NYC DATA SCIENCE
ACADEMY

Cluster Analysis

Data Science Bootcamp

Outline

- Part 1: Cluster Analysis
- Part 2: K-Means Clustering
- Part 3: Hierarchical Clustering
- Part 4: Clustering Takeaways
- Part 5: Review

PART 1

Cluster Analysis

What Is Cluster Analysis?

- Up to this point, for the most part we've been concerned with building models that perform predictions:
  - Regression systems that attempt to predict a numeric output.
  - Classifiers that attempt to predict class membership.

- In contrast, cluster analysis is an unsupervised task that:
  - Does not aim to specifically predict a numeric output or class membership.
  - Does aim to uncover underlying structure of the data and see what patterns exist in the data.
    - We aim to group together observations that are similar while separating observations that are dissimilar.

What Is Cluster Analysis?

- Cluster analysis attempts to explore possible subpopulations that exist within your data.

- Typical questions that cluster analysis attempts to answer are:
  - Approximately how many subgroups exist in the data?
  - Approximately what are the sizes of the subgroups in the data?
  - What commonalities exist among members in similar subgroups?
  - Are there deeper subgroups that can further segment current subgroups?
  - Are there any outlying observations?

- Notice that these questions are largely exploratory in nature.

PART 2

K-Means Clustering

The K-Means Algorithm

- With the K-means clustering algorithm, we aim to split up our observations into a predetermined number of clusters.
  - You must specify the number of clusters K in advance.
  - These clusters will be distinct and non-overlapping.

- The points of each of the clusters are determined to be similar to a specific centroid value:
  - The centroid of a cluster represents the average observation of a given cluster; it is a single theoretical observation that represents the prototypical member that exists within the cluster.
  - Each observation will be assigned to exactly one of the K clusters depending on where the observation falls in space in respect to the cluster centroid locations.

The K-Means Algorithm: Mathematically

- Suppose C_1, C_2, …, C_K denote the various sets containing the indices of the observations in the respective clusters. Then, under the K-means clustering algorithm, the following must be true:

- $C_1 \cup C_2 \cup \ldots \cup C_\mathrm{K} = \{1, 2, \ldots, n\}$
    - ➢ In other words, each observation belongs to at least one of the K clusters.

- $C_\mathrm{k} \cap C_{\mathrm{k}'} = \varnothing$
    - ➢ In other words, the clusters are distinct and non-overlapping; there does not exist an observation that belongs to more than one cluster.

- It follows then that each observation must fall into exactly one cluster.

The K-Means Algorithm: Mathematically

- What makes a "good" clustering solution? Conceptually, we desire each point in a specific cluster to be near:
  - The centroid of that cluster.
  - All other points within the same cluster.

- Mathematically, this would mean that we desire the within-cluster variation to be as small as possible.

The K-Means Algorithm: Mathematically

- Suppose we define the concept of distance using Euclidean measurement. Then the within-cluster variation is defined as:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

- Here:
  - $|C_\text{k}|$ denotes the total number of observations in cluster k.
  - i and i' denote indices of observations in cluster C_k.
  - p is the number of variables/parameters in our dataset.

- In other words, the within-cluster variation for the k^th cluster is the sum of all of the pairwise squared Euclidean distances between the observations in the k^th cluster divided by the total number of observations in the k^th cluster.

The K-Means Algorithm: Mathematically

- Since the within-cluster variation is a measure of the amount by which the observations in a specific cluster differ from one another, we want to minimize this quantity W(C$_k$) over all clusters:

$$\min_{C_1,\ldots,C_K}\left\{\sum_{k=1}^{K} W(C_k)\right\}$$

- In other words, we desire to partition the observations into K clusters such that the total within-cluster variation added together across all K clusters is as small as possible; the optimization problem for K-means is as follows:

$$\min_{C_1,\ldots,C_K}\left\{\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2\right\}$$

The K-Means Algorithm: Procedurally

- Theoretically, it is quite computationally expensive to directly solve this minimization problem. Why?
  - If we were to check across all possible clustering assignments, we would have to calculate the within-cluster variations for K^n different solutions!

- Instead, the typical K-means algorithm follows the following procedure:
  a. Initialize by placing K centroids at random locations in the feature space.
  b. Assign each observation to the cluster whose centroid is closest by some distance measure (Euclidean).
  c. Recalculate the cluster centroids.
     i. The k^{th} cluster centroid is the vector of the p variable averages for all the observations in the k^{th} cluster. Return to part b.

- Halt when the cluster assignments no longer change.

Visualization

The K-Means Algorithm: Procedurally

- Why does the K-means algorithm end up necessarily reducing the within-cluster variances? Let's inspect the following identity:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

$$= 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

- We can rewrite the pairwise variation as the the variation around the component-wise means (centroids). During the algorithm, if we had just fixed the:
  - Centroids, then the observation reassignment step finds the closest centroid (and thus reduces the within-cluster variances).
  - Observation assignments, then the resulting sample cluster means minimize the sum of squared distances (and thus reduces the within-cluster variances).

