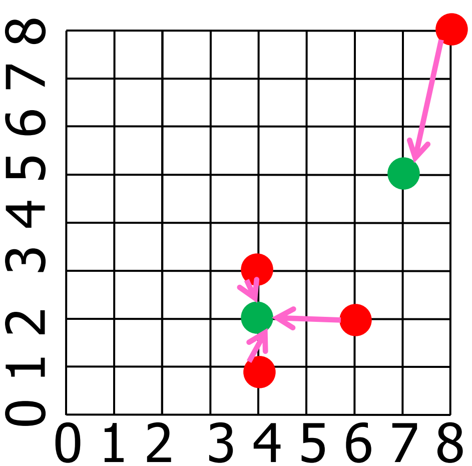
$Dist(X_1,C_1) = 1$
$Dist(X_2,C_1) = 1$
$Dist(X_3,C_1) = 2$
$Dist(X_4,C_1) = 10$

$Dist(X_1,C_2) = 7$
$Dist(X_2,C_2) = 5$
$Dist(X_3,C_2) = 4$
$Dist(X_4,C_2) = 4$

# K-Means Example

## Step 2b




$X_1 = <4,1>$ $\quad$ $Dist(X_1,C_1) = 1$

$X_2 = <4,3>$ $\quad$ $Dist(X_2,C_1) = 1$

$X_3 = <6,2>$ $\quad$ $Dist(X_3,C_1) = 2$

$X_4 = <8,8>$ $\quad$ $Dist(X_4,C_1) = 10$

$C_1 = <4,2>$

$C_2 = <7,5>$

$Dist(X_1,C_2) = 7$

$Dist(X_2,C_2) = 7$

$Dist(X_3,C_2) = 4$

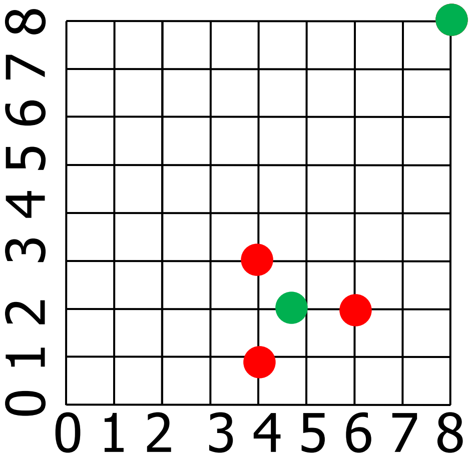
$Dist(X_4,C_2) = 4$

$$C_1 = \left< \frac{4+4+6}{3}, \frac{1+3+2}{3} \right> = <4.67,2>$$

$$C_2 = \left< \frac{8}{1}, \frac{8}{1} \right> = <8,8>$$

# K-Means Example

## Step 3a




$X_1 = \langle 4,1 \rangle$
$X_2 = \langle 4,3 \rangle$
$X_3 = \langle 6,2 \rangle$
$X_4 = \langle 8,8 \rangle$

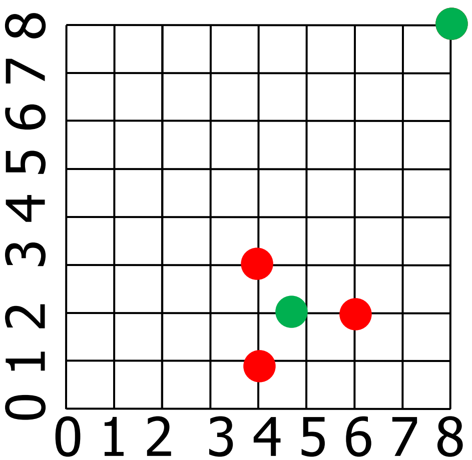
$C_1 = \langle 4.67,2 \rangle$
$C_2 = \langle 8,8 \rangle$

$Dist(X_1,C_1) = 1.67$
$Dist(X_2,C_1) = 1.67$
$Dist(X_3,C_1) = 1.67$
$Dist(X_4,C_1) = 10.33$

$Dist(X_1,C_2) = 11$
$Dist(X_2,C_2) = 9$
$Dist(X_3,C_2) = 8$
$Dist(X_4,C_2) = 0$

# K-Means Example

## Step 3b




$X_1 = <4,1>$ Dist($X_1$,$C_1$) = 1.67
$X_2 = <4,3>$ Dist($X_2$,$C_1$) = 1.67
$X_3 = <6,2>$ Dist($X_3$,$C_1$) = 1.67
$X_4 = <8,8>$ Dist($X_4$,$C_1$) = 10.33

$C_1 = <4.67,2>$ Dist($X_1$,$C_2$) = 11
$C_2 = <8,8>$ Dist($X_2$,$C_2$) = 9
Dist($X_3$,$C_2$) = 8
Dist($X_4$,$C_2$) = 0

