CS 383 – Machine Learning

Clustering

Slides adapted from material created by E. Alpaydin
Prof. Mordohai, Prof. Greenstadt, Pattern Classification (2nd Ed.),
Pattern Recognition and Machine Learning

Overview

- Unsupervised Learning
- Clustering
  - K-Means/mediods
  - Agglomerative Clustering
- Reading
  - Springer Section 10.3

Supervised vs. Unsupervised Learning

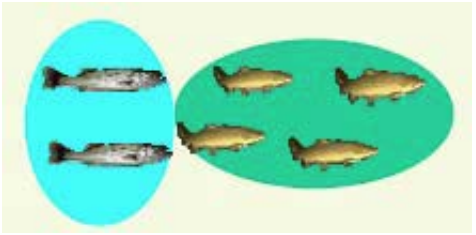
- In general our data comes in two flavors:
  1. Supervised
  2. Unsupervised
- With *supervised* data we have not only the observations, $\{X_i\}_{i=1}^N$ but also associated labels, $\{Y_i\}_{i=1}^N$.
  - Together these form our dataset: $\{X_i, Y_i\}_{i=1}^N$
  - We will later use this type of data to do
    - Regression
    - Classification
- With *unsupervised* data we only have the observations, $\{X_i\}_{i=1}^N$

Why Unsupervised Learning?

- It's harder ☹
  - How do we know if results are meaningful since there no answers (labels) to test against?
- We need it though
  - Labeling large datasets is very costly
  - May have no idea what/how many classes there are (data mining)
  - May want to use clustering to gain some insight into the structure of the data before designing a classifier

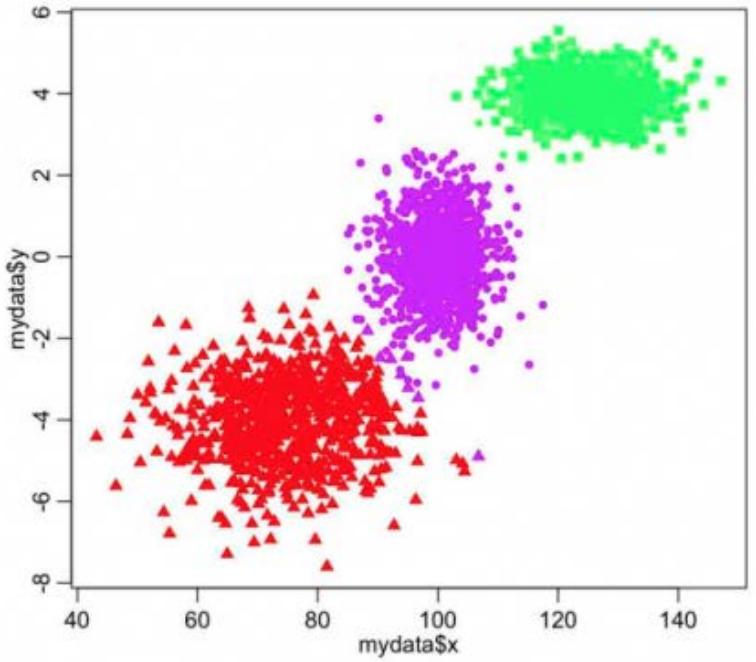
Unsupervised Learning

- The most common type of unsupervised learning is *clustering* where we attempt to learn underlying patterns from data



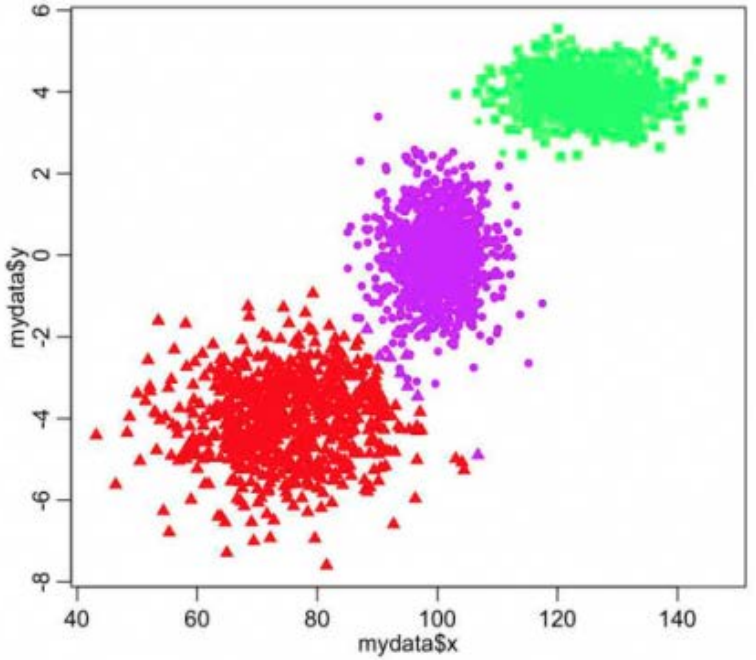
What is Clustering?