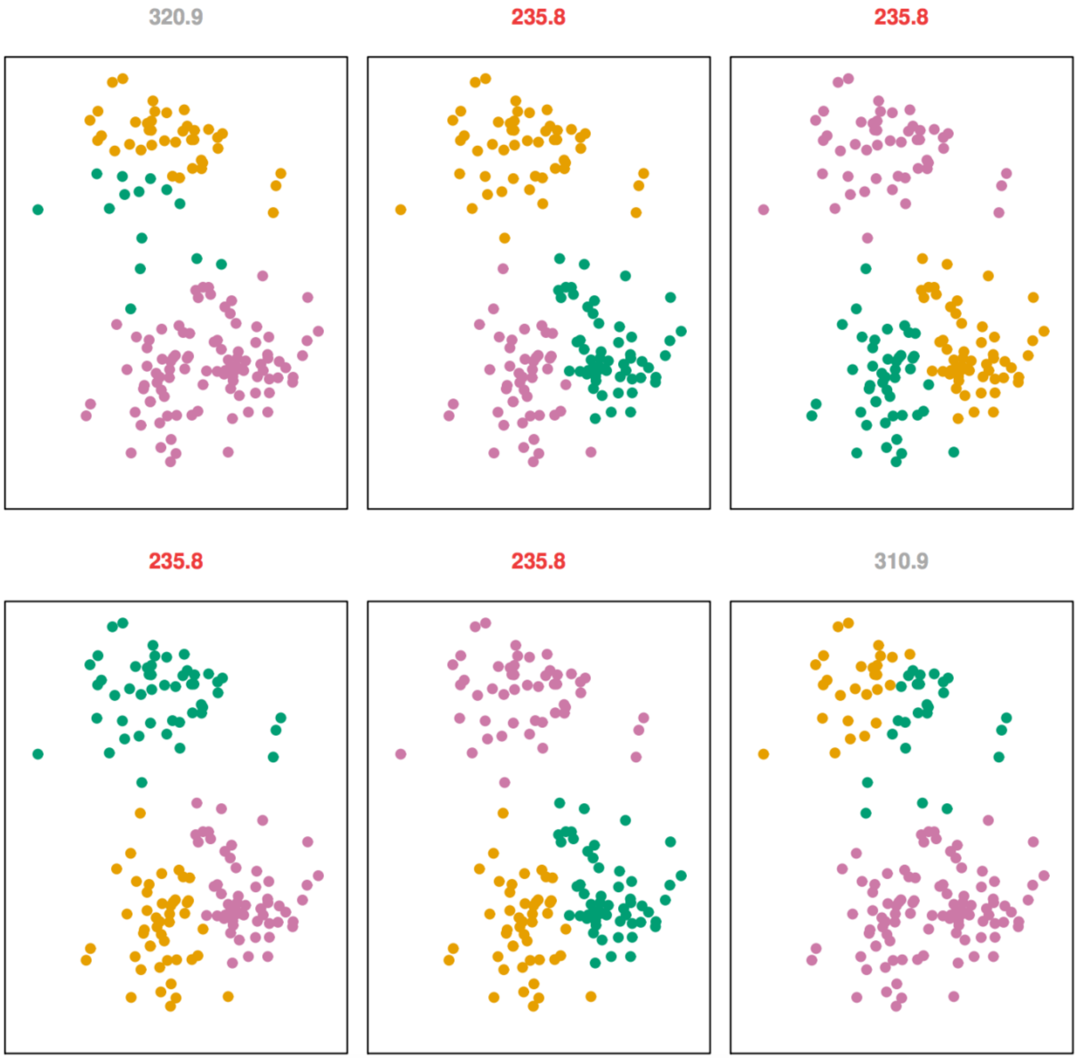
The K-Means Algorithm: Procedurally

- The K-means procedure always reaches convergence:
  - If you run the algorithm from a fixed beginning point, it will reach a stable endpoint where the clustering solution will no longer change.

- Unfortunately, the guaranteed convergence is to a local minimum.
  - Thus, if we begin the K-means algorithm with a different initial configuration, it is possible that convergence will find different centroids and therefore ultimately different cluster memberships.

- What do we do to get around this?
  - Run the K-means procedure several times and pick the clustering solution that yields the smallest aggregate within-cluster variance.

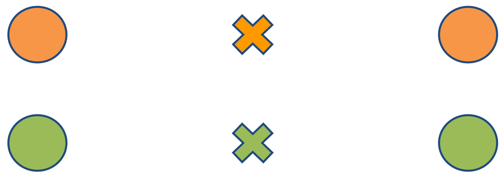
The K-Means Algorithm: Procedurally



Image

The K-Means Algorithm: Drawbacks

- Points that are nearby each other (have a small Euclidean distance between them) are not guaranteed to be clustered together.
  - It could be the case that a stable solution produces clusters that don't necessarily cluster the closest points together:



- K-means assumes that true clusters have a globular shape (i.e., a spherical shape that has a well-defined center).
  - When the data has non-globular or chain-like shapes, K-means may not perform well.

How to Choose K?

- The biggest question we encounter with K-means is the determination of the appropriate number of clusters.
  - The K-means algorithm does not answer this question; instead, it expects that we know the answer prior to running the algorithm in the first place!

- Recall that K-means is attempting to reduce the within-cluster variance. What if we could check a lot of values for K, record the overall within-cluster variation, and just use the value of K that yields the lowest variance in the data? Why will this not work?
  - As K increases, the overall within-cluster variance will continue to decrease. In general, the more centroids you have in a space, the closer all points will be to one of those centroids.
  - Envision the scenario where every data point is its own centroid. What is the overall within-cluster variance?

How to Choose K?

- In practice, this decision comes to us by visual inspection of a scree plot (also known as an "elbow graph") of the data.

- We plot the within-cluster variance as a function of the number of clusters to create a segmented curve.
  - We know the within-cluster variance will necessarily decrease as we increase the number of clusters, but it won't decrease uniformly.
  - The within-cluster variance will tend to decrease quickly at first, but then begin to taper off.
  - The task reduces to simply finding the point where the within-cluster variance no longer decreases dramatically.

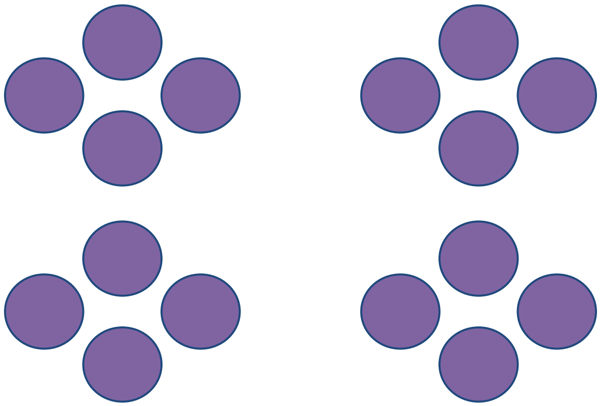
PART 3

Hierarchical Clustering

Granularity in Clustering

- The biggest disadvantage of K-means is that we must pre-specify the number of clusters. But selecting K often poses the problem of perceived granularity.

- To better understand this idea, try answering the question: How many clusters are in the following dataset?
  - Are there 2?
  - Are there 4?
  - Are there 16?



