# *Unsupervised Learning and Clustering*

Supervised vs. Unsupervised Learning

- Up to now we considered *supervised learning* scenario, where we are given
    1. samples $x_1, \ldots, x_n$
    2. class labels for all samples $x_1, \ldots, x_n$
    - This is also called learning with teacher, since correct answer (the true class) is provided

- Today we consider *unsupervised learning* scenario, where we are only given
    1. samples $x_1, \ldots, x_n$
    - This is also called learning without teacher, since correct answer is not provided
    - do not split data into training and test sets

Clustering

- Seek "natural" clusters in the data
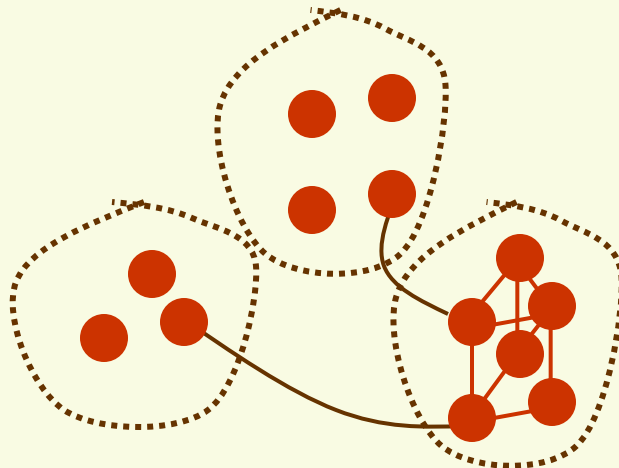
- What is a good clustering?
  - internal (within the cluster) distances should be small
  - external (intra-cluster) should be large
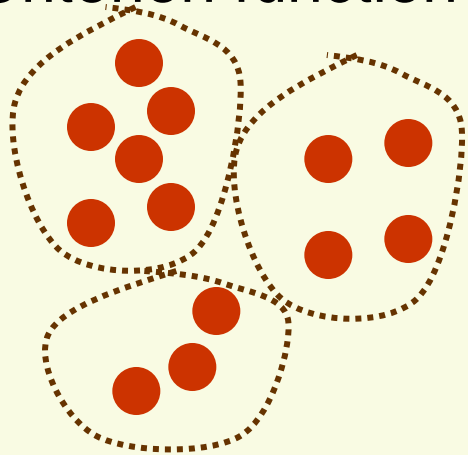- Clustering is a way to discover new categories (classes)

What we Need for Clustering
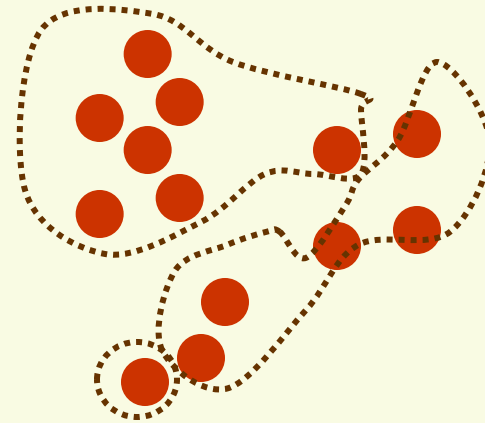
1. Proximity measure, *either*
    - similarity measure $s(x_i, x_k)$: large if x_i, x_k are similar
    - dissimilarity(or distance) measure $d(x_i, x_k)$: small if x_i, x_k are similar

    *large **d**, small **s***   *large **s**, small **d***
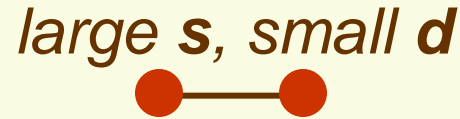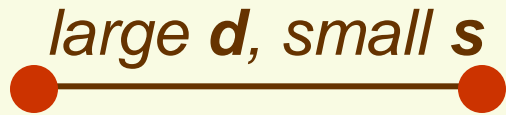
2. Criterion function to evaluate a clustering

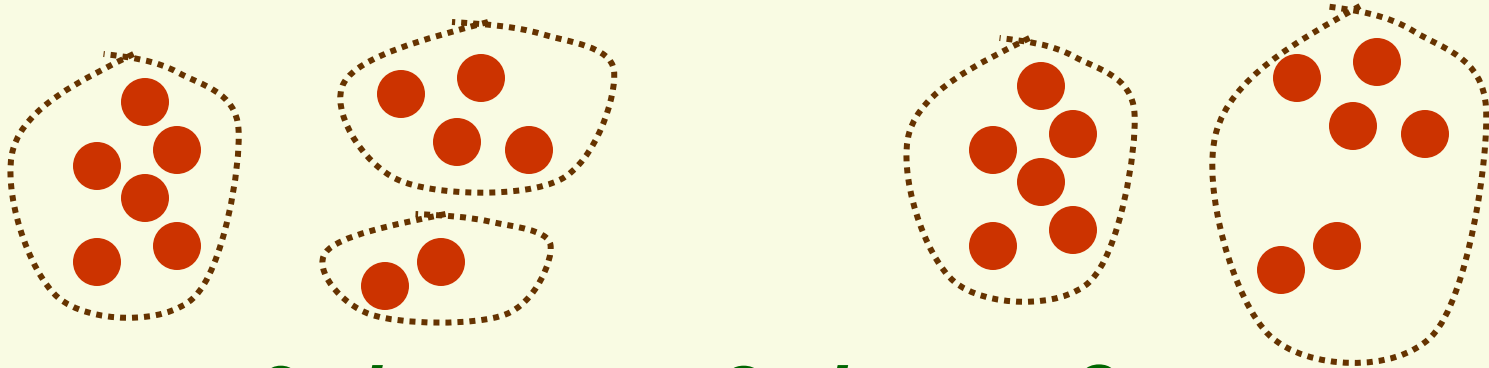   *good clustering*   *bad clustering*

3. Algorithm to compute clustering
    - For example, by optimizing the criterion function

How Many Clusters?

3 clusters or 2 clusters?

- Possible approaches
  1. fix the number of clusters to **k**
  2. find the best clustering according to the criterion function (number of clusters may vary)

Proximity Measures

- good proximity measure is VERY application dependent
  - Clusters should be invariant under the transformations "natural" to the problem
  - For example for object recognition, should have invariance to rotation
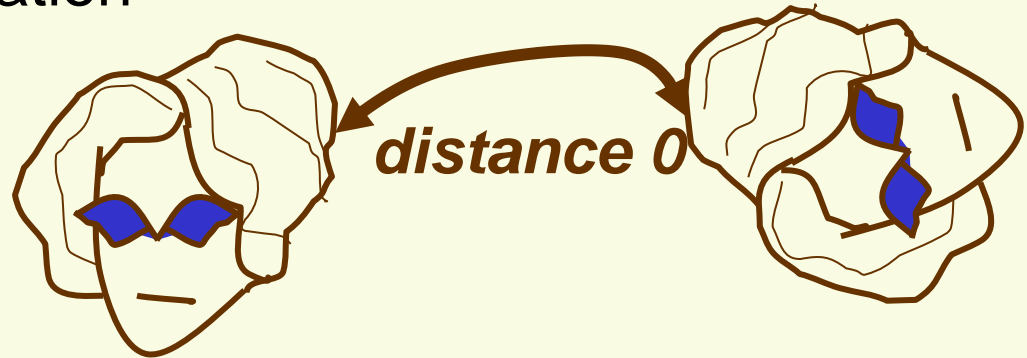
  distance 0