- Clustering: The process of grouping a set of objects into classes of similar objects
  - Items within cluster should be similar
  - Observations from different clusters should be dissimilar
- This is the most common form of *unsupervised learning*



Issues for Clustering

- Need a notion of similarity/distance
  - Similarity: Gaussian, Cosine
  - Distance: L2 (Euclidian), L1 (Manhattan)
  - See Blackboard for these
- How many clusters?
  - Fixed a priori?
  - Completely data driven?
    - Avoid “trivial” clusters – too small or too large
    - Use some statistics?
- How to evaluate cluster quality?
  - It’s unsupervised after all….



Hard vs Soft Clustering

- Hard clustering: Each observation belongs to exactly one cluster
  - More common and easier to do
- Soft clustering: An observation can belong to more than one cluster
  - And/or can belong to clusters with some probability

Clustering Algorithms

- Flat algorithms
  - Usually start with a random clustering/model
  - Refine it iteratively
- Hierarchical algorithms
  - Bottom-up, agglomerative
  - Top-down, divisive

K-Means

K-Means

- Assumes features are continuous.
- Each observation is associated with a single *reference vector*.
  - This is typically the reference vector that the observation is closest to, or most similar to.
  - For simplicity/intuition, we'll compare observations using Euclidean distance.
    - Therefore we assign an observation to the reference vector it is closest to.

K-Means Pseudocode

1. Select k vectors $\{a_1, a_2, \ldots, a_k\}$ as the initial reference vectors.
2. Until clustering converges, or other stopping condition:
    1. For each observation x
        1. Compare x to each reference vector, and associate x with the reference vector it is most similar to/closest to.
        2. Update the reference vectors based on the observations associated with it. If we're doing k-means, the reference vector will be updated to be the *mean* of the observations associated with it. If C_i is the set of observations associated with reference vector a_i then:

$$a_i \rightarrow \frac{1}{|C_i|} \sum_{x \in C_i} x$$

K-Means Example

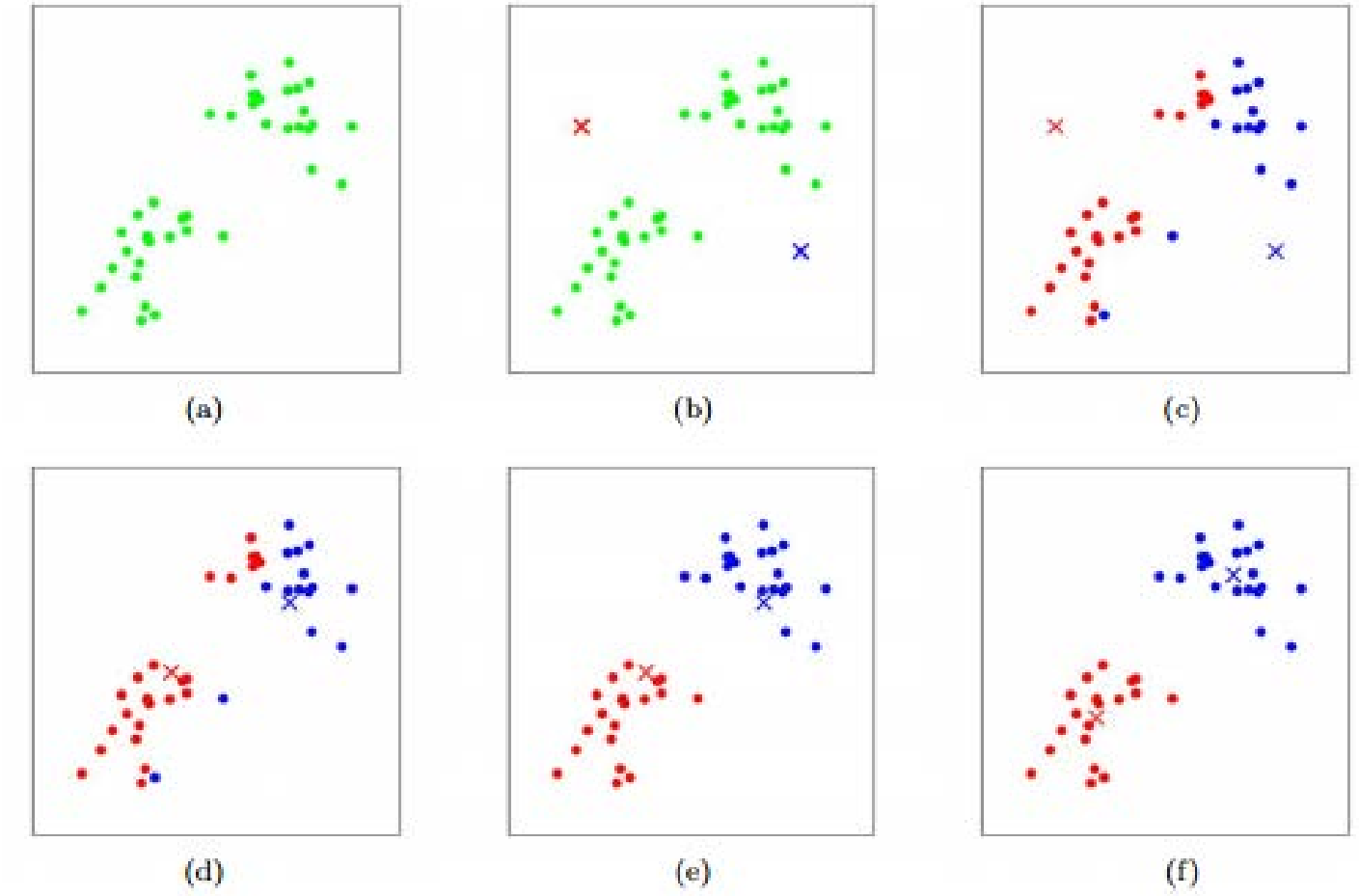


Image from Stanford.edu

k-means

- A few design considerations:
  - What should k be?
    - Application likely dictates this
  - What similarity /distance function to use?
    - Nature of the features likely dictates this
  - Where should the reference vectors start?
    - See next slide
  - What are some termination criteria?
    - See next slide

Initial Reference Vector Choice

- Results can vary based on initial reference vector choice.
- Some choices can result in poor convergence rate, or convergence to sub-optimal clustering
- Some ideas might be:
  - Select good initial reference vectors using a heuristic (e.g. instances least similar to any existing mean)
  - Try out multiple starting points
  - Initialize with the results of another method

Termination Conditions

- Several possibilities
  1. A fixed number of iterations
  2. Between iterations
     1. Cluster assignments don't change
     2. Reference vectors don't change (by much)

Derivation of K-Means

- Why is the mean of the observations in a cluster a good reference vector for it?
- K-Means is based on using the sum of the square of the distances to measure the "goodness" of our solution:

$$J(a_i) = \sum_{x \in C_i} (x - a_i)^2$$

- This is referred to as the *least squared error*.
- Since x and a_i are actually *vectors*, this equation should be written as:

$$J(a_i) = \sum_{x \in C_i} (x - a_i)(x - a_i)^T$$

Derivation of K-Means

- We want to find a value for the reference vector a_i that minimizes this distance.
- So we'll take the derivative of $J(a_i)$ with respect to a_i

$$\frac{dJ(a_i)}{da_i} = \frac{d}{da_i}\left(\sum_{x \in C_i}(x - a_i)(x - a_i)^T\right) = \sum_{x \in C_i} -2(x - a_i)$$

- Now let's set this equal to zero:

$$\sum_{x \in C_i} -2(x - a_i) = 0$$

- We can re-write this as

$$\sum_{x \in C_i} x = \sum_{x \in C_i} a_i$$

Derivation of K-Means

$$\sum_{x \in C_i} x = \sum_{x \in C_i} a_i$$

- Let $|C_i|$ be the number of members in cluster C_i
- Then $\sum_{x \in C_i} a_i = |C_i| a_i$
- Therefor (via substitution):

$$a_i = \left(\frac{1}{|C_i|} \sum_{x \in C_i} x \right)$$

- Which is the definition of the mean of the cluster, $\mu(C_i)$
- So if $a_i = \mu(C_i)$ then we have minimized the least squared error