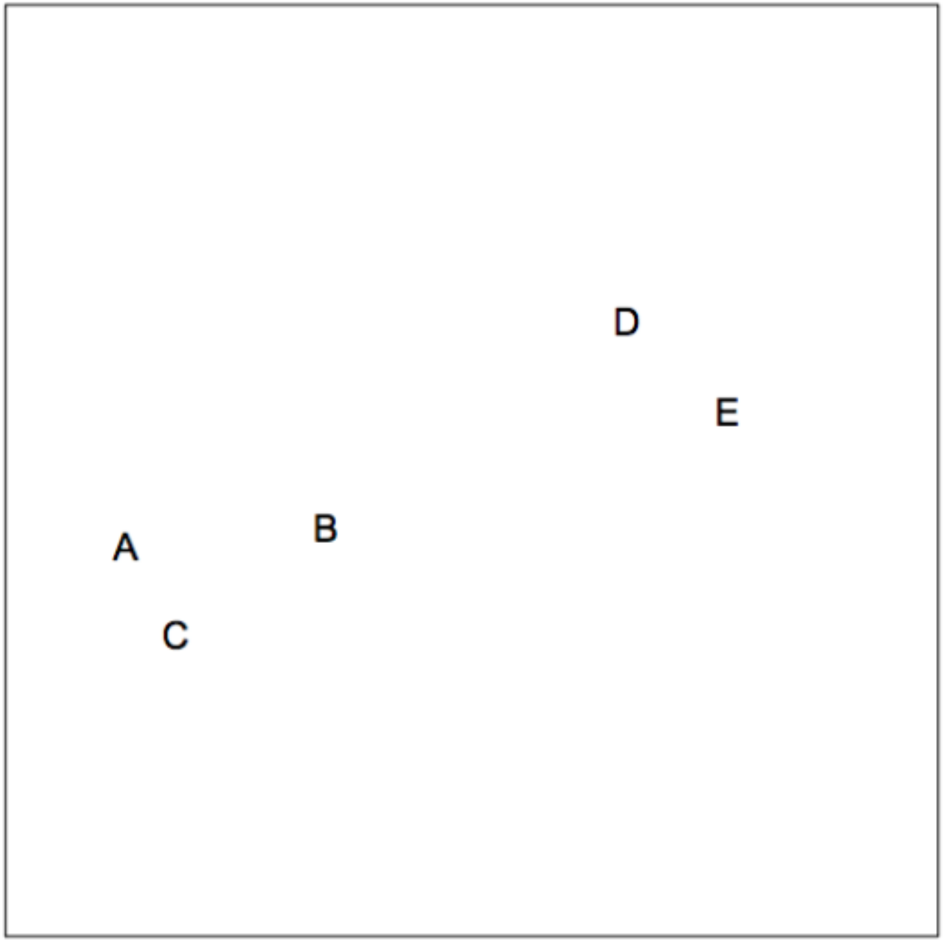
Agglomerative Clustering

- One possible approach to get around the granularity problem is to approach it head-on by means of agglomerative clustering.

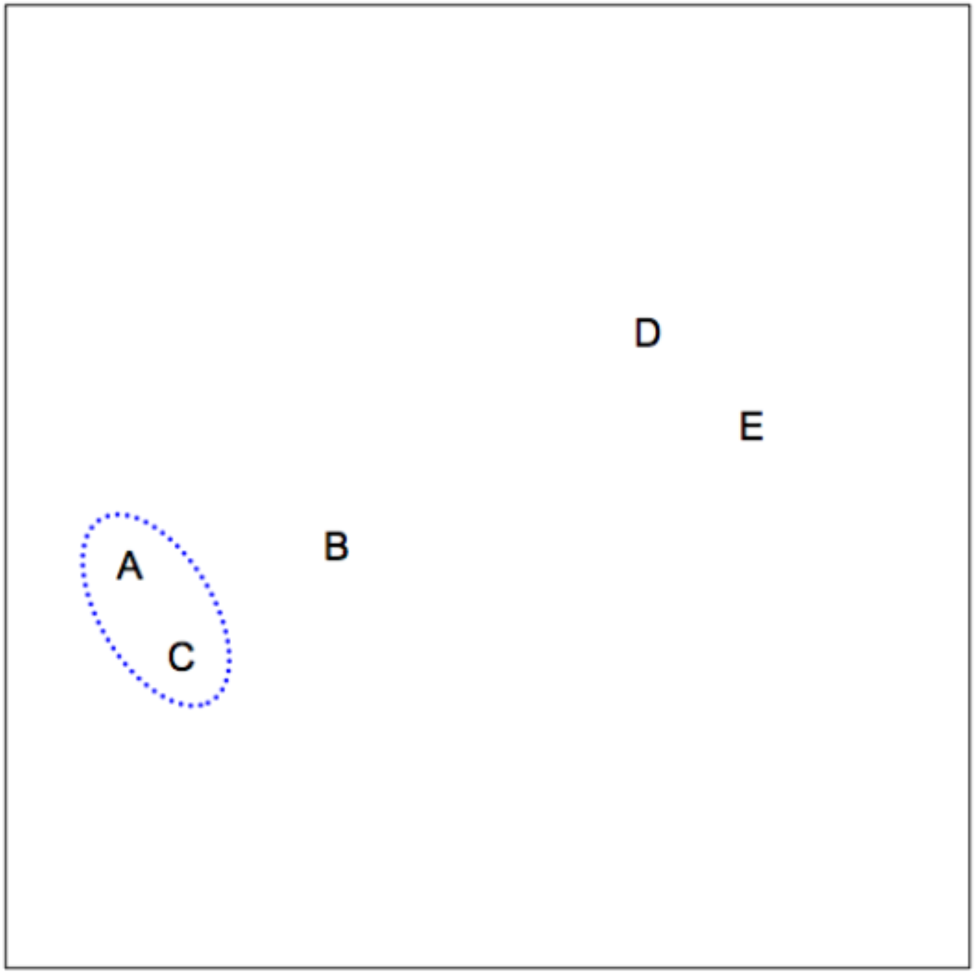
- Instead of picking a fixed number of clusters, we instead build a hierarchy of clustering structures. Envision building a tree:
  - At the bottom level, the extreme case would be each observation is partitioned into its own cluster (as if $K = n$).
  - At each intermediary level, we can recursively define the closest two clusters and fuse them together.
  - At the top level, the extreme case would be each observation is partitioned into the exact same cluster (as if $K = 1$).

- In hierarchical clustering, a visualization of this hierarchical tree is called a dendrogram.