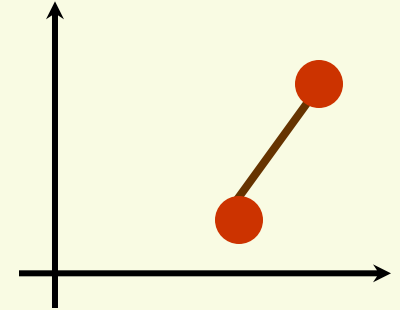 - For character recognition, no invariance to rotation

  large distance

  9   *6*

Distance (dissimilarity) Measures

- Euclidean distance

$$d\left(x_i, x_j\right) = \sqrt{\sum_{k=1}^{d}\left(x_i^{(k)} - x_j^{(k)}\right)^2}$$

  - translation invariant

- Manhattan (city block) distance

$$d\left(x_i, x_j\right) = \sum_{k=1}^{d}\left|x_i^{(k)} - x_j^{(k)}\right|$$

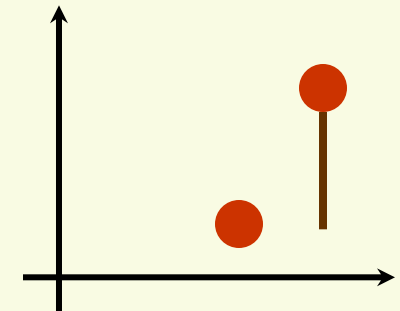 - approximation to Euclidean distance, cheaper to compute
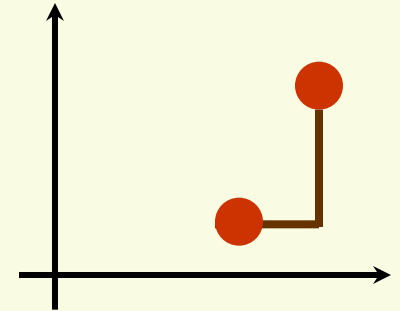
- Chebyshev distance

$$d\left(x_i, x_j\right) = \max_{1 \leq k \leq d} |x_i^{(k)} - x_j^{(k)}|$$

  - approximation to Euclidean distance, cheapest to compute

Similarity Measures
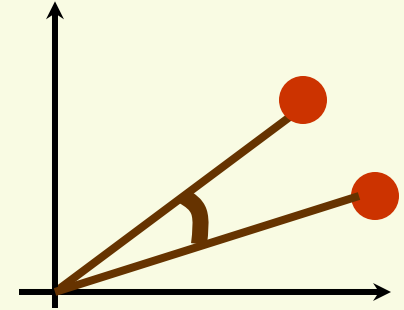
- ## Cosine similarity:

$$s(x_i, x_j) = \frac{x_i^T x_j}{\|x_i\| \|x_j\|}$$
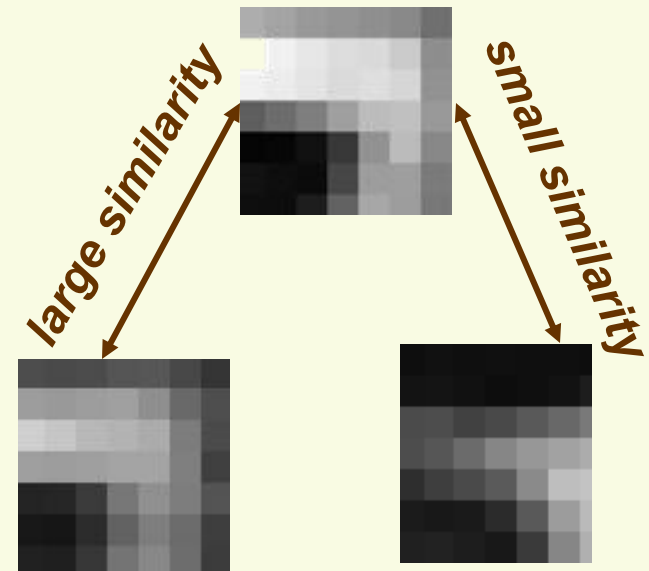
  - the smaller the angle, the larger the similarity
  - scale invariant measure
  - popular in text retrieval

- ## Correlation coefficient
  - popular in image processing

$$s(x_i, x_j) = \frac{\sum_{k=1}^{d}\left(x_i^{(k)} - \overline{x}_i\right)\left(x_i^{(k)} - \overline{x}_i\right)}{\left[\sum_{k=1}^{d}\left(x_i^{(k)} - \overline{x}_i\right)^2 \sum_{k=1}^{d}\left(x_j^{(k)} - \overline{x}_j\right)^2\right]^{1/2}}$$
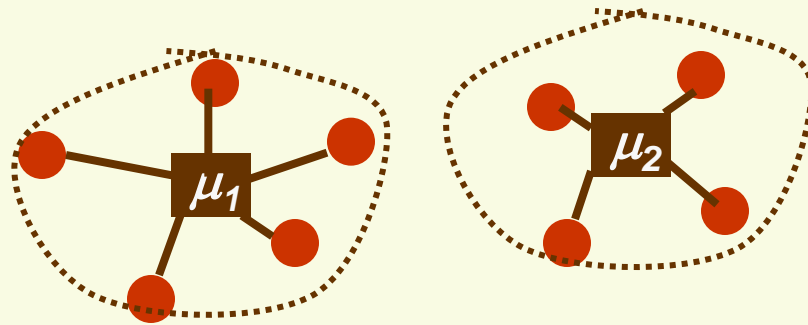
large similarity

small similarity

SSE Criterion Function

- Let n_i be the number of samples in D_i, and define the mean of samples in is D_i

$$\mu_i = \frac{1}{n_i} \sum_{x \in D_i} x$$

- Then the sum-of-squared errors criterion function (to minimize) is:

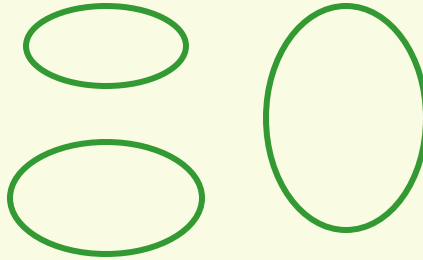$$J_{SSE} = \sum_{i=1}^{c} \sum_{x \in D_i} \| x - \mu_i \|^2$$

- Note that the number of clusters, c, is fixed
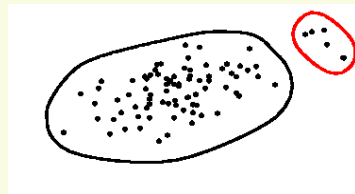
SSE Criterion Function

$$J_{SSE} = \sum_{i=1}^{c} \sum_{x \in D_i} \| x - \mu_i \|^2$$

- SSE criterion appropriate when data forms compact clouds that are relatively well separated
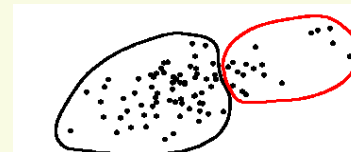
- SSE criterion favors equally sized clusters, and may not be appropriate when "natural" groupings have very different sizes
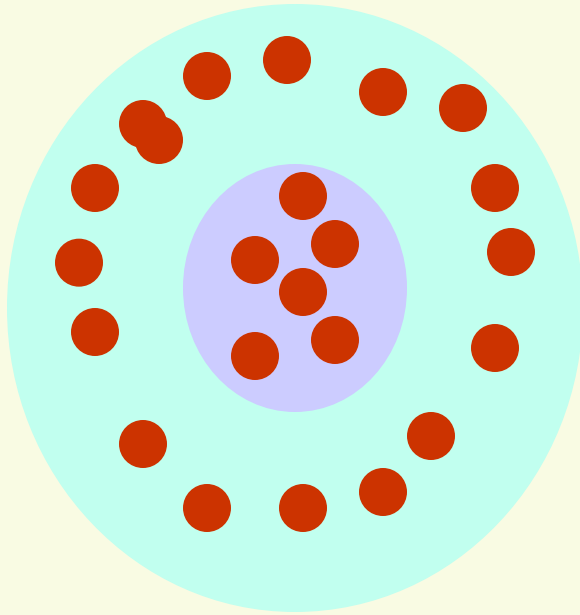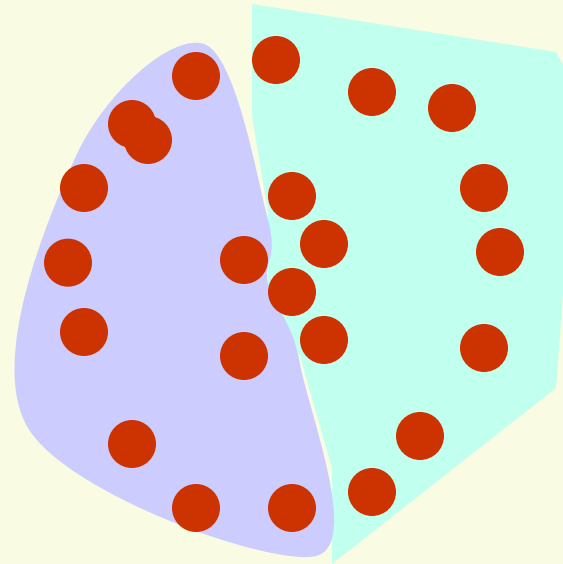
large J_{SSE}   *small J_{SSE}*