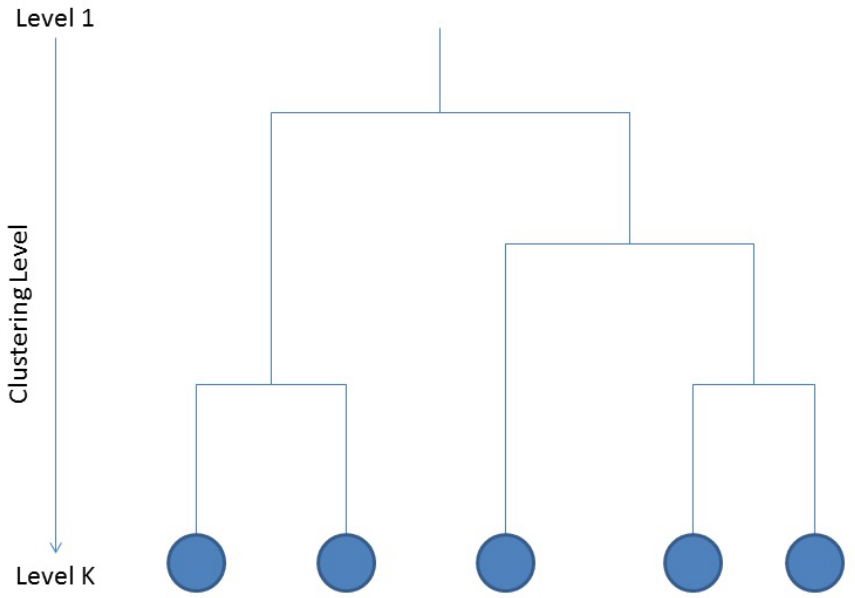
Weaknesses of k-Means

- The algorithms is only applicable if the *mean* is defined
  - For **categorical** data use **k-mode** where the centroid is represented by most frequent values
- The user needs to specify k
- The algorithm is sensitive to outliers
  - Outliers are data points that are very far away from other data points
  - Outliers could be errors in the data recording or some special data points with very different values
  - One solution is to use **k-medoids** (which uses the L1 distance and chooses the median of each feature)

Hierarchical Clustering

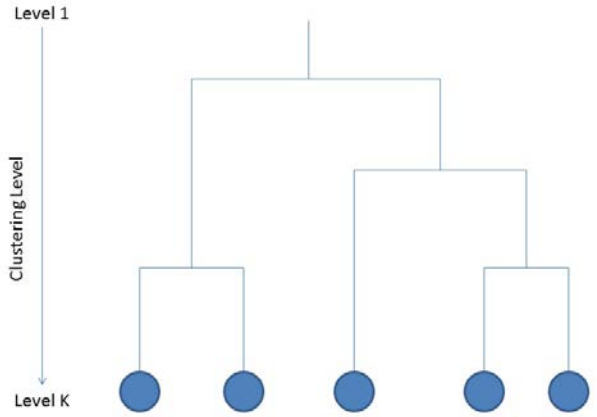
Hierarchical Agglomerative Clustering (HAC)

- With hierarchical agglomerative clustering we're building a clustering binary tree
- This can be done either bottom up, or top down
- At each iteration we have a new set of clusters



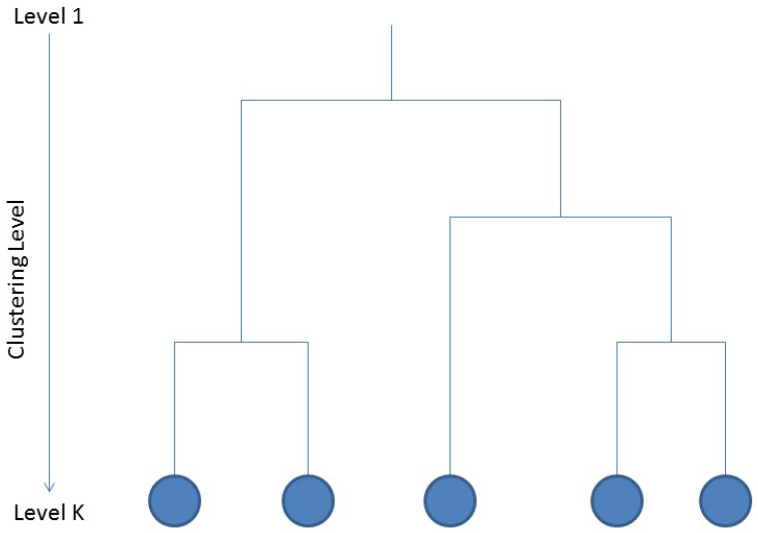
Hierarchical Agglomerative Clustering (HAC)

- Top-Down Approach
  - At first everything is part of a single cluster.
  - Then split this into two clusters based on some criterial
  - Now choose one of these two clusters, and split it into two
    - Now we have three total clusters
  - Etc.. until each cluster only has one observation in it (called a singleton)



Hierarchical Agglomerative Clustering (HAC)

- Bottom-Up Approach
  - At first everything is its own cluster
    - If there's N observations, then there's N clusters
  - Choose two of these clusters to merge
    - Now there's $N - 1$ clusters
  - From these $N - 1$ clusters, choose two to merge
    - Now there's $N - 2$ clusters
  - Etc.. until there is only one cluster
- Let's just look at the bottom-up approach.