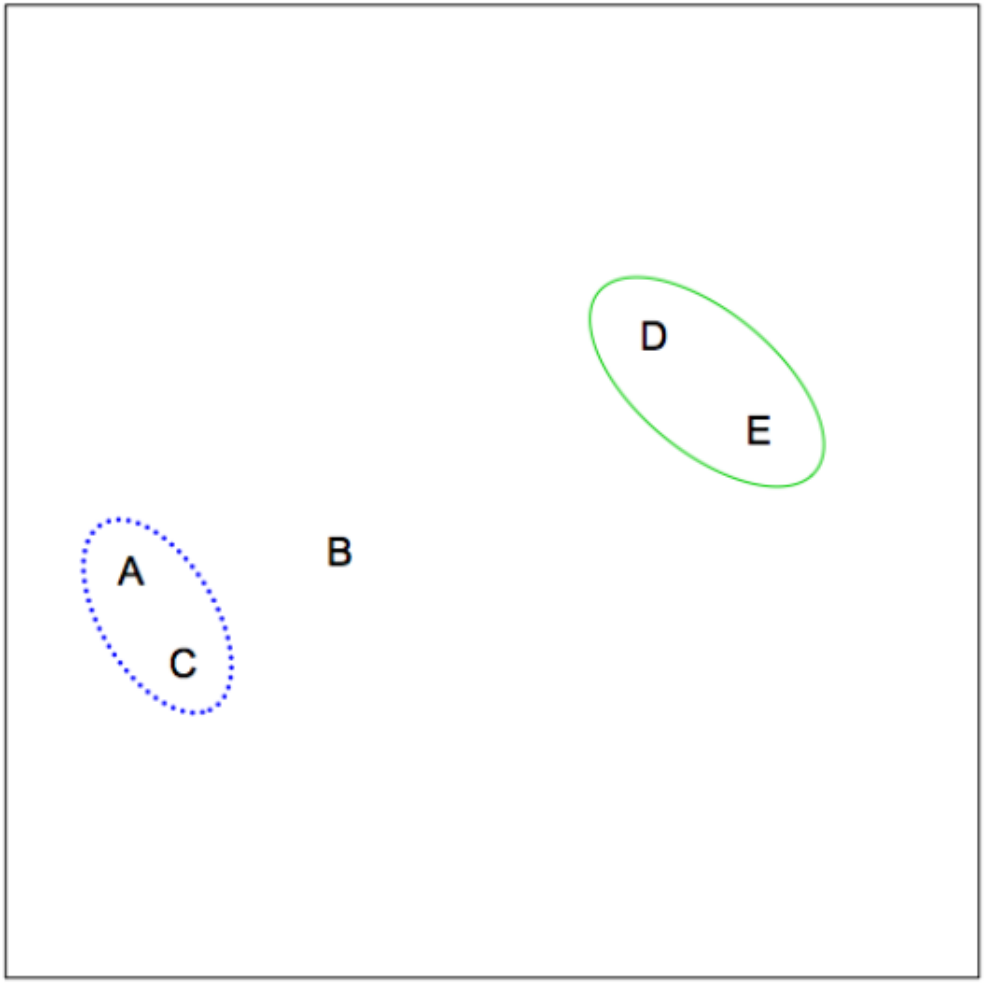
Hierarchical Clustering: Visually



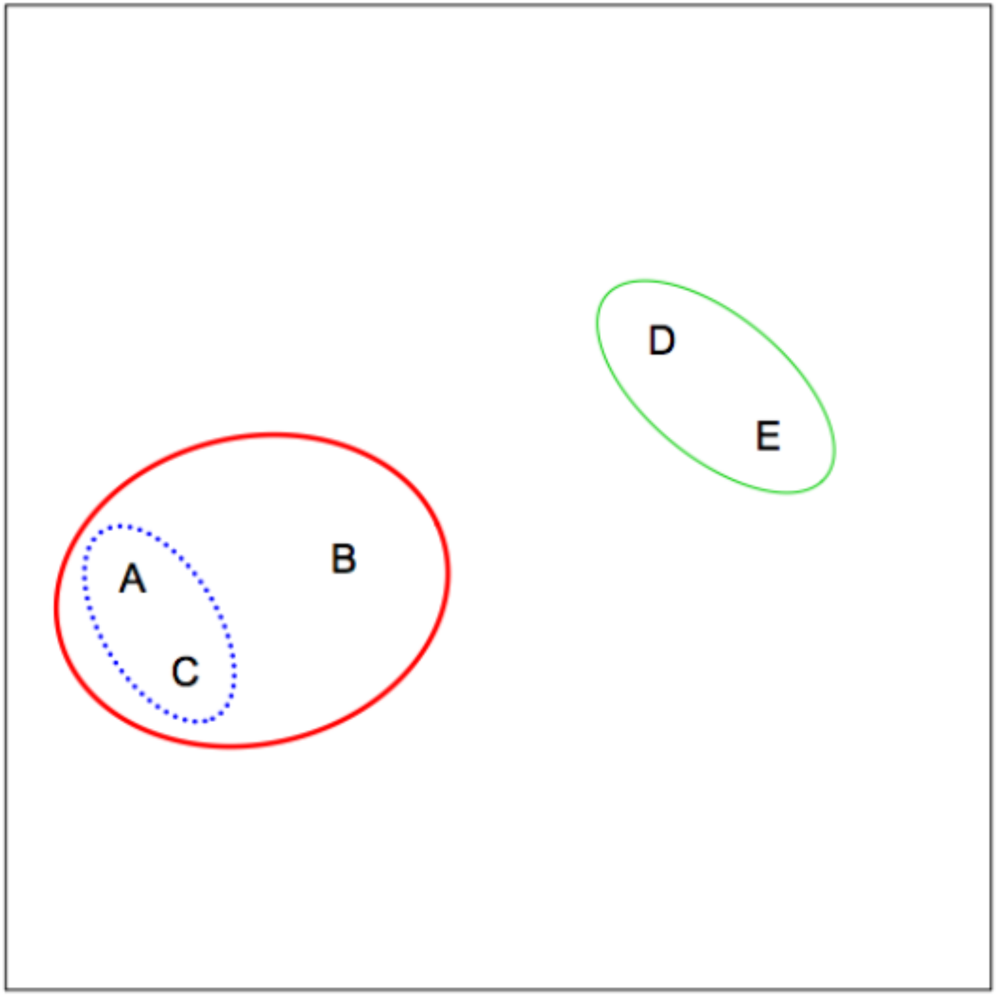
Hierarchical Clustering: Visually



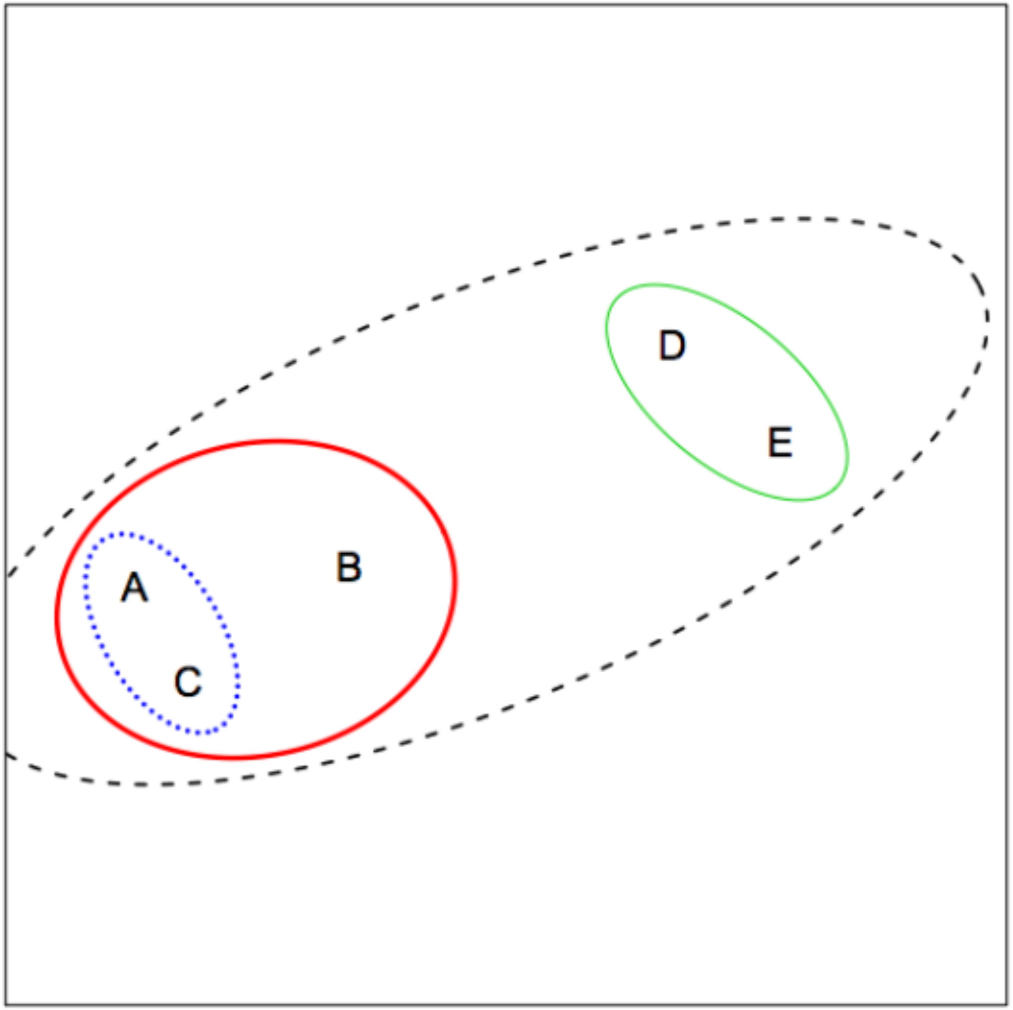
Hierarchical Clustering: Visually



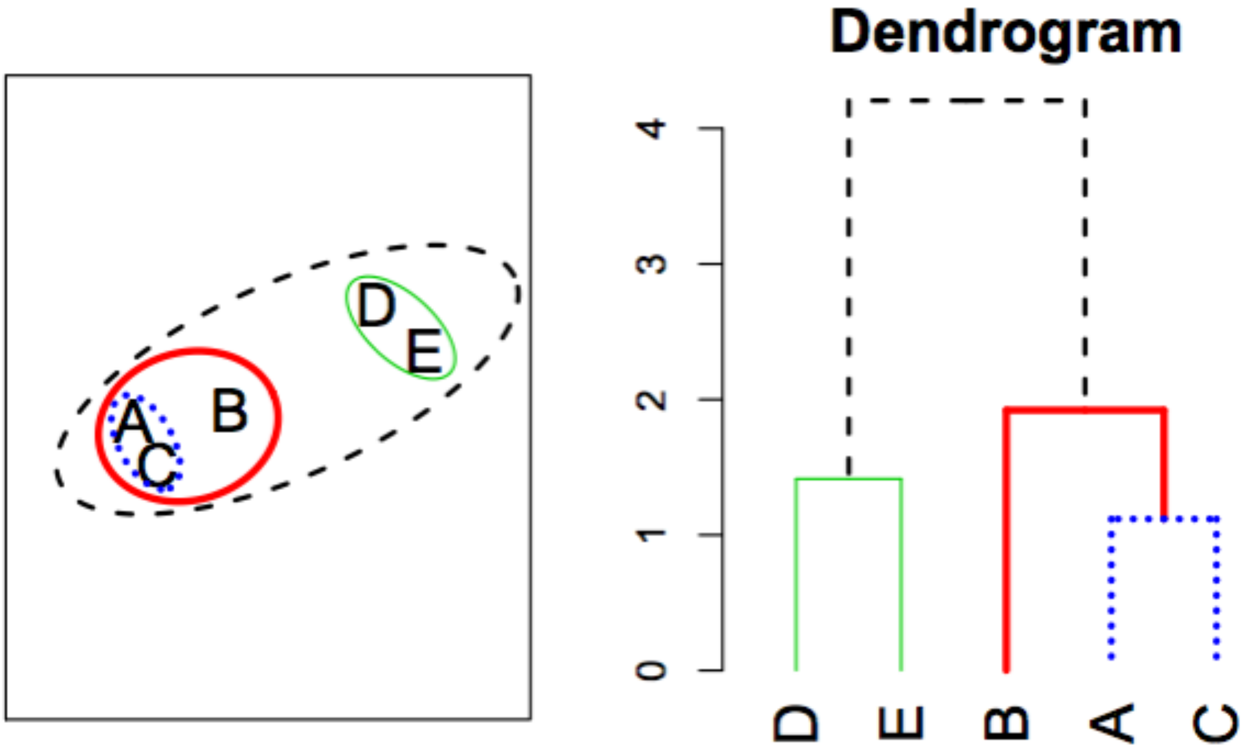
Hierarchical Clustering: Visually



Hierarchical Clustering: Visually



Hierarchical Clustering: Visually



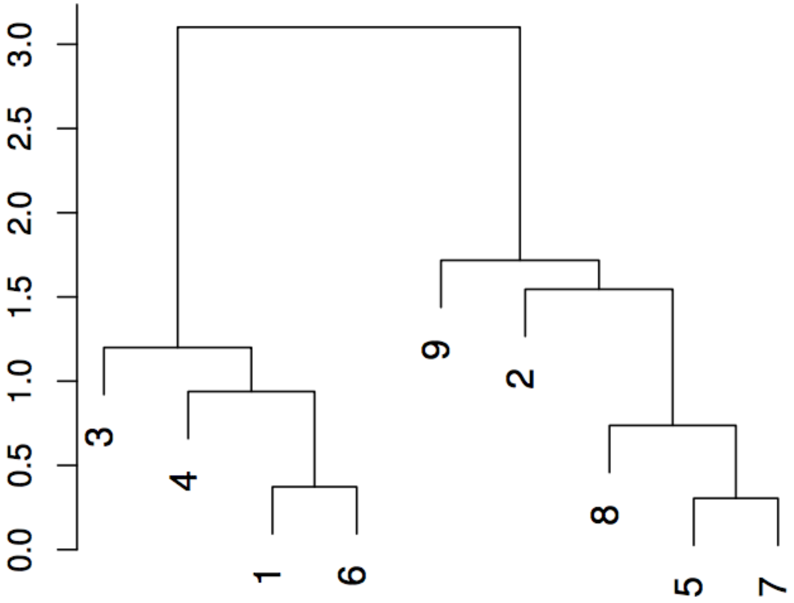
Dendrogram

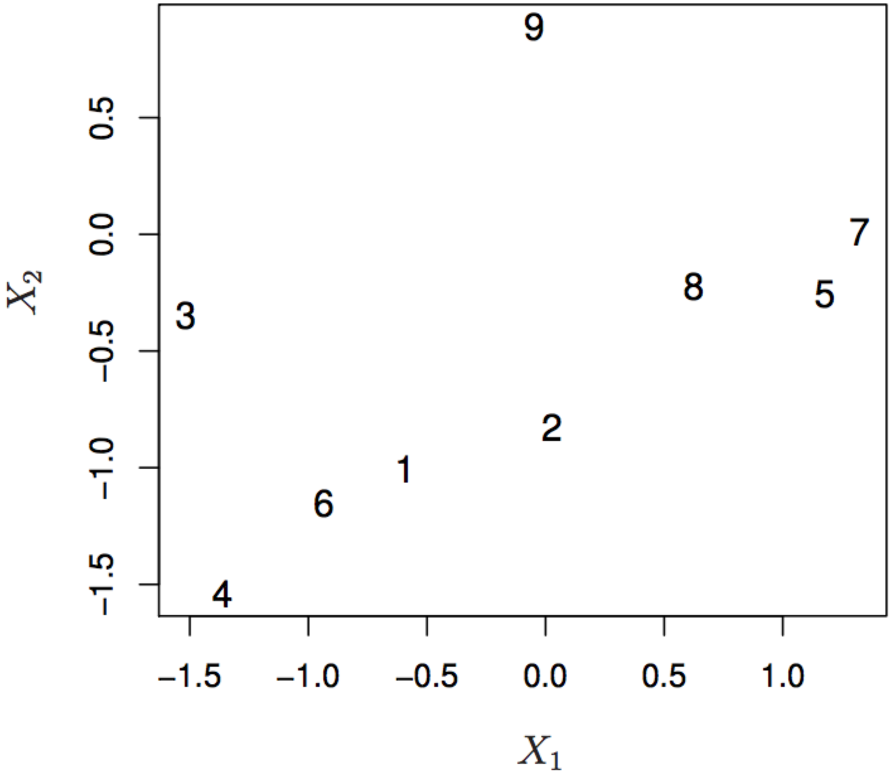
Interpreting the Dendrogram

- There are some interpretative advantages to visualizing the dendrogram created from hierarchical clustering:
  - The lower down in the dendrogram a fusion occurs, the more similar the groups of observations that have been fused are to each other.
  - The higher up in the dendrogram a fusion occurs, the more dissimilar the group of observations that have been fused are to each other.