Assignment are unchanged -> converged

Note: Not showing centroid recomputatoin

# Time Complexity

- Distance between two instances: O($n$), where $n$ is the dimensionality of the vectors
- Reassigning clusters: O($km$) distance computations, or O($kmn$)
- Computing centroids: Each instance vector gets added once to some centroid: O($nm$)
- Assume these two steps are each done once for $I$ iterations: O($Iknm$)
- Linear in all relevant factors, with fixed number of iterations, more efficient than O($m^2$) HAC

# Comments on the *K-Means* Method

- Strengths
  - *Relatively efficient*: $O(Ikmn)$, where $m$ is # objects, $k$ is # clusters, and $I$ is # iterations. Normally, $k$, $I << m$
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as *simulated annealing* and *genetic algorithms*
- Weaknesses
  - Applicable only when *mean* is defined (what about categorical data?)
  - Need to specify *k,* the *number* of clusters, in advance
  - Trouble with noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

# Outline

- Homework 4: VC-Dimension problem
- Clustering
  - Unsupervised learning, clustering intro
  - Hierarchical clustering
  - Partitional clustering
  - Model-based clustering
  - Applications

# Model-Based Clustering

- Basic idea: Clustering as probability estimation
- One model for each cluster
- *Generative* model:
  - Probability of selecting a cluster
  - Probability of generating an object in cluster
- Find max. likelihood or MAP model
- Missing information: Cluster membership
- Use EM algorithm
- Quality of clustering: Likelihood of test objects

# EM Clustering

- In k-means, instances are assigned to exactly one cluster
- We can do "soft" k-means with an Expectation Maximization algorithm
  - Each cluster represented by a distribution
  - E step: Determine how likely it is that each that each cluster generated each instance
  - M step: Adjust cluster parameters to maximize likelihood

# Mixtures of Gaussians

- Cluster model: Normal distribution (mean, covariance)
- Assume: diagonal covariance, known variance, same for all clusters
- Max. likelihood: mean = avg. of samples
- But what points are samples of a given cluster?
- Estimate prob. that point belongs to cluster
- Mean = weighted avg. of points, weight = prob.
- But to estimate probs. we need model
- "Chicken and egg" problem: use EM algorithm

# EM Algorithm for Mixtures

- **Initialization:** Choose means at random
- **E step:**
  - For all points and means, compute Prob(point|mean)
  - Prob(mean|point) =
    Prob(mean) Prob(point|mean) / Prob(point)
- **M step:**
  - Each mean = Weighted avg. of points
  - Weight = Prob(mean|point)
- Repeat until convergence

# Representing Clusters

- Represent clusters with a Gaussian

$$N_j(x_i) = \frac{1}{(2\pi\sigma^2)^{0.5}} e^{\frac{-1}{2}\left[\frac{(x_i - \mu_j)}{\sigma}\right]^2}$$

- Where
  - $\mu_j$ is the mean
  - $\sigma^2$ is the variance
  - $N_j(x_i)$ = probability($x_i | \mu_j$)

# EM Clustering: Hidden Variables

- On each iteration of *k-means clustering, we had to assign* each instance to a cluster
- In the EM approach, we'll use hidden variables to represent this idea
- For each instance $x_i$ we have a set of hidden variables $z_{i1},\dots,z_{ik}$
- We can think of $z_{ij}$ as being 1 if is a member of cluster *j and 0 otherwise*

# E-Step

- Recall that $z_{ij}$ is a hidden variable which is 1 if $N_j$ generated $x_i$ and 0 otherwise
- In the E-step, we compute $h_{ij}$, the expected value of this hidden variable

$$h_{ij} = \frac{P_j * N_j(x_i)}{\Sigma_l \, P_l * N_l(x_i)}$$

# M-Step

- Given the expected values $h_{ij}$, we re-estimate the means of the Gaussians and the cluster probabilities

$$\mu_j = \frac{\Sigma_i\, x_i * h_{ij}}{\Sigma_i\, h_{ij}}$$

$$P_j = \frac{\Sigma_i\, h_{ij}}{n}$$

Note: i goes over examples

# EM Clustering Example

- Consider a one-dimensional clustering problem:
  - x1 = -4
  - x2 = -3
  - x3 = -1
  - x4 = 3
  - x5 = 5
- Settings
  - $\mu 1 = 0$, $\mu_2 = 2$, both have $\sigma = 2$
  - Density function is: $f(x, \mu) = \frac{1}{(8\pi)^{0.5}} e^{\frac{-1}{2}\left[\frac{(x-\mu)}{2}\right]^2}$
  - Initially, we set P1 = P2 = 0.5

# EM Clustering Example

$$F(-4, \mu_1) = \frac{1}{(8\pi)^{0.5}} e^{\frac{-1}{2}\left[\frac{(4-0)}{2}\right]^2}$$

- $f(-4, \mu_1) = 0.0269$
- $f(-3, \mu_1) = 0.0646$
- $f(-1, \mu_1) = 0.176$
- $f(3, \mu_1) = 0.0646$
- $f(5, \mu_1) = 0.00874$

- $f(-4, \mu_2) = 0.0022$
- $f(-3, \mu_2) = 0.00874$
- $f(-1, \mu_2) = 0.0646$
- $f(3, \mu_2) = 0.176$
- $f(5, \mu_2) = 0.0646$