Failure Example for J$_{SSE}$

larger J$_{SSE}$ *smaller J$_{SSE}$*

- The problem is that one of the "natural" clusters is not compact (the outer ring)

Other Minimum Variance Criterion Functions

- We can eliminate constant terms from

$$J_{SSE} = \sum_{i=1}^{c} \sum_{x \in D_i} \| x - \mu_i \|^2$$

- We get an equivalent criterion function:

$$J_E = \frac{1}{2} \sum_{i=1}^{c} n_i \left[\frac{1}{n_i^2} \sum_{y \in D_i} \sum_{x \in D_i} \| x - y \|^2 \right]$$

d_i = *average Euclidian distance between all pairs of samples in D_i*

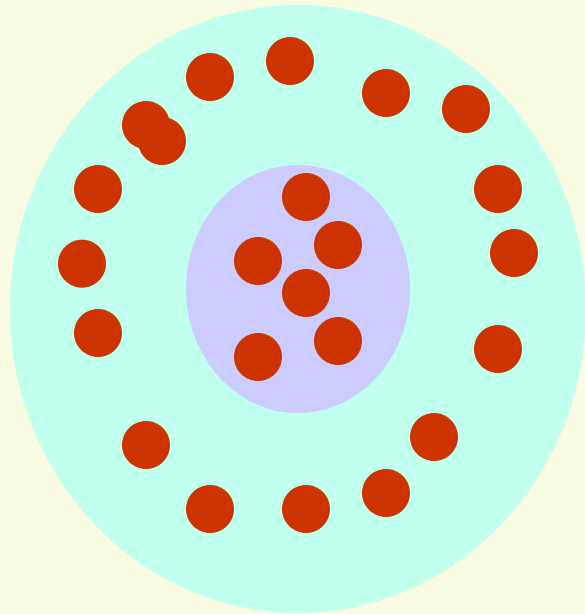- Can obtain other criterion functions by replacing $\|x - y\|^2$ by any other measure of distance between points in D_i

- Alternatively can replace d_i by the median, maximum, etc. instead of the average distance

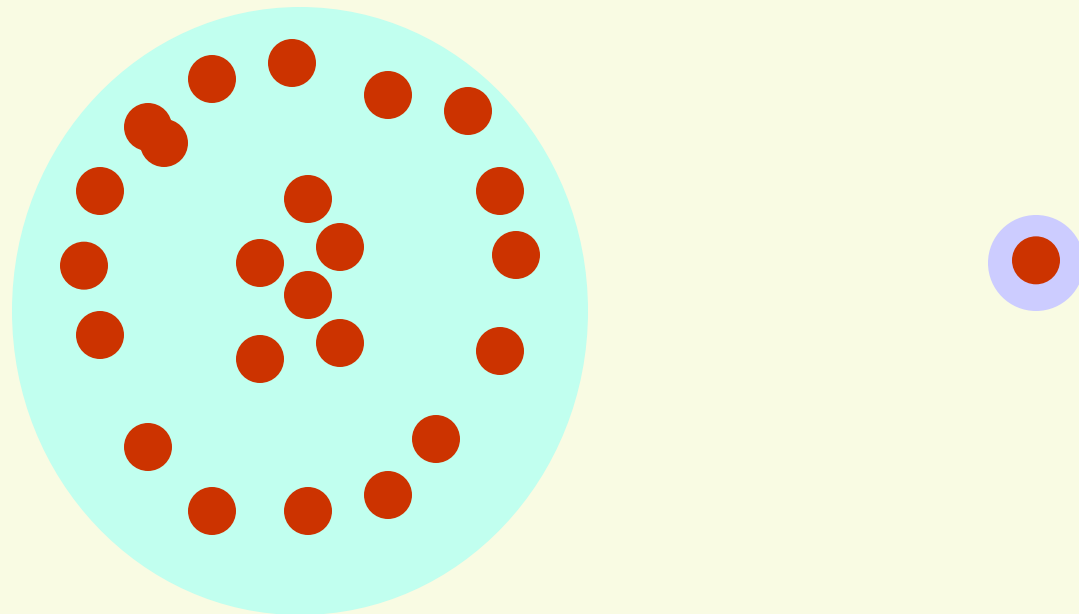# Maximum Distance Criterion

- Consider $\quad J_{max} = \sum_{i=1}^{c} n_i \left[\max_{y \in D_i, x \in D_i} \| x - y \|^2 \right]$

- Solves previous case

- However J_{max} is not robust to outliers

smallest J_{max}

smallest J_{max}

Iterative Optimization Algorithms

- Now have both proximity measure and criterion function, need algorithm to find the optimal clustering

- Exhaustive search is impossible, since there are approximately $c^n/c!$ possible partitions

- Usually some iterative algorithm is used

  1. Find a reasonable initial partition
  2. Repeat: *move samples from one group to another s.t. the objective function J is improved*

move

samples to improve J
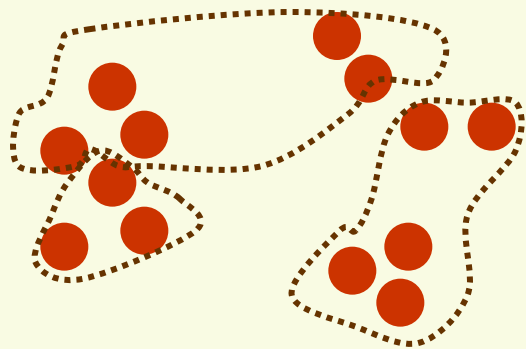
$J = 777,777$
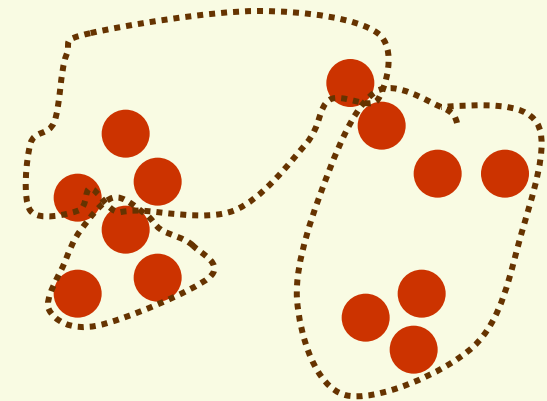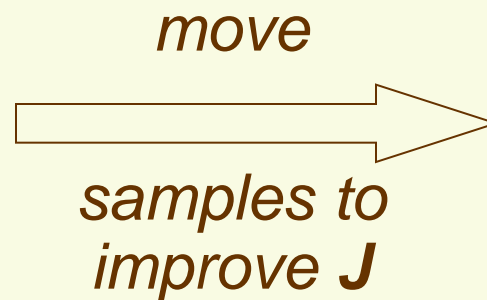
$J = 666,666$