- In general, for any two observations we can inspect the dendrogram and find the point at which the groups that contain those two observations are fused together to get an idea of their dissimilarity.
  - Be careful to consider groups of points in the fusions within the dendrograms, not just individual points.

Interpreting the Dendrogram: Visually





Image

The Hierarchical Clustering Algorithm

1. Begin with n observations and a distance measure of all pairwise dissimilarities. At this step, treat each of the n observations as their own clusters.

1. For i = n, (n - 1), …, 2:
   a. Evaluate all pairwise inter-cluster dissimilarities among the i clusters and fuse together the pair of clusters that are the least dissimilar.
   b. Note the dissimilarity between the recently fused cluster pair and mark that as the associated height in the dendrogram.
   c. Repeat the process by now calculating the new pairwise inter-cluster dissimilarities among the remaining (i - 1) clusters.

The Hierarchical Clustering Algorithm

- While we do not need to specify K, in order to perform hierarchical clustering there are a few choices we need to make. Particularly:
  - A dissimilarity measure.
  - A linkage method.

- We're already familiar with the idea of choosing a dissimilarity measure with the choice of distance metric. In most cases, it is sufficient to use the Euclidean distance.

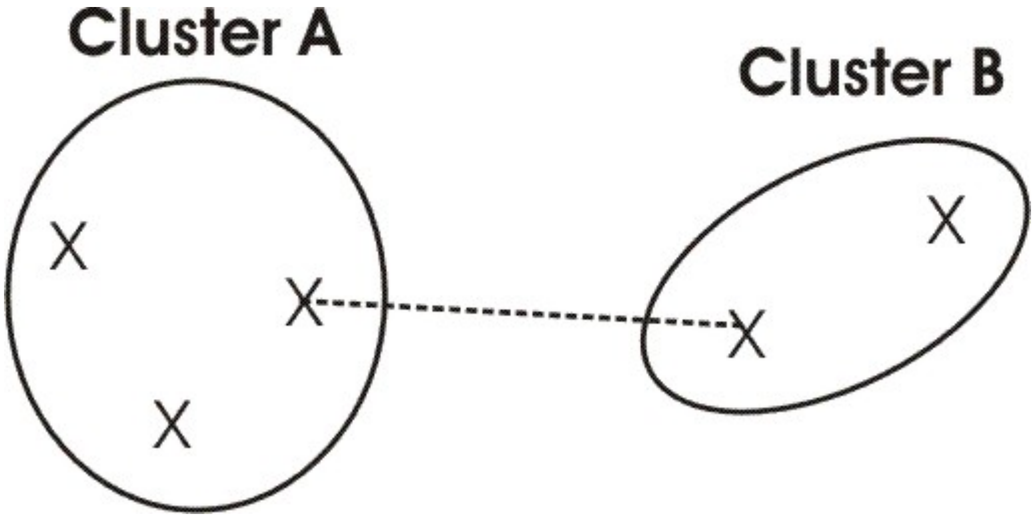
- What we have not yet addressed is the linkage method. We know how to define how two points are similar to one another, but what do we do when we want to assess the similarity among groups of points?