# EM Clustering Example: E Step

$$h_{11} = \frac{P_1 * f(x_1, \mu_1)}{P_1 * f(x_1, \mu_1) + P_2 * f(x_1, \mu_2)} = \frac{0.5 * .0269}{0.5*0.0269+0.5*0.0022} = 0.924$$

- $h_{11} = 0.924$
- $h_{21} = 0.881$
- $h_{31} = 0.732$
- $h_{41} = 0.268$
- $h_{51} = 0.119$

- $h_{12} = 0.076$
- $h_{22} = 0.119$
- $h_{32} = 0.268$
- $h_{42} = 0.732$
- $h_{52} = 0.881$

# EM Clustering Example: M Step

$$\mu_1 = \frac{\Sigma_i x_i * h_{i1}}{\Sigma_i h_{i1}} \qquad \mu_2 = \frac{\Sigma_i x_i * h_{i2}}{\Sigma_i h_{i2}}$$

$$\mu_1 = \frac{-4*0.924 + -3*0.881 + -1*0.732 + 3*0.268 + 5*0.119}{0.924 + 0.881 + 0.732 + 0.268 + 0.119} = -1.94$$

$$\mu_2 = \frac{-4*0.076 + -3*0.119 + -1*0.268 + 3*0.732 + 5*0.881}{0.076 + 0.119 + 0.268 + 0.732 + 0.881} = 3.39$$

$$P_1 = \frac{\Sigma_i h_{i1}}{n} = \frac{0.924 + 0.881 + 0.732 + 0.268 + 0.119}{5} = 0.58$$

$$P_2 = \frac{\Sigma_i h_{i2}}{n} = \frac{0.076 + 0.119 + 0.268 + 0.732 + 0.881}{5} = 0.42$$

# EM Clustering

- Will converge to a local maximum
- Sensitive to initial means of clusters
- Have to choose the number of clusters in advance
- k-means is a special case of EM clustering

# Evaluating Cluster Results

- Given random data without any “structure”, clustering algorithms will still return clusters
- The gold standard: do clusters correspond to natural categories?
- Do clusters correspond to categories we care about? (there are lots of ways to partition the world)

# Approaches to Cluster Evaluation

- External validation
  - e.g. do genes clustered together have some common function?
- Internal validation
  - How well does clustering optimize intra-cluster similarity and inter-cluster dissimilarity?
- relative validation
  - How does it compare to other clusterings?
  - e.g. with a probabilistic method (such as EM) we can ask: how probable does held-aside data look as we vary the number of clusters.

# Outline

- Homework 4: VC-Dimension problem

- Clustering
  - Unsupervised learning, clustering intro
  - Hierarchical clustering
  - Partitional clustering
  - Model-based clustering
  - Applications

# Low Quality of Web Searches

- System perspective:
  - small coverage of Web (<16%)
  - dead links and out of date pages
  - limited resources
- IR perspective (relevancy of doc ~ similarity to query):
  - very short queries
  - huge database
  - novice users

# Document Clustering

- User receives many (200 - 5000) documents from Web search engine

- Group documents in clusters
  - by topic
- Present clusters as interface

Clusty

web news images wikipedia blogs jobs more »

ncaa basketball tournament [Search] advanced preferences

clusters sources sites

remix

- **All Results** (216)
- **Brackets** (40)
- **Tickets** (38)
- **March Madness** (30)
- **NCAA Men's Basketball Tournament** (18)
- **Women's** (17)
- **Pool** (9)
- **Photos** (11)
- **Hoops** (5)
- **Memphis** (7)
- **Programs** (5)

more | all clusters

find in clusters: [Find]

Font size: A A A A

Top **212** results of at least **2,237,000** retrieved for the query **ncaa basketball tournament** (details)

Sponsored Results

College Hoops Contest - Know college **basketball** ? Prove it. Go for the $100,000 grand prize! - www.wagerline.com

Sports Contest Promotions - Run a sports contest promotion for your business or website. - www.poolhost.com

Search Results

1. **NCAA** Men's Division I **Basketball** Championship - Wikipedia, the free ...
The **NCAA** Men's Division I **Basketball** Championship is a single elimination **tournament** held each spring featuring 65 [1] college **basketball** teams in the United States. This **tournament**, organized by the National Collegiate Athletic Association (**NCAA**), was first developed by the National Association of **Basketball** Coaches in 1939. [2]**Tournament** format · Format history · March Madness and ...
en.wikipedia.org/wiki/NCAA_Men's_Division_I_Basketball_Championship - [cache] - Live, Ask, Gigablast

2. Welcome To Your Official **NCAA** Web Sites
Enter **NCAA**.com For complete March Madness coverage, brackets and other championship **tournament** information for all **NCAA** sports. Enter **NCAA**.org For information about the **NCAA**,
www.ncaa.org - [cache] - Live, Gigablast

3. 2009 **NCAA Basketball Tournament** | CollegeHoops.net
2009 **NCAA Tournament** preview, schedule, bracket, and bracketology.
www.collegehoopsnet.com/ncaatournament - [cache] - Live, Ask