K-means Clustering

- We now consider an example of iterative optimization algorithm for the special case of J_{SSE} objective function

$$J_{SSE} = \sum_{i=1}^{k} \sum_{x \in D_i} \| x - \mu_i \|^2$$
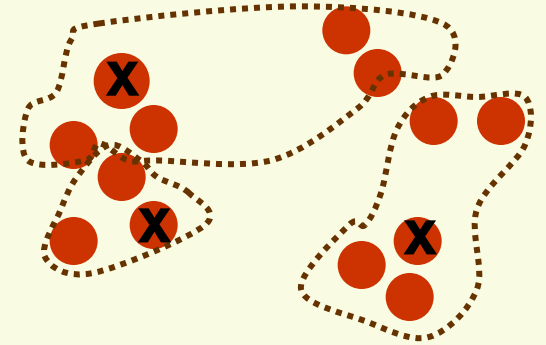
  - for a different objective function, we need a different optimization algorithm, of course

- Fix number of clusters to k ($c = k$)

- k-means is probably the most famous clustering algorithm

  - it has a smart way of moving from current partitioning to the next one

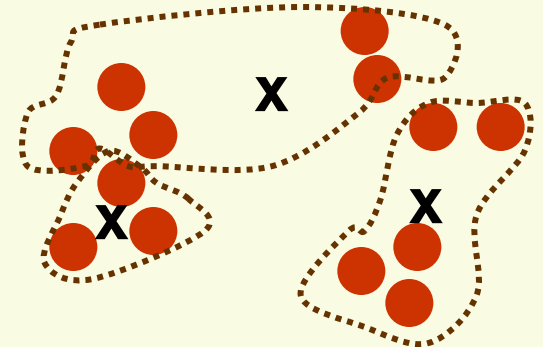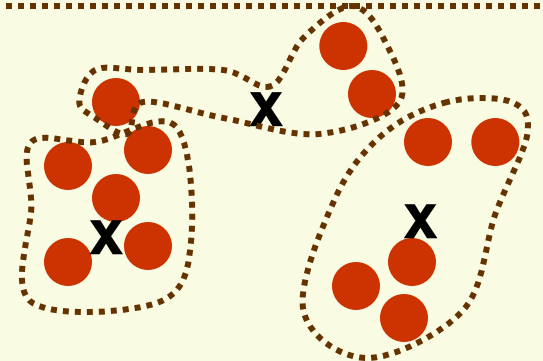# *K-means Clustering*

1. Initialize
   - pick *k* cluster centers arbitrary
   - assign each example to closest center

2. compute sample means for each cluster
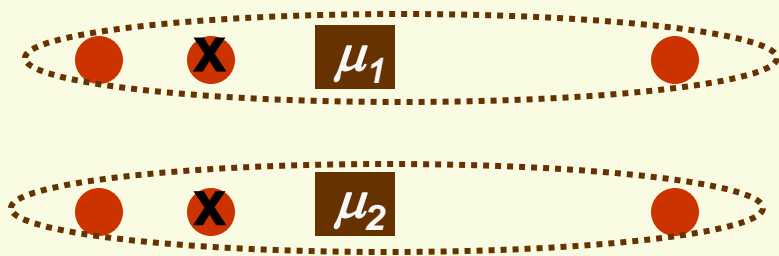
3. reassign all samples to the closest mean

4. if clusters changed at step 3, go to step 2

K-means Clustering

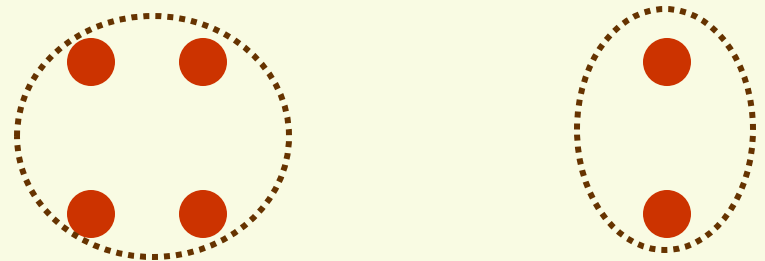- Thus the algorithm converges after a finite number of iterations of steps *2* and *3*

- However the algorithm is not guaranteed to find a global minimum

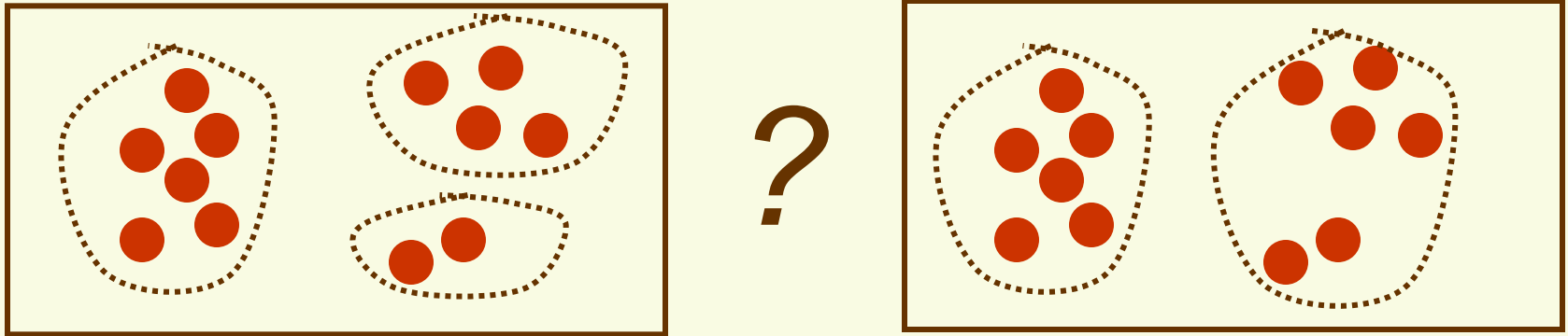*2-means gets stuck here* | *global minimum of J_{SSE}*

Hierarchical Clustering

- Up to now, considered "flat" clustering

?

- For some data, hierarchical clustering is more appropriate than "flat" clustering

- Hierarchical clustering
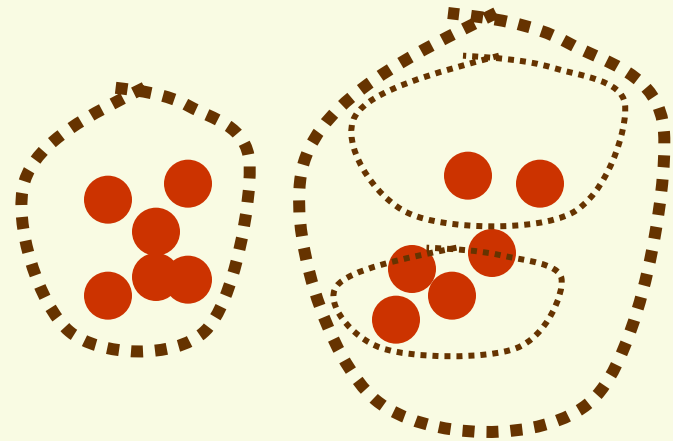
Hierarchical Clustering: Dendogram

- preferred way to represent a hierarchical clustering is a dendrogram

  - Binary tree
  - Level k corresponds to partitioning with $n-k+1$ clusters

  - if need k clusters, take clustering from level $n-k+1$

  - If samples are in the same cluster at level k, they stay in the same cluster at higher levels