The Hierarchical Clustering Algorithm: Linkage

- The most common types of linkage are described below.

- First, compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B. Then:
  - Complete Linkage: Maximal inter-cluster dissimilarity.
    - Record the largest of the dissimilarities listed between A and B as the overall inter-cluster dissimilarity.
  - Single Linkage: Minimal inter-cluster dissimilarity.
    - Record the smallest of the dissimilarities listed between A and B as the overall inter-cluster dissimilarity.
  - Average Linkage: Mean inter-cluster dissimilarity.
    - Record the average of the dissimilarities listed between A and B as the overall inter-cluster dissimilarity.

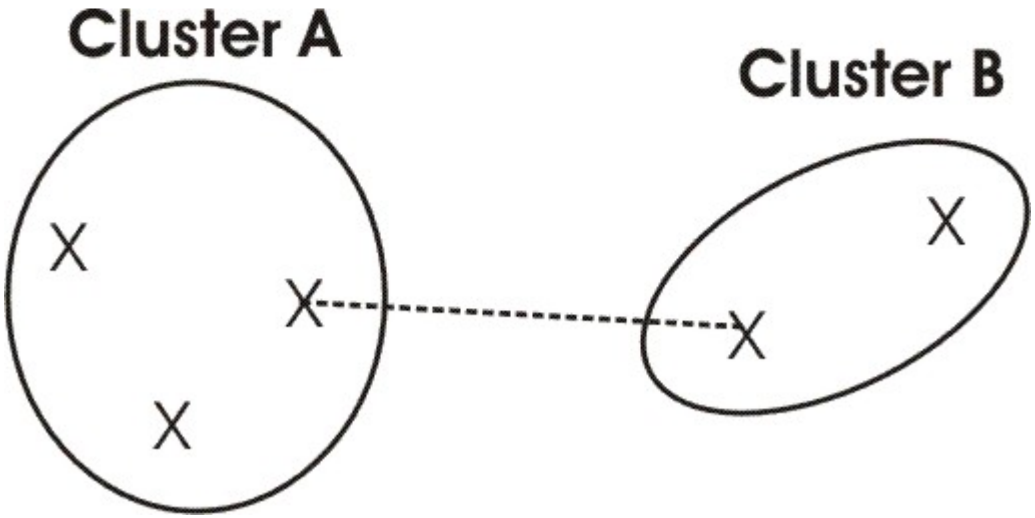
The Hierarchical Clustering Algorithm: Linkage



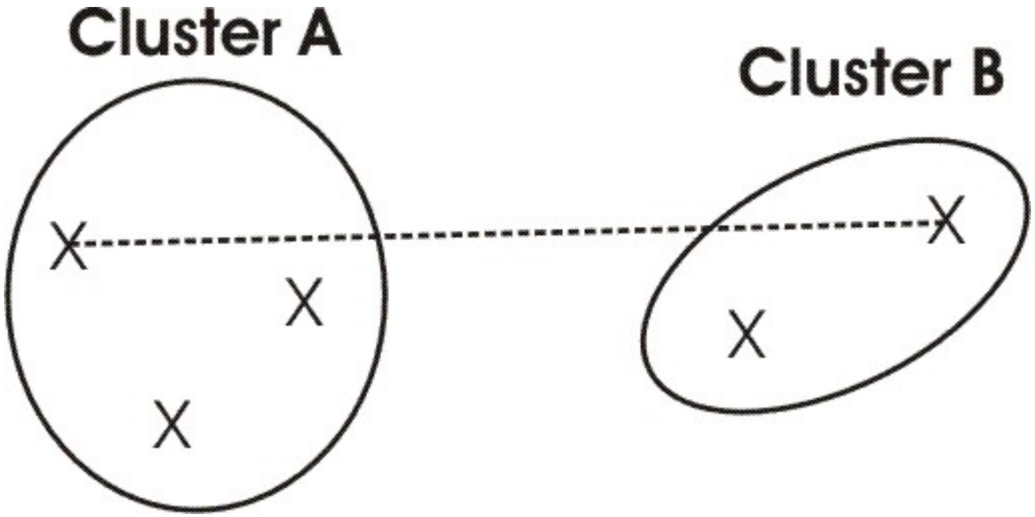
Image

The Hierarchical Clustering Algorithm: Linkage

Single Linkage



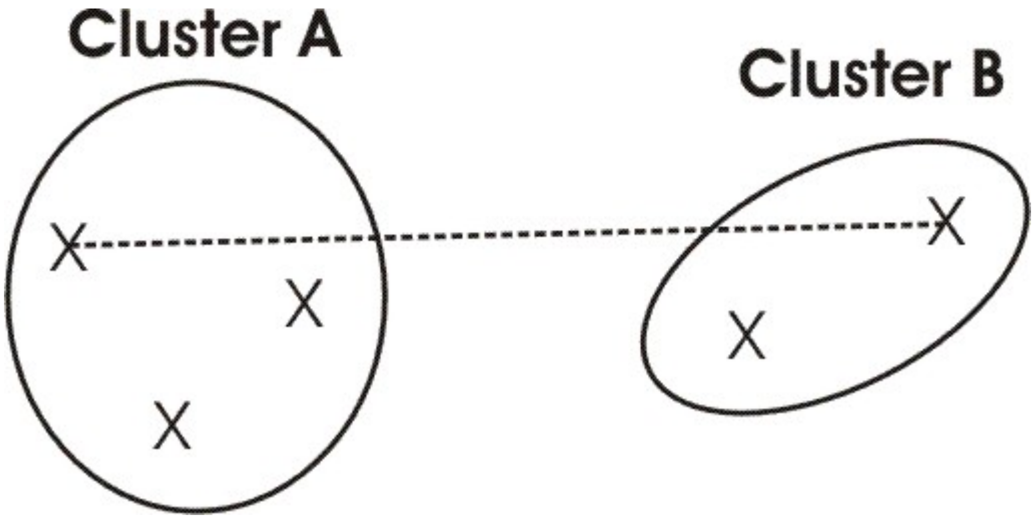
The Hierarchical Clustering Algorithm: Linkage