4. **NCAA Tournament** Tickets, 2009 **NCAA Basketball Tournament** Info, Final
March Madness is here and GoTickets.com has your 2009 **NCAA**® Men's **Basketball Tournament** tickets and Final Four® tickets.
www.gotickets.com/sports/college_basketball/ncaa_tournament.php - [cache] - Ask, Gigablast

5. **NCAA**.com - The Official Web Site of the **NCAA**
Selection Sunday Challenge. Think you deserve a seat on the **NCAA** Men's **Basketball** Selection Committee? See if you can pick the correct field of 65.
www.ncaa.com - [cache] - Live, Gigablast

6. **NCAA Tournament** Tickets, 2009 **NCAA Basketball Tournament** Ticket ...
**NCAA Tournament** Tickets from TickCo Premium Seating; rapid delivery on **NCAA Tournament**/March Madness tickets order and save today!
www.tickco.com/sports_basketball_ncaa_tournament_tickets.htm - [cache] - Live, Ask

7. Working Class Software
**NCAA basketball tournament** program.
www.wcsoftware.com - [cache] - Open Directory, Ask, Gigablast

8. **NCAA Basketball Tournament** Most Outstanding Player - Wikipedia, the ...
At the conclusion of the **NCAA** men's and women's Division I **basketball** championships (the "Final Four" **tournaments**), the Associated Press selects a Most Outstanding Player. The MOP need not be, but almost always is a member of the Championship team. The last man to win the award despite not being on the Championship team was Hakeem Olajuwon in 1983; the last woman to do so was Dawn Staley in 1991.
en.wikipedia.org/wiki/NCAA_Basketball_Tournament_Most_Outstanding_Player - [cache] - Live, Ask

web news images wikipedia blogs jobs more »

ncaa basketball tournament Search advanced preferences

clusters sources sites

**All Results** (216) remix

- **Brackets** (40)
- **Tickets** (38)
- **March Madness** (30)
- **NCAA Men's Basketball Tournament** (18)
- **Women's** (17)
- **Pool** (9)
- **Photos** (11)
- **Hoops** (5)
- **Memphis** (7)
- **Programs** (5)

more | all clusters

find in clusters: Find

Font size: A A A A

Cluster **Brackets** contains **40** documents.

Search Results

1. **NCAA** March Madness on Demand - **NCAA**.com
   Official website for **NCAA** sports news. News, articles, scores, **brackets**, venues, history, photos, team capsules. ... The **NCAA** Final Four® tip off Saturday at 6:07 PM ET! Until then,
   www.ncaasports.com - [cache] - Ask
2. 2009 College **Basketball Tournament Brackets** - CBSSports.com
   Play CBSSports.com March Madness **NCAA basketball brackets** ... **NCAA** College **Basketball** Sports News
   www.sportsline.com/collegebasketball/mayhem/brackets/viewable_men - [cache] - Ask
3. **NCAA** College **Basketball** - CBSSports.com News, Fantasy, Video
   What's Hot in **NCAA** College **Basketball** ... **NCAA tournament brackets**: Live, updating **bracket** | Printable | **Bracket** games | Experts; **NCAA tournament** history: Past champions and **brackets** | Team-by
   www.sportsline.com/collegebasketball - [cache] - Ask
4. March Madness Beyond the Idiot Box
   CBS is expecting a huge payday after jacking up the **basketball tournament**'s presence on the Web ... For CBS (CBS), which has the rights to **NCAA**'s championship, it's a fast break to the Net.
   www.businessweek.com/magazine/content/08_13/b4077070416250.htm - [cache] - Ask
5. CBSSports.com to Share March Madness on Demand - 2008-03-11 06:37:00 |
   Developer Platform Will Allow Other Web Sites to Link to **NCAA Tournament** Coverage ... CBSSports.com, with the OK of the **NCAA**, said Tuesday that it dropped the registration requirements for its
   www.broadcastingcable.com/article/CA6540037.html - [cache] - Ask
6. FOX Sports on MSN - COLLEGE **BASKETBALL** - 2009 **NCAA tournament** -
   Does your **bracket** have the winning touch? Check how your picks are doing as the **NCAA tournament** rolls on. ... COLLEGE **BASKETBALL** HEADLINES
   msn.foxsports.com/cbk/story/7912002/Bracket-central:-Print-your-brackets - [cache] - Ask
7. Win $100,000 By Entering Your **NCAA Basketball Tournament** Picks In
   PR: Sign up now, fill out your **bracket** and win! ... sports game **basketball** march march madness **ncaa tournament** pool **bracket** free cash jacked
   www.prweb.com/releases/2008/02/prweb722733.htm - [cache] - Ask
8. Celebrity **Bracket** Challenge: Mike Conley - 2009 March Madness | **NCAA**
   Get March Madness **NCAA basketball tournament** news, **brackets**, scores, facts, rankings, schedules, picks & more. Comment on the news, see photos and join the forum discussions at cleveland.com...
   www.cleveland.com/.../index.ssf/2008/03/celebrity_bracket_challenge_mi.html - [cache] - Ask
9. Why the **NCAA basketball tournament** seedings make sense
   When the **NCAA** Men's **Basketball Tournament** committee releases its **brackets** on Selection Sunday, there's always a fair amount of second guessing -- and this past Sunday was no exception.
   www.post-gazette.com/pg/06076/672233-291.stm - [cache] - Ask