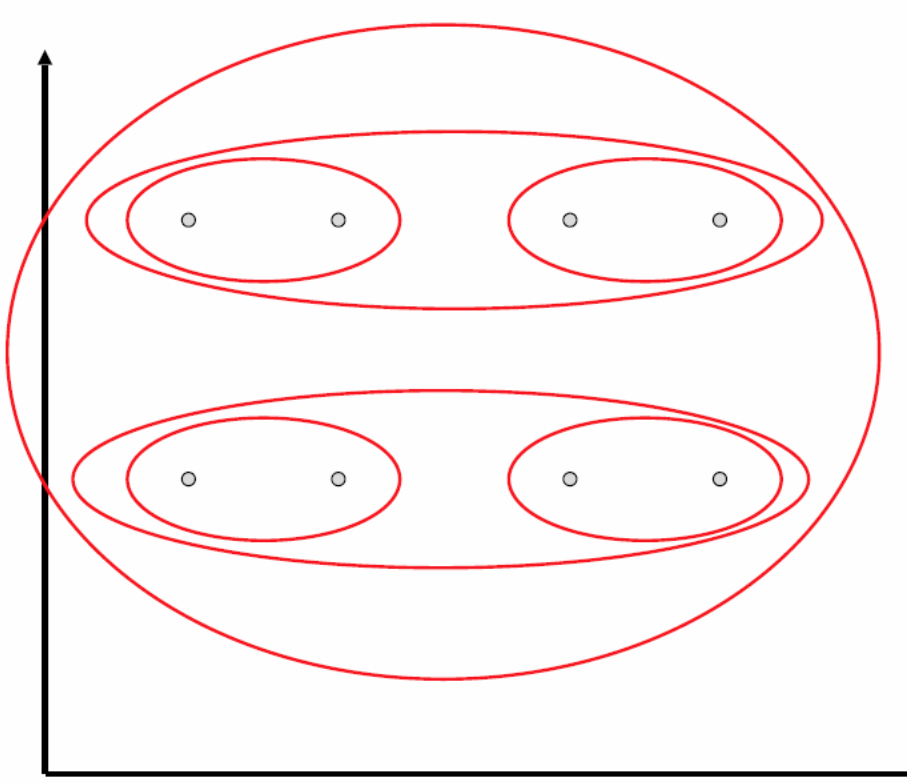
Closest Pair of Clusters

- When deciding which to clusters to merge we typically need to determine which two clusters are closest.
- There's many variants to defining closest pair of clusters
  - Single link – Similarity of the most similar
  - Complete link – Similarity of the furthest points
  - Average link – Average pair-wise similarity between clusters

Single Link Example

Single link – Similarity of the most similar

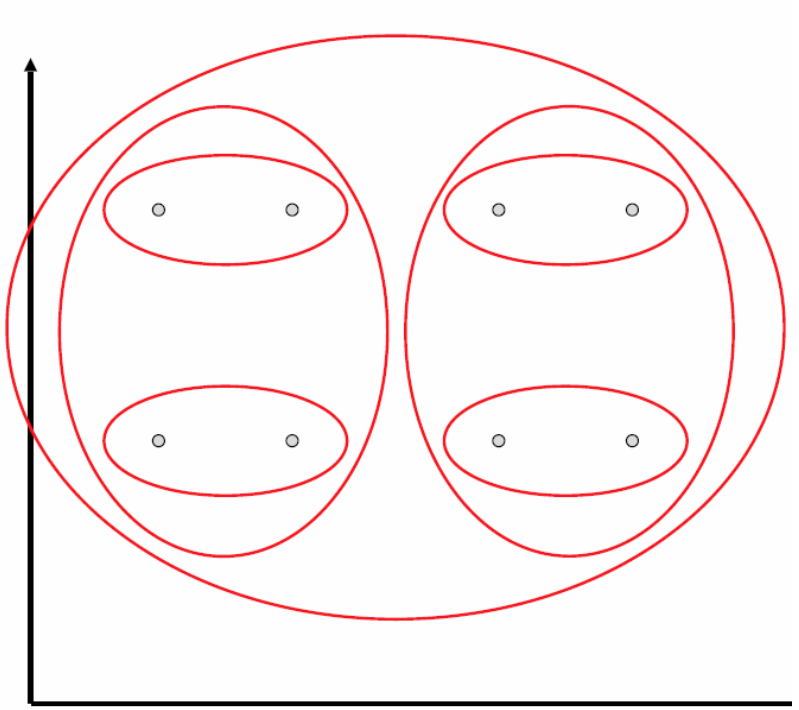
$$sim(C_i, C_j) = \max_{x \in C_i, y \in C_j} sim(x, y)$$