Image

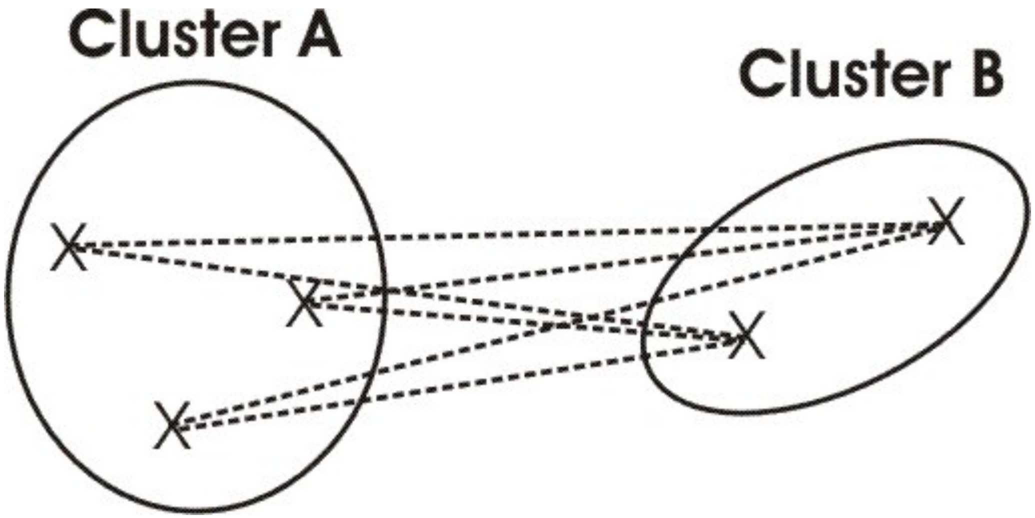
The Hierarchical Clustering Algorithm: Linkage

Complete Linkage



Image

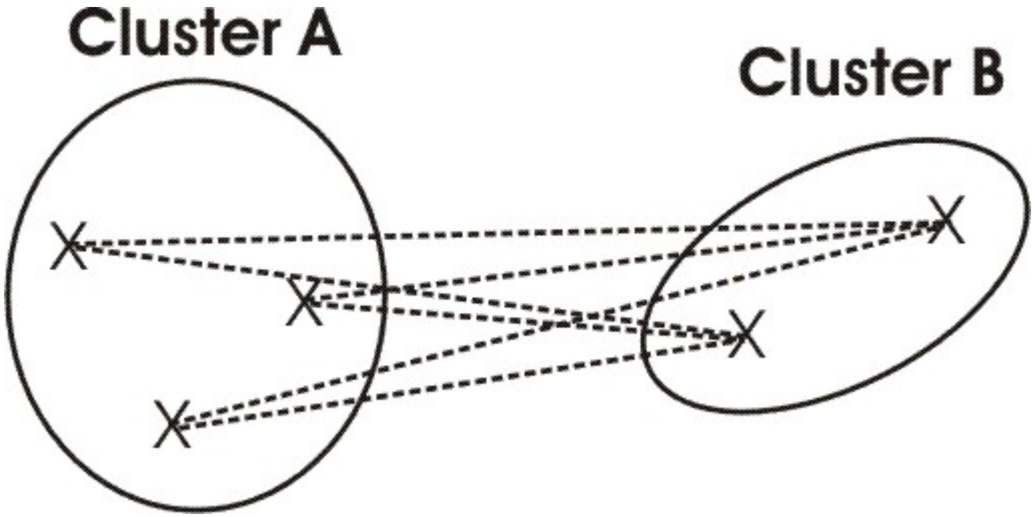
The Hierarchical Clustering Algorithm: Linkage



Image

The Hierarchical Clustering Algorithm: Linkage

Average Linkage



Image

The Hierarchical Clustering Algorithm: Linkage

- Complete linkage is sensitive to outliers, yet tends to identify clusters that are compact, somewhat spherical objects with relatively equivalent diameters.

- Single linkage is not as sensitive to outliers, yet tends to identify clusters that have a chaining effect; these clusters can often not represent intuitive groups among our data, and many pairs of observations might be quite distant from one another.

- Average linkage tends to strike a balance between the pros and cons of both complete linkage and single linkage.

PART 4

Clustering Takeaways

Things to Remember

- Clustering is an unsupervised method in machine learning; the main goal is to uncover structure among subsets of the data.
  - The clustering procedure is generally used for more for data exploration; we’re not predicting any outcomes.

- In good clustering solutions, points in the same cluster should be more similar to each other than to points in other clusters.

- The units by which each variable is measured matters; different unit measurements cause different distance calculations and thus change clustering solutions.
  - Usually we desire a unit change in one dimension to correspond to the same unit change in another dimension; in that perspective, we should standardize our data prior to clustering.

Things to Remember

- The process of clustering is more iterative and interactive; there's no one correct way to cluster your data.

- Supervised methods generally have one solution to the optimization problems posed, whereas some clustering methods (e.g., K-means) aren't deterministic.

- Different clustering methods yield different results (e.g., hierarchical clustering with varied linkage methodologies). Consider the output of different approaches.

PART 5

Review

Review

- Part 1: Cluster Analysis
  - What is Cluster Analysis?
- Part 2: K-Means Clustering
  - The K-Means Algorithm
    - Mathematically
    - Procedurally
    - Visually
    - Drawbacks
  - How to Choose K?
- Part 3: Hierarchical Clustering
  - Granularity in Clustering
  - Agglomerative Clustering
  - Hierarchical Clustering
    - Visually
  - Interpreting the Dendrogram
    - Visually
  - The Hierarchical Clustering Algorithm
    - Complete Linkage
    - Single Linkage
    - Average Linkage
- Part 4: Clustering Takeaways
  - Things to Remember