# Q: Need Way to Compare Queries and Documents

- Vector space model:
  - How to determine important words in a document?
  - How to determine the degree of importance of a term within a document and within the entire collection?
  - How to determine the degree of similarity between a document and the query?
  - In the case of the web, what is a collection and what are the effects of links, formatting information, etc.?

# Vector-Space Model

- Assume $t$ distinct terms remain after preprocessing: vocabulary
- These "orthogonal" terms form a vector space

  Dimension = $t$ = |vocabulary|
- Each term, $i$, in a document or query, $j$, is given a real-valued weight, $w_{ij}$.
- Both documents and queries are expressed as $t$-dimensional vectors:

$$d_j = (w_{1j}, w_{2j}, \ldots, w_{tj})$$

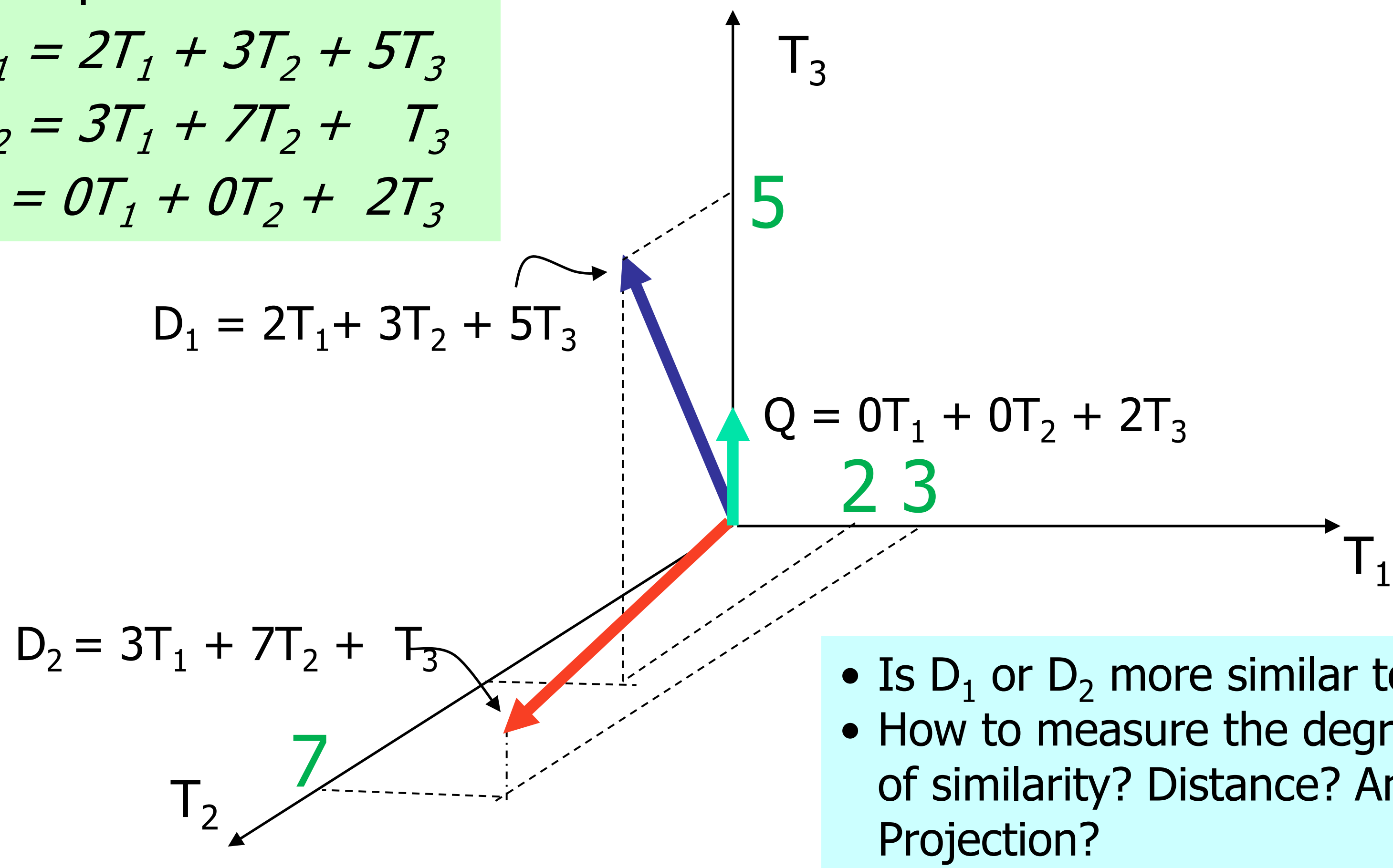
# Graphical Representation

Example:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + \ \ T_3$

$Q = 0T_1 + 0T_2 + \ 2T_3$




- Is $D_1$ or $D_2$ more similar to Q?
- How to measure the degree of similarity? Distance? Angle? Projection?