Complete Link HAC

Complete link – Similarity of the furthest points

$$sim(C_i, C_j) = \min_{x \in C_i, y \in C_j} sim(x, y)$$



Average HAC

- Average Link
  - Compromise between single and complete link
  - Average over all pairs *between* the two original clusters

$$sim(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} sim(x, y)$$

HAC Pseudocode

- Let $C^{(k)}$ be the set of clusters at clustering level k and $C_i^{(k)}$ to be the i^{th} cluster in cluster-set $C^{(k)}$
- Initialize cluster level, $k = N$ and cluster-set $C_i^{(k)} = \{x_i\}$ for $i = 1, \ldots N$
- While $k \geq 1$
  - Find closest clusters in $C^{(k)}$, $C_a^{(k)}$, $C_b^{(k)}$ according to some metric.
  - Create new cluster set $C^{(k-1)} = C^{(k)}$
  - Remove from $C^{(k-1)}$, $\mathrm{C_a^{(k-1)}}$ and $C_b^{(k-1)}$ and add $C_a^{(k)} \cup C_b^{(k)}$
  - $k \rightarrow k - 1$

Choosing The Clustering Level

- Ok so we have a HAC tree.
- What can we use it for?
- If we know how many clusters we want (if the problem dictates k), then we just choose the level of the tree that has that many clusters.
- What if we don't know it a-priori?

Clustering Quality

- We need to provide some way to measure the quality of a given clustering
- A good clustering will produce clusters in which
  - The intra-class (that is, intra-cluster) similarity is high
  - The inter-class similarity is low

Intra-Cluster Distance

- For a given cluster i and a chosen distance/similarity function d, the follow computes the average pairwise intra-cluster distance G_i

$$G_i = \frac{\sum_{x,y \in C_i} d(x,y)}{(2|C_i|)}$$

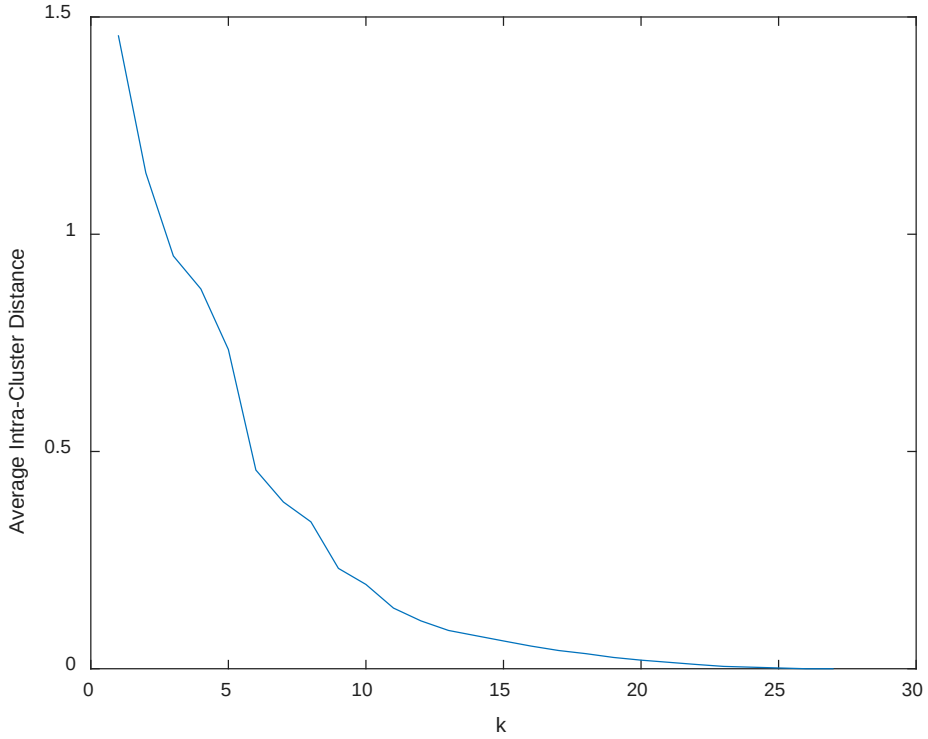
- The weighed (by cluster size) average intra-cluster distance for clustering level j is then:

$$W_j = \frac{\sum_{i=1}^{j} |C_i| G_i}{N}$$

- One idea might be to look at how the weighed average intra-cluster distance varies as a function of the number of clusters.

