GENERAL ASSEMBLY

HIERARCHICAL CLUSTERING

Matt Speck, Data Science Immersive (h/t Joseph Nelson)

AGENDA

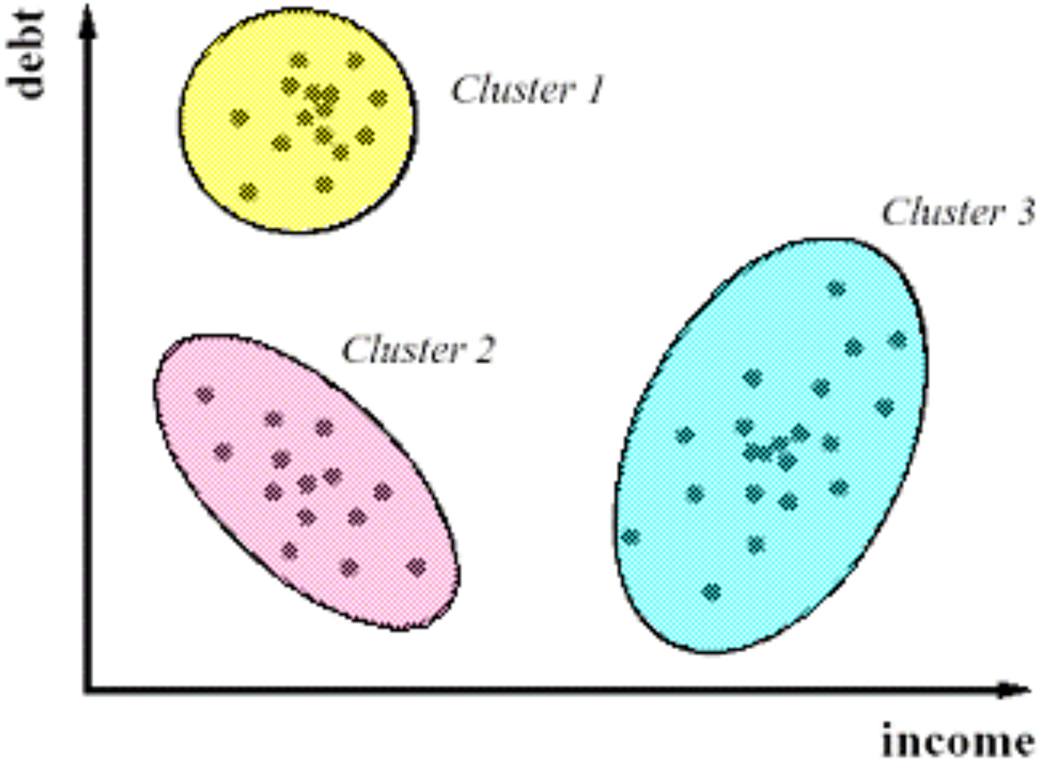
- What is hierarchical clustering?
- How does hierarchical clustering work?
- Code Implementation

WHAT IS HIERARCHICAL CLUSTERING?

- Review: what is clustering?

WHAT IS CLUSTERING?

- Review: what is clustering?
- Clustering is (generally) an unsupervised learning technique we employ to group “similar” data points together
- With unsupervised learning, remember: there is no **clear** objective, there is no “right answer” (hard to tell how we’re doing), there is no response variable, just observations with features, and labeled data is not required