  - dendrogram typically shows the similarity of grouped clusters
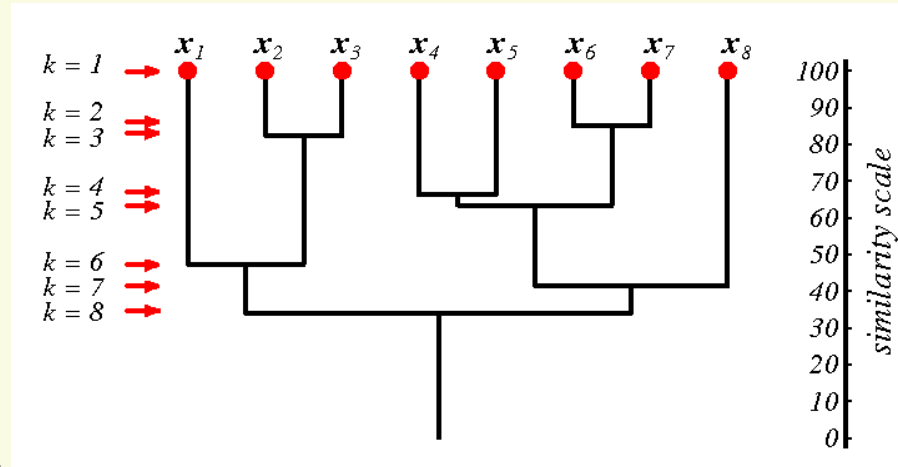
Example

Hierarchical Clustering

- Algorithms for hierarchical clustering can be divided into two types:

    1. Agglomerative (bottom up) procedures
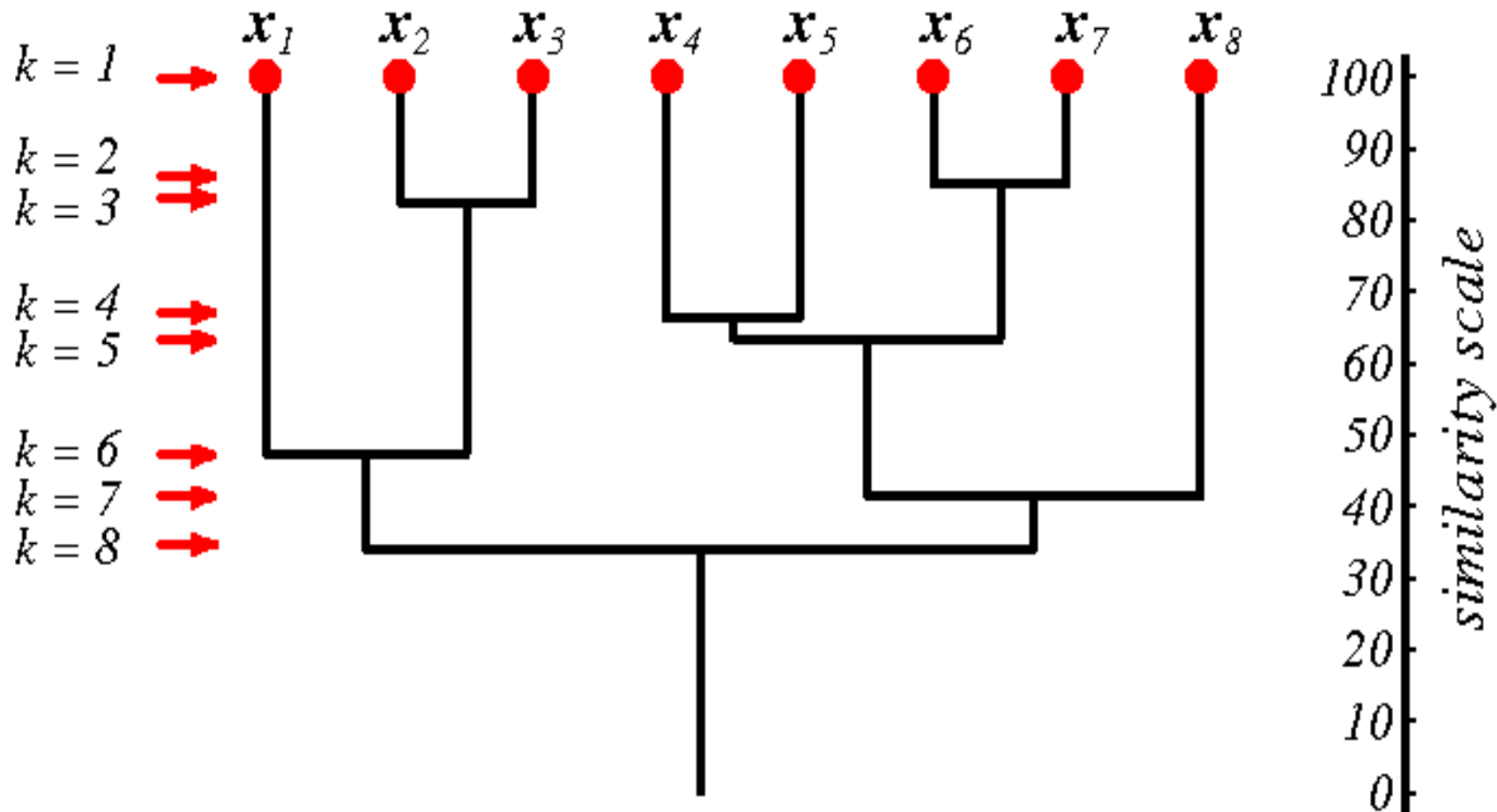   - Start with *n* singleton clusters
        - Form hierarchy by merging most similar clusters
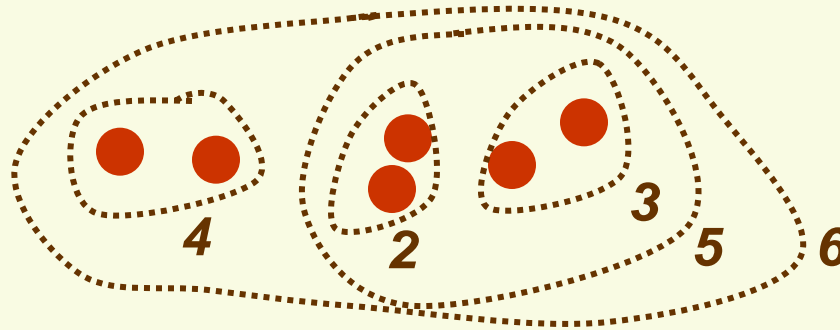
    2. Divisive (top bottom) procedures
        - Start with all samples in one cluster
        - Form hierarchy by splitting the "worst" clusters