Graph Based Approaches

- Below is a graph showing the weighted average intra-cluster distance
- There are 30 observations
  - Clustering level 1 has everything in one cluster, and thus a large weighted intra-cluster distance
  - Cluster level 27 has everything as its own cluster, and thus a weighed average intra-cluster distance of zero.



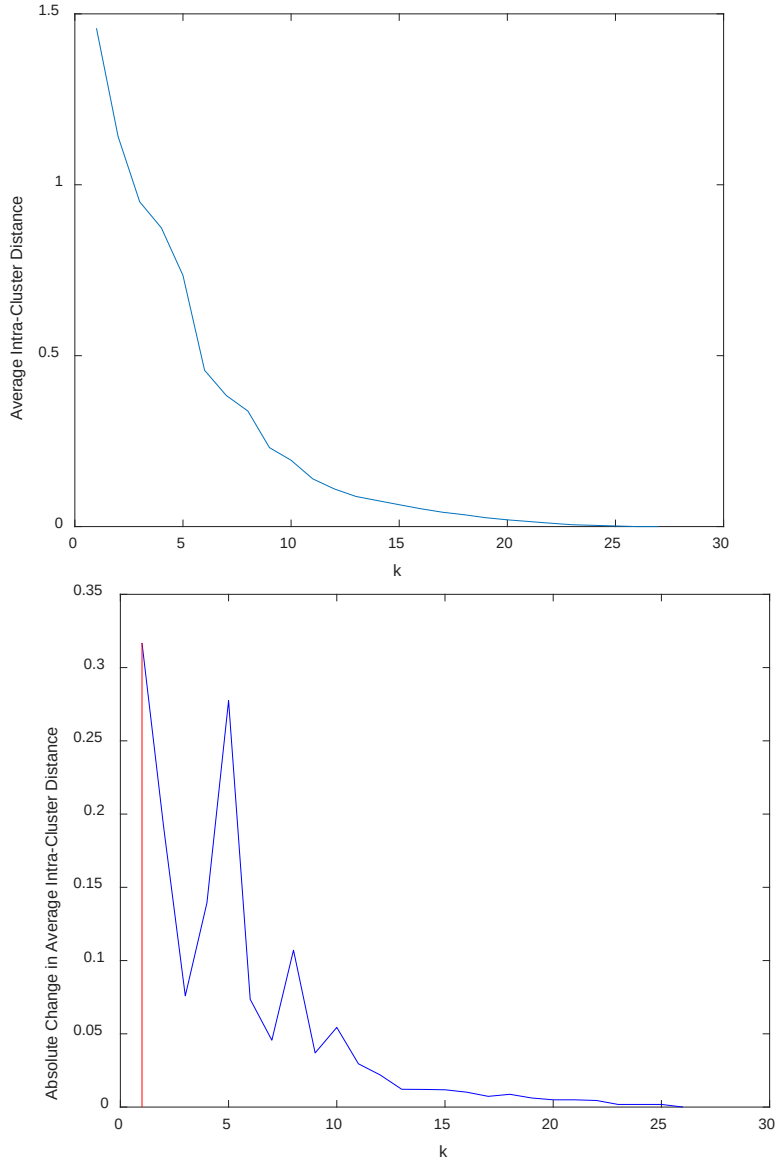
Graph Based Approaches

- Obviously just choosing the minimum of this isn't all that useful
  - It will always choose $k = N$
- How about the slope?
- The slope at location W_j is:

$$W_j' = \frac{(W_{j+1} - W_{j-1})}{2}$$

- Maybe select the place where there's the steepest absolute slope?

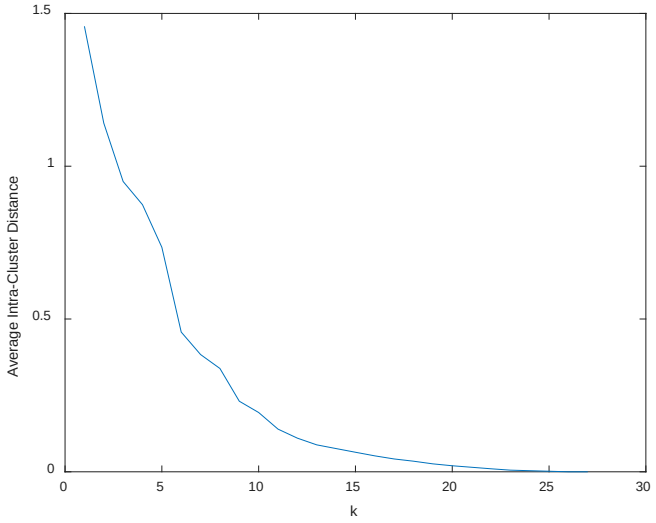
$$k = \underset{j}{\operatorname{argmax}}(|W_j'|)$$