# Document Collection

- Vector space model represents a collection of *n* documents by a term-document matrix
- Each entry: "weight" of a term in the document

$$
\begin{array}{c|cccc}
 & T_1 & T_2 & \cdots & T_t \\
\hline
D_1 & w_{11} & w_{21} & \cdots & w_{t1} \\
D_2 & w_{12} & w_{22} & \cdots & w_{t2} \\
\vdots & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & & \vdots \\
D_n & w_{1n} & w_{2n} & \cdots & w_{tn}
\end{array}
$$

# Term Weights: Term Frequency

- More frequent terms in a document are more important, i.e. more indicative of the topic

  $f_{ij}$ = frequency of term $i$ in document $j$

- May want to normalize *term frequency* (*tf*) by dividing by the frequency of the most common term in the document:

  $$tf_{ij} = f_{ij} / max_i\{f_{ij}\}$$

# Term Weights: Inverse Document Frequency

- Terms that appear in many *different* documents are *less* indicative of overall topic

  $df_i$ = document frequency of term $i$

  = number of documents containing term $i$

  $idf_i$ = inverse document frequency of term $i$,

  $= \log_2 (N / df_i)$ ($N$: number of documents)
- An indication of a term's *discrimination* power
- Log used to dampen the effect relative to *tf*

# TF-IDF Weighting

- A typical combined term importance indicator is *tf-idf weighting*:

$$w_{ij} = tf_{ij}\, idf_i = tf_{ij} \log_2 (N / df_i)$$

- A term occurring frequently in the document but rarely in the rest of the collection is given high weight
- Many other ways of determining term weights have been proposed
- Experimentally, *tf-idf* works well

# TF-IDF Example

Given a document containing terms with given frequencies:

A(3), B(2), C(1)

Assume collection contains 10,000 documents and

document frequencies of these terms are:

A(50), B(1300), C(250)

Then:

A: tf = 3/3; idf = $\log_2(10000/50)$ = 7.6; tf-idf = 7.6

B: tf = 2/3; idf = $\log_2(10000/1300)$ = 2.9; tf-idf = 2.0

C: tf = 1/3; idf = $\log_2(10000/250)$ = 5.3; tf-idf = 1.8

# Query Vector

- Query vector is typically treated as a document and also tf-idf weighted
- Alternative is for the user to supply weights for the given query terms

# Similarity Measures

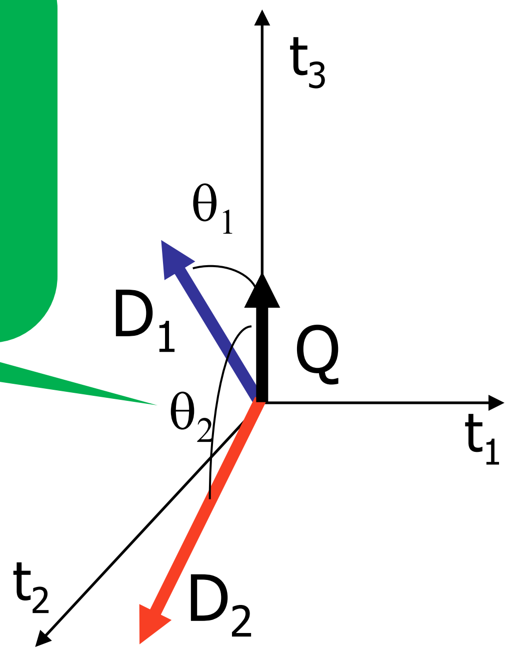
- Inner product: $sim(d_j,q) = \Sigma\, w_{ij} * w_{iq}$
  - $w_{ij}$ = weight of term i in doc j
  - $w_{iq}$ is weight of term i in query

- Cosine similarity: $sim(d_j,q) = \dfrac{\Sigma_i\, w_{ij} * w_{iq}}{\Sigma_i\, (w_{ij})^2\, \Sigma_i\, (w_{iq})^2}$

  - Measures the cosine of the angle between two vectors
  - Inner product normalized by the vector lengths

# Cosine Similarity Visually

Take cosine of this angle as similarity between query and document



# Comparison

- $D_1 = 2T_1 + 3T_2 + 5T_3$
- $D_2 = 3T_1 + 7T_2 + 1T_3$
- $Q = 0T_1 + 0T_2 + 2T_3$
- Weighted inner product
  - $sim(D_1 , Q) = 2*0 + 3*0 + 5*2 = 10$
  - $sim(D_2 , Q) = 3*0 + 7*0 + 1*2 = 2$
- Cosine
  - $sim(D_1 , Q) = 10 / \sqrt{(4+9+25)(0+0+4)} = 0.81$
  - $sim(D_2 , Q) = 2 / \sqrt{(9+49+1)(0+0+4)} = 0.13$

$D_1$ is 6 times better than $D_2$ using cosine similarity but only 5 times better using inner product.