Divisive Hierarchical Clustering

- Any "flat" algorithm which produces a fixed number of clusters can be used
  - set $c = 2$
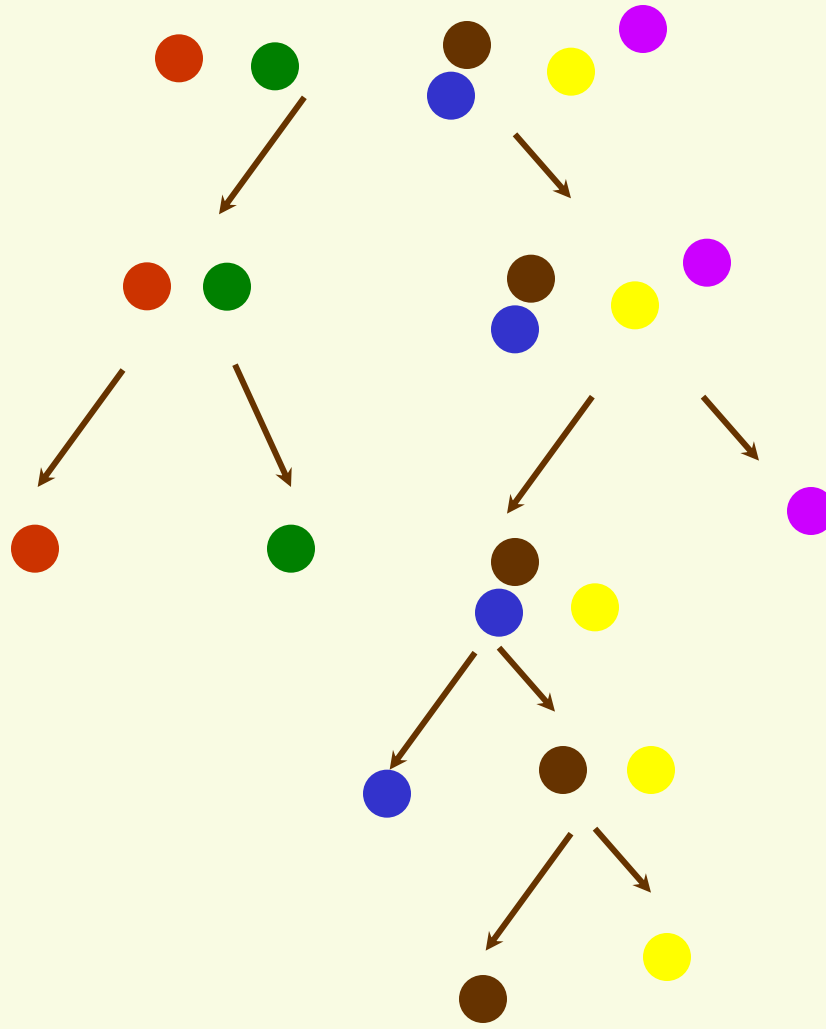
Agglomerative Hierarchical Clustering

initialize with each example in singleton cluster

while there is more than *1* cluster

1. find 2 nearest clusters
2. merge them
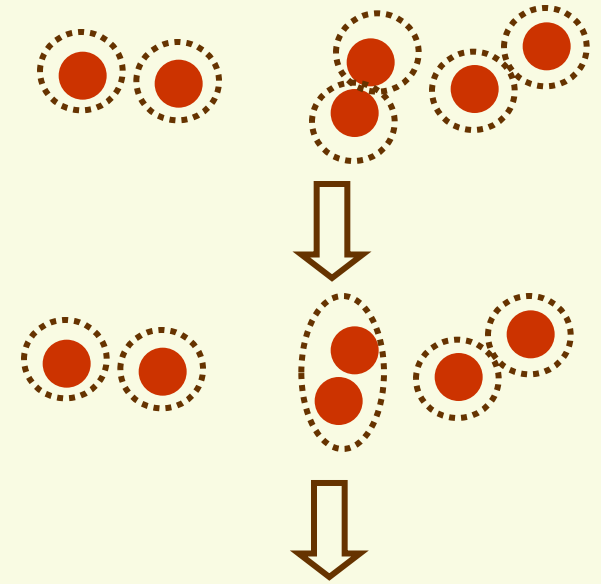
- Four common ways to measure cluster distance

1. minimum distance $d_{\min}(D_i, D_j) = \min_{x \in D_i, y \in D_j} \| x - y \|$

2. maximum distance $d_{\max}(D_i, D_j) = \max_{x \in D_i, y \in D_j} \| x - y \|$

3. average distance $d_{avg}(D_i, D_j) = \dfrac{1}{n_i n_j} \sum_{x \in D_i} \sum_{y \in D_j} \| x - y \|$

4. mean distance $d_{mean}(D_i, D_j) = \| \mu_i - \mu_j \|$

Single Linkage or Nearest Neighbor

- Agglomerative clustering with minimum distance

$$d_{\min}(D_i, D_j) = \min_{x \in D_i, y \in D_j} \| x - y \|$$
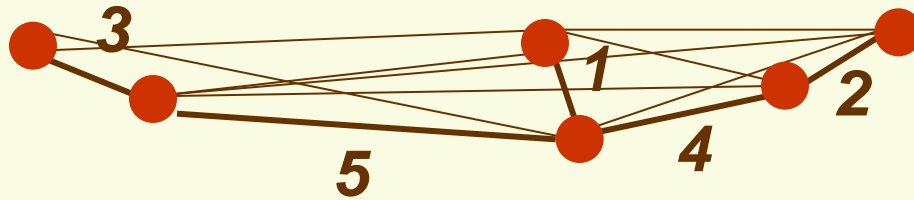
- generates minimum spanning tree
- encourages growth of elongated clusters
- disadvantage: very sensitive to noise

what we want at level with c=3

what we get at level with c=3

noisy sample

Complete Linkage or Farthest Neighbor

- Agglomerative clustering with maximum distance

$$d_{\max}(D_i, D_j) = \max_{x \in D_i, y \in D_j} \| x - y \|$$

- encourages compact clusters

- Does not work well if elongated clusters present
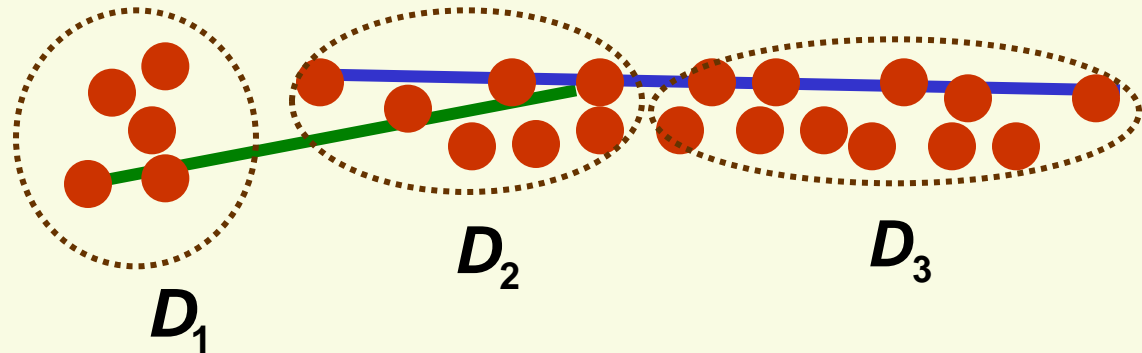
$$d_{\max}(D_1, D_2) < d_{\max}(D_2, D_3)$$

- thus D_1 and D_2 are merged instead of D_2 and D_3

Average and Mean Agglomerative Clustering
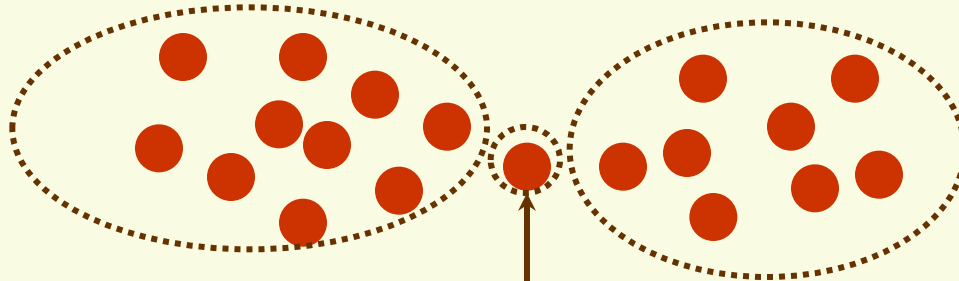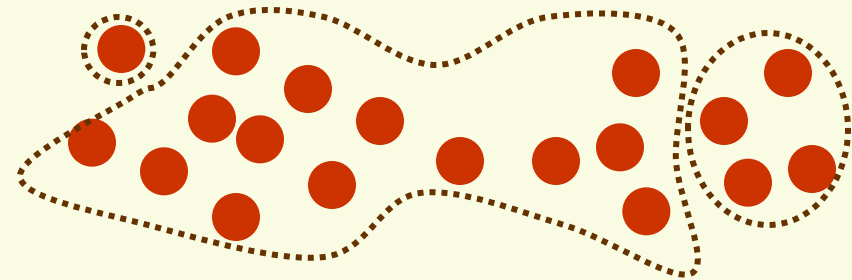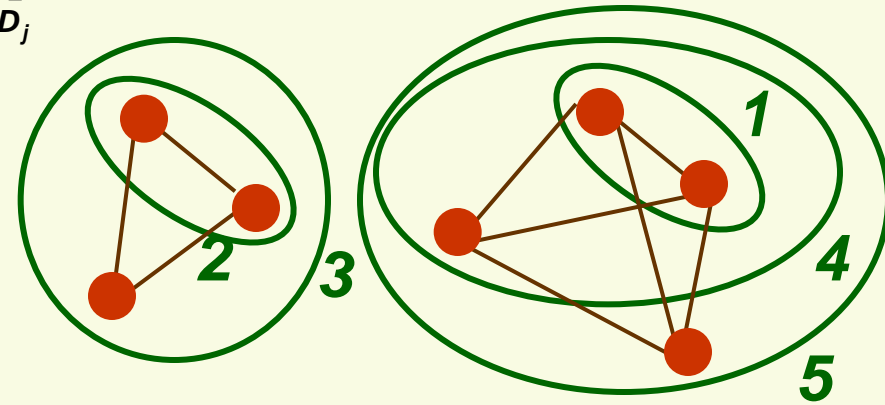
- Agglomerative clustering is more robust under the average or the mean cluster distance