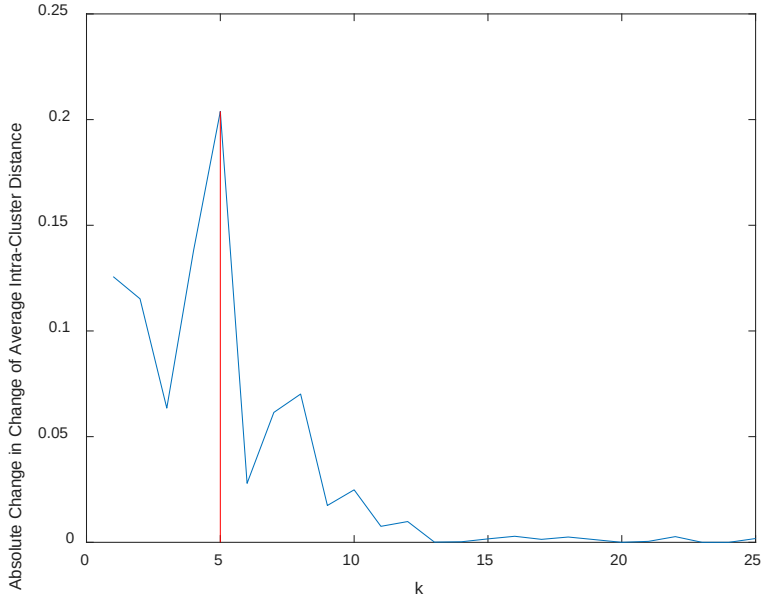


Graph Based Approaches

- How about the place where there's the greatest *change in slope*
  - Maximize the curvature
  - Second derivative!

$$W_j'' = \frac{(W'_{j+1} - W'_{j-1})}{2} = \frac{\left(\frac{(W_{j+2} - W_j)}{2} - \frac{(W_j - W_{j-2})}{2}\right)}{2} = \frac{(W_{j+2} - 2W_j + W_{j-2})}{4}$$





External Criteria for Clustering Quality

- Of course in the end we probably want to figure out how our algorithm is behaving.
- Maybe we can ask some people “after the fact” to do the clustering task and compare theirs to ours.
- Still unsupervised since the labels didn’t influence how we clustered. We just used it for evaluation

External Evaluation of Cluster Quality

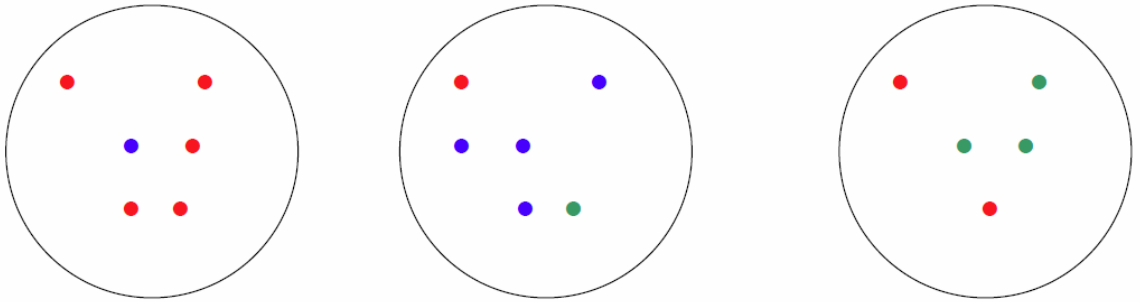
- Simple measure: **purity**
  - Let N_{ij} be the number of instances of (supervised) label *j* within cluster C_i
  - The purity of cluster C_i is then defined as

$$Purity(C_i) = \frac{1}{|C_i|} \max_j N_{ij}$$

  - Then we can define the average purity of this clustering as

$$Purity = \frac{1}{N} \sum_{i=1}^{k} |C_i| Purity(C_i)$$

Purity Example



Cluster I Cluster II Cluster III

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

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

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

$$\text{Total Purity} = \frac{1}{17}\left(6 * \frac{5}{6} + 6 * \frac{4}{6} + 5 * \frac{3}{5}\right) = \frac{12}{17} \approx 70\%$$

External Evaluation of Cluster Quality

- Purity is biased because having $k = N$ clusters maximizes purity
  - But it can be useful in compare methods with the same clustering level
- Other measurements include
  - Silhouette
    - https://en.wikipedia.org/wiki/Silhouette_(clustering)
  - V-Measure
    - https://www.aclweb.org/anthology/D07-1043