# Comments On Vector Space Model

- Simple, mathematically based approach
- Considers both local (*tf*) and global (*idf*) word occurrence frequencies
- Provides partial matching and ranked results.
- Tends to work quite well in practice despite obvious weaknesses
- Allows efficient implementation for large document collections

# Weakness with Vector Space Model

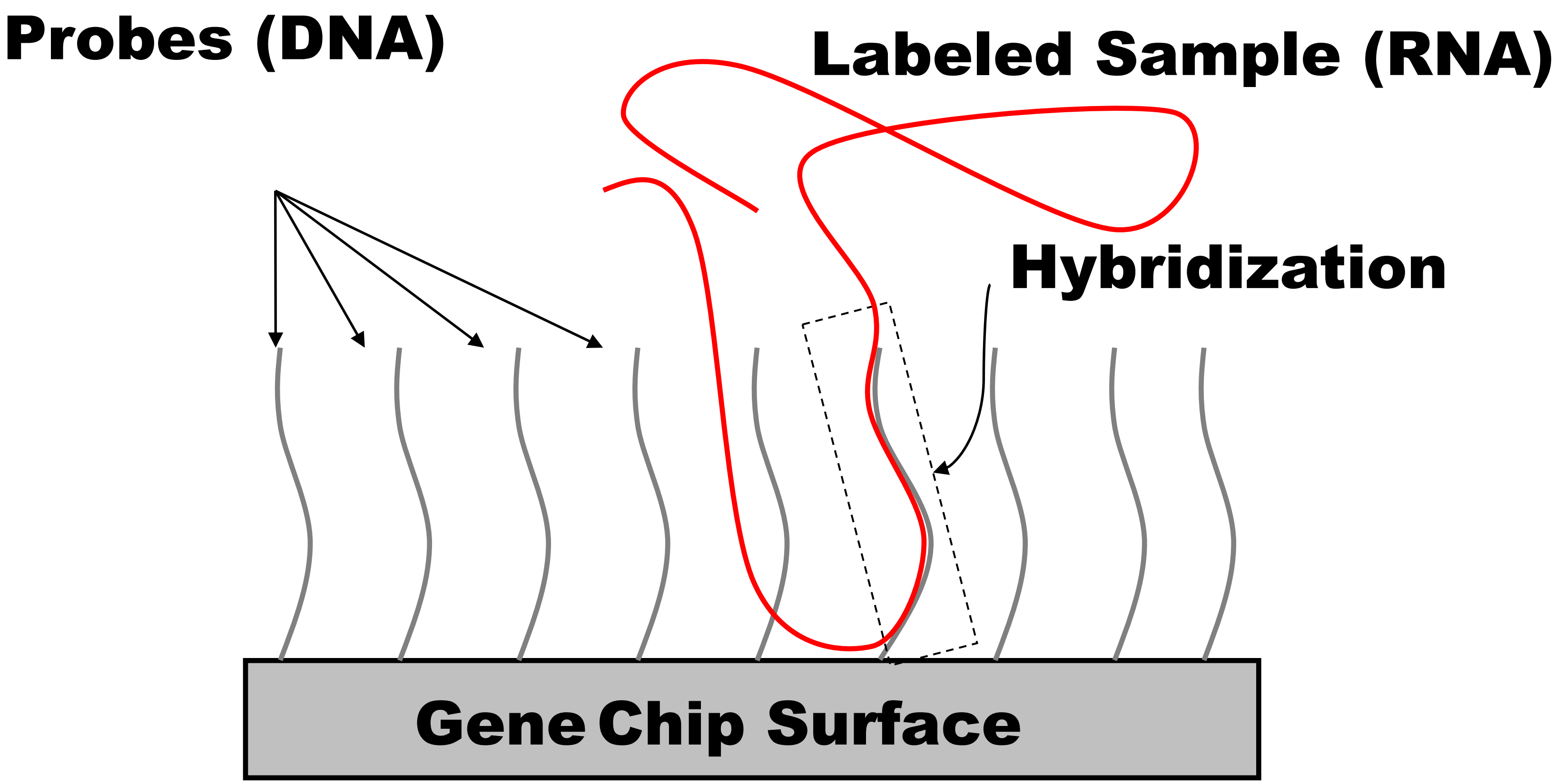
- Missing semantic information (e.g. word sense)
- Missing syntactic information (e.g. phrase structure, word order, proximity information)
- Assumption of term independence (e.g. ignores synonomy)

# Analyzing Microarray Data

- Microarrays allow us to measure gene expression

- Central Dogma:
  - Genes encode proteins
  - DNA transcribed into messenger RNA
  - mRNA translated into proteins
  - Triplet code (codons)