$$d_{avg}(D_i, D_j) = \frac{1}{n_i n_j} \sum_{x \in D_i} \sum_{y \in D_j} \| x - y \|$$

$$d_{mean}(D_i, D_j) = \| \mu_i - \mu_j \|$$

- mean distance is cheaper to compute than the average distance

- unfortunately, there is not much to say about agglomerative clustering theoretically, but it does work reasonably well in practice

Agglomerative vs. Divisive

- Agglomerative is faster to compute, in general
- Divisive may be less "blind" to the global structure of the data

Divisive	*Agglomerative*
when taking the first step (split), have access to all the data; can find the best possible split in 2 parts	when taking the first step merging, do not consider the global structure of the data, only look at pairwise structure
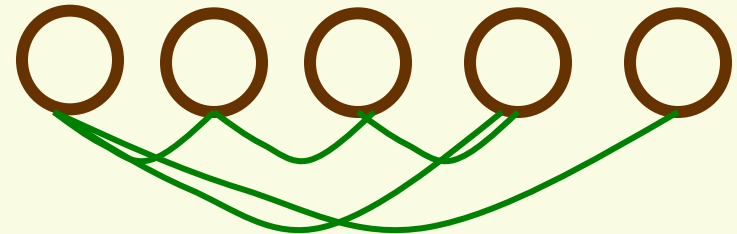
First (?) Application of Clustering

- John Snow, a London physician plotted the location of cholera deaths on a map during an outbreak in the 1850s.

- The locations indicated that cases were clustered around certain intersections where there were polluted wells -- thus exposing both the problem and the solution.

From: Nina Mishra HP Labs
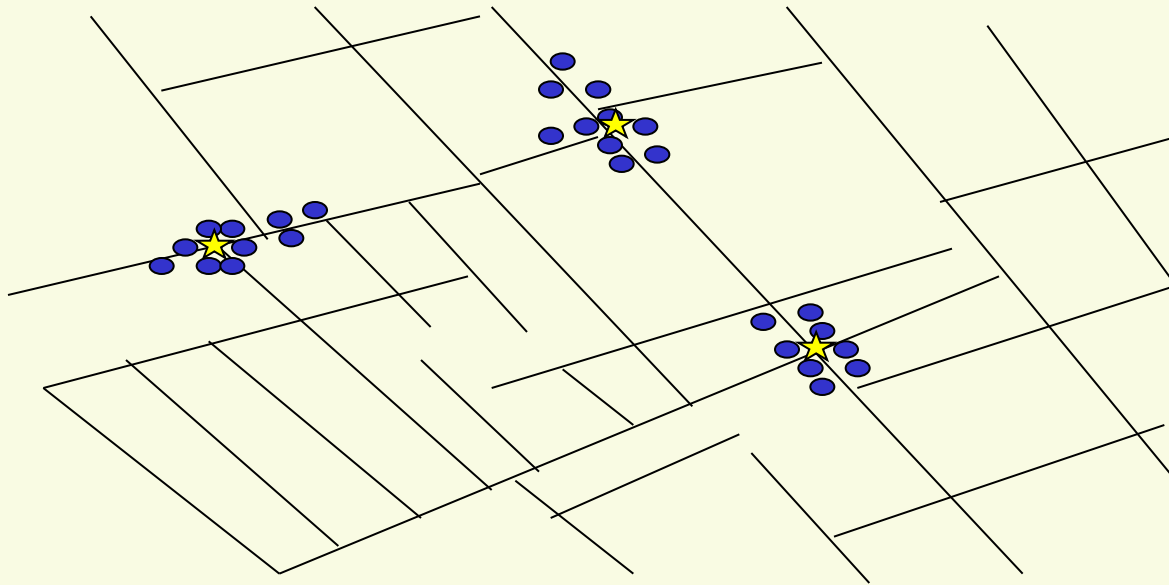
Application of Clustering

- Astronomy

  - SkyCat: Clustered 2×10^9 sky objects into stars, galaxies, quasars, etc based on radiation emitted in different spectrum bands.

From: Nina Mishra HP Labs

Applications of Clustering

- Image segmentation
  - Find interesting "objects" in images to focus attention at

From: Image Segmentation by Nested Cuts, O. Veksler, CVPR2000
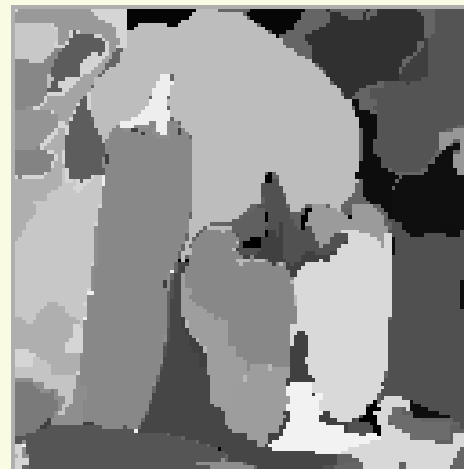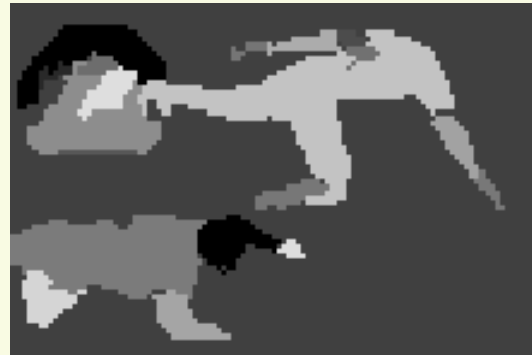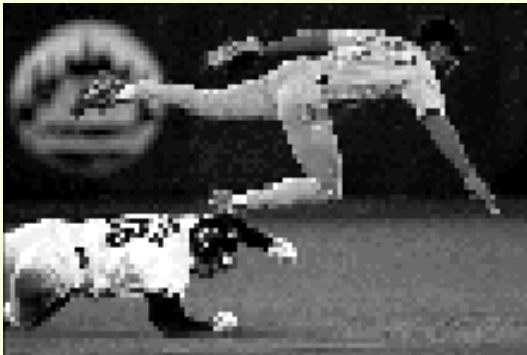
Applications of Clustering