# How Microarrays Work



# Two Views of Microarray Data

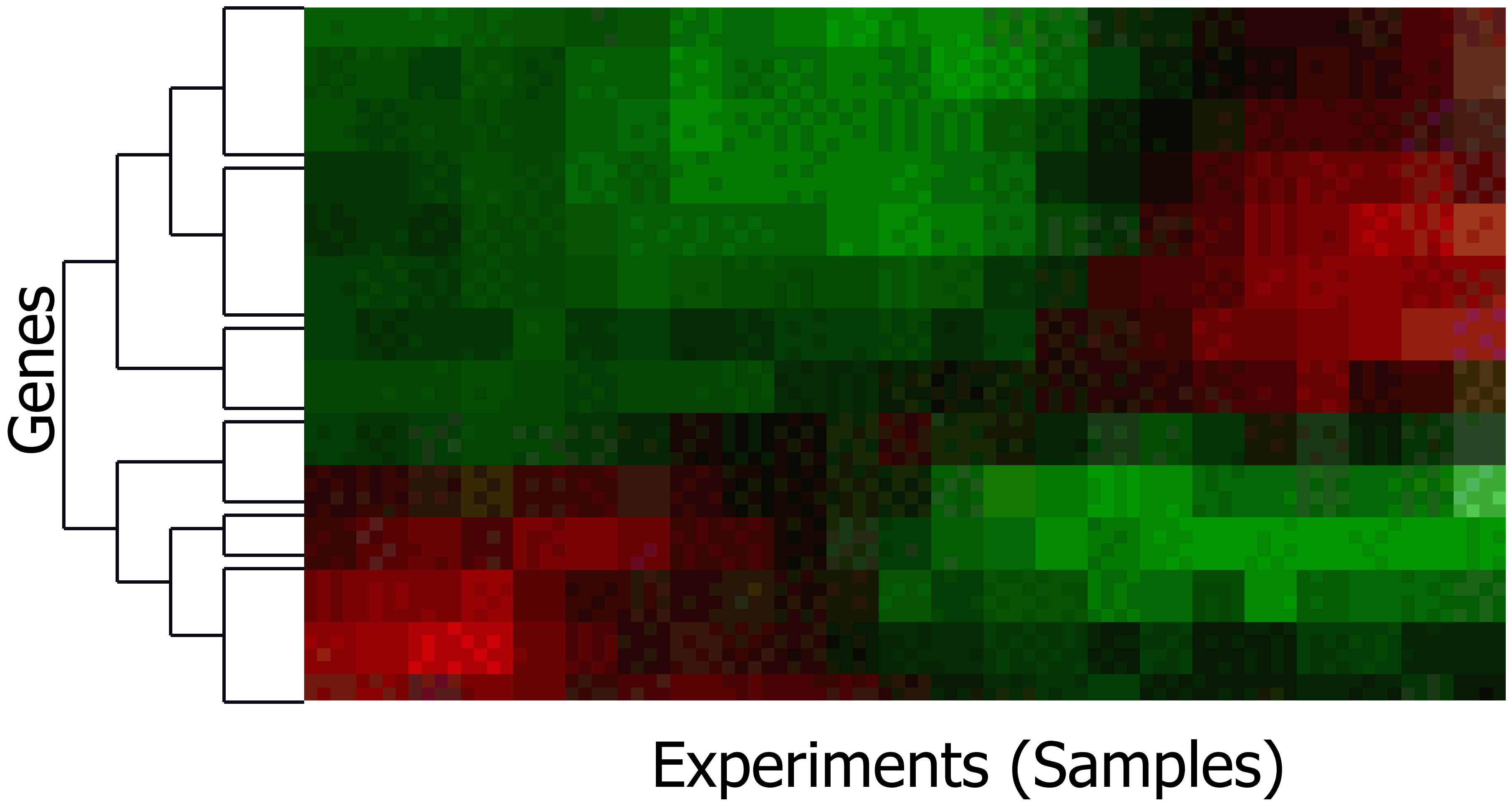
- Data points are genes
  - Represented by expression levels across <u>different samples</u> (ie, features=samples)
  - **Goal**: categorize new genes
- Data points are samples (eg, patients)
  - Represented by expression levels of <u>different genes</u> (ie, features=genes)
  - **Goal**: categorize new samples

# Unsupervised Learning Task

- **Given**: a set of microarray experiments under different conditions
- **Do**: cluster the genes, where a gene described by its expression levels in different experiments

# Example

(Green = up-regulated, Red = down-regulated)



# Unsupervised Learning Task 2

- **Given**: a set of microarray experiments (samples) corresponding to different conditions or patients

- **Do**: <u>cluster</u> the experiments

# Examples

- Cluster samples from mice subjected to a variety of toxic compounds
  (Thomas *et al*., 2001)
- Cluster samples from cancer patients, potentially to discover different subtypes of a cancer
- Cluster samples taken at different time points

# Summary

- Unsupervised learning technique: Gain insight into the data
- Clustering approaches
  - Hierarchical methods
  - Partitioning methods
  - Model-based methods
- Used in many applications
  - Information retrieval
  - Bioinformatics

# Next Class

- Association rule mining
- Reading: http://infolab.stanford.edu/~ullman/mining/assocrules.pdf

# Questions?