- Image Database Organization
  - for efficient search
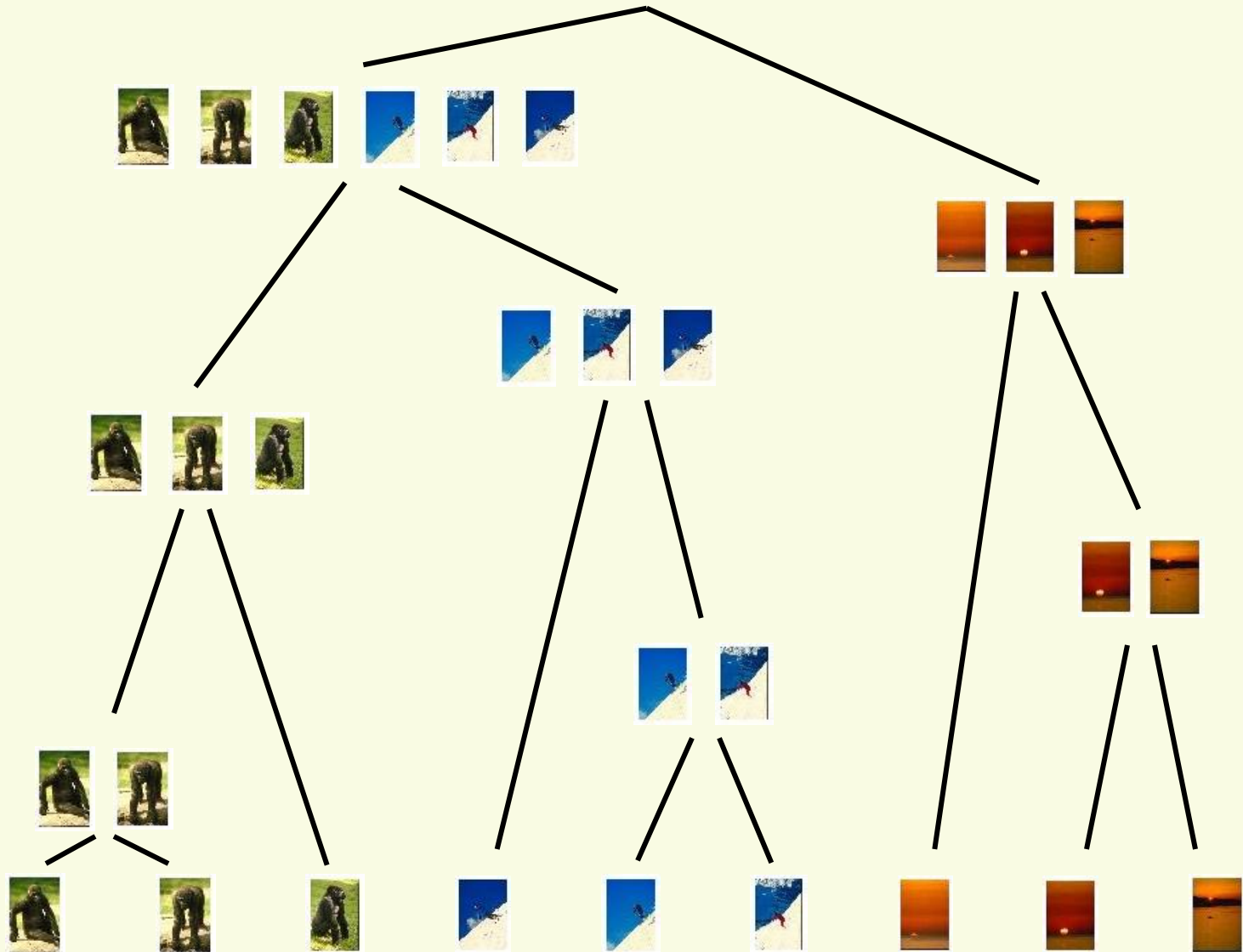
Applications of Clustering

- Data Mining
  - Technology watch
    - Derwent Database, contains all patents filed in the last 10 years worldwide
    - Searching by keywords leads to thousands of documents
    - Find clusters in the database and find if there are any emerging technologies and what competition is up to
  - Marketing
    - Customer database
    - Find clusters of customers and tailor marketing schemes to them

Applications of Clustering

- ## gene expression profile clustering

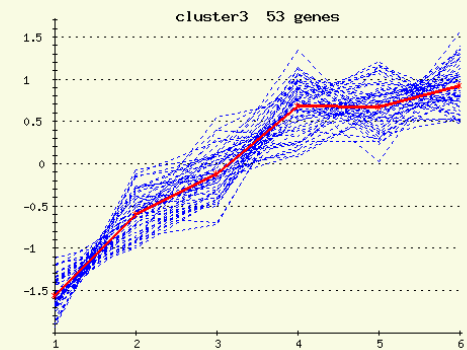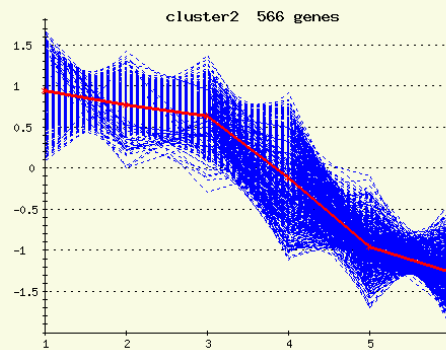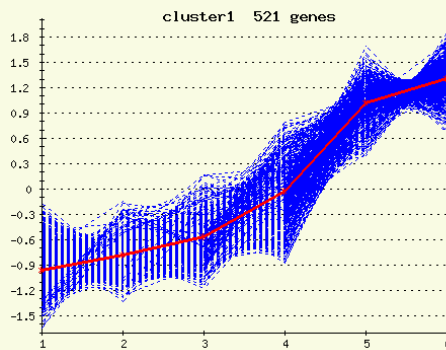 - ### similar expressions , expect similar function

  U18675 4CL -0.151 -0.207 0.126 0.359 0.208 0.091 -0.083 -0.209
  M84697 a-TUB 0.188 0.030 0.111 0.094 -0.009 -0.173 -0.119 -0.136
  M95595 ACC2 0.000 0.041 0.000 0.000 0.000 0.000 0.000 0.000
  X66719 ACO1 0.058 0.155 0.082 0.284 0.240 0.065 -0.159 -0.010
  U41998 ACT 0.096 -0.019 0.070 0.137 0.089 0.038 0.096 -0.070
  AF057044 ACX1 0.268 0.403 0.679 0.785 0.565 0.260 0.203 0.252
  AF057043 ACX2 0.415 0.000 -0.053 0.114 0.296 0.242 0.090 0.230
  U40856 AIG1 0.096 -0.106 -0.027 -0.026 -0.005 -0.052 0.054 0.006
  U40857 AIG2 0.311 0.140 0.257 0.261 0.158 0.056 -0.049 0.058
  AF123253 AIM1 -0.040 0.002 -0.202 -0.040 0.077 0.081 0.088 0.224
  X92510 AOS 0.473 0.560 0.914 0.625 0.375 0.387 0.019 0.141

From:De Smet F., Mathys J., Marchal K., Thijs G., De Moor B. & Moreau Y. 2002.
Adaptive Quality-based clustering of gene expression profiles, Bioinformatics, **18**(6), 735-746.

Applications of Clustering

- Profiling Web Users
  - Use web access logs to generate a feature vector for each user
  - Cluster users based on their feature vectors
  - Identify common goals for users
    - Shopping
    - Job Seekers
    - Product Seekers
    - Tutorials Seekers
  - Can use clustering results to improving web content and design

Summary

- Clustering (nonparametric unsupervised learning) is useful for discovering inherent structure in data

- Clustering is immensely useful in different fields

- Clustering comes naturally to humans (in up to 3 dimensions), but not so to computers

- It is very easy to design a clustering algorithm, but it is very hard to say if it does anything good

- General purpose clustering does not exist, for best results, clustering should be tuned to application at hand