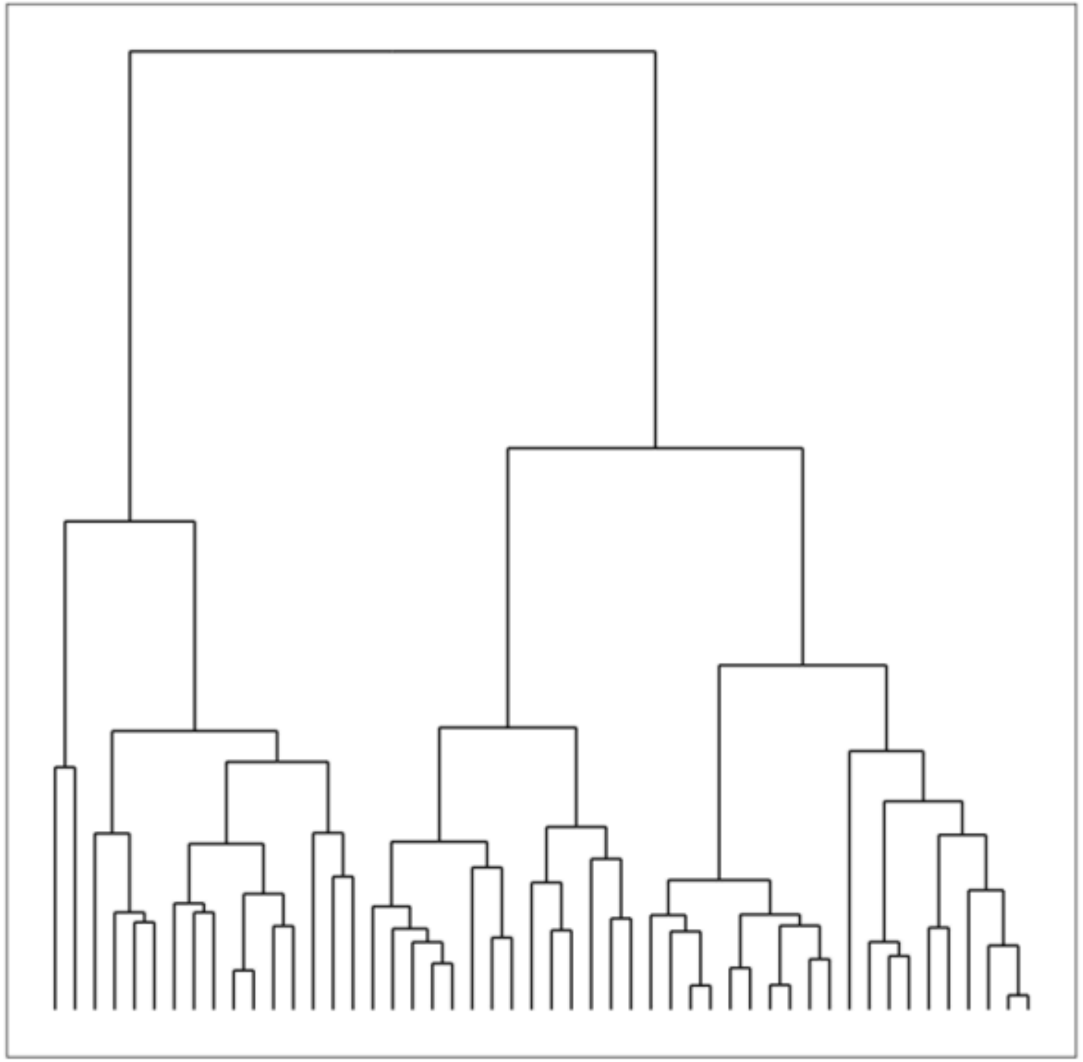


WHAT IS HIERARCHICAL CLUSTERING?

- So now, what is a hierarchy?

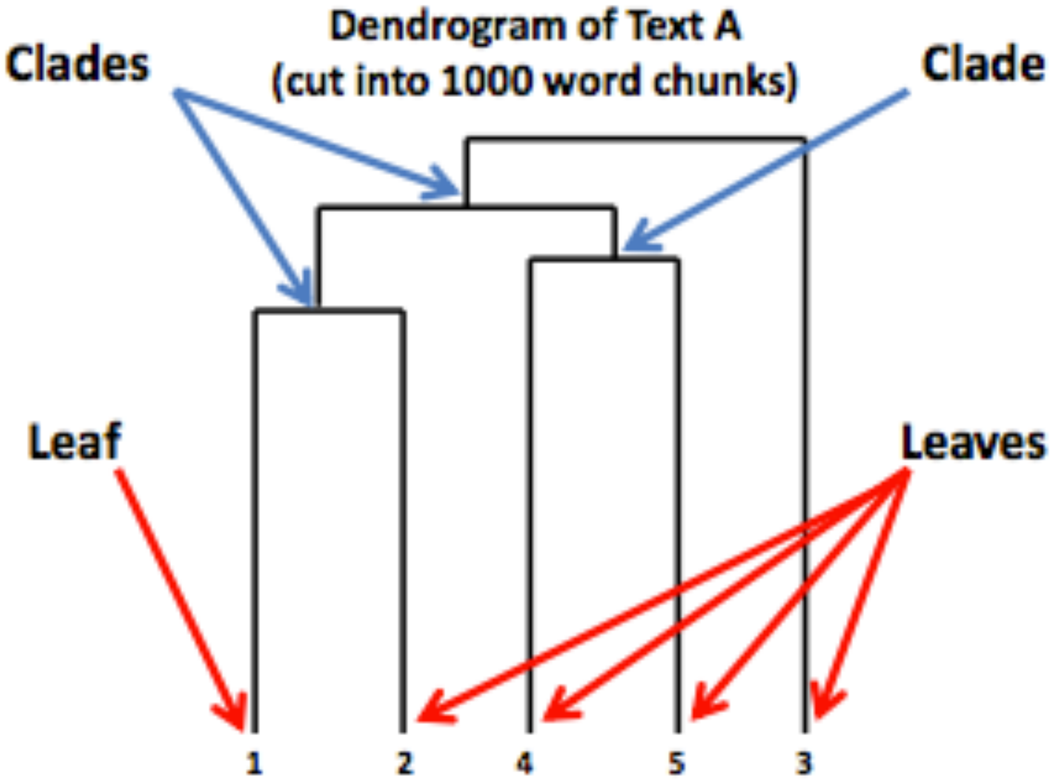
WHAT IS HIERARCHICAL CLUSTERING?

- Hierarchical clustering, like k-means clustering, is another common form of clustering analysis. With this type of clustering - we seek to do exactly what the name suggests: build hierarchies of links that ultimately form clusters.
- Once these links are determined, they are displayed in what is called a **dendrogram** - a graph that displays all of these links in a hierarchical manner.



WHAT IS HIERARCHICAL CLUSTERING?

- A **dendrogram** is a branching diagram that represents the relationships of similarity among a group of entities
- The arrangement of the clades tells us which leaves are most similar to each other. The height of the branch points indicates how similar or different they are from each other: **the greater the height, the greater the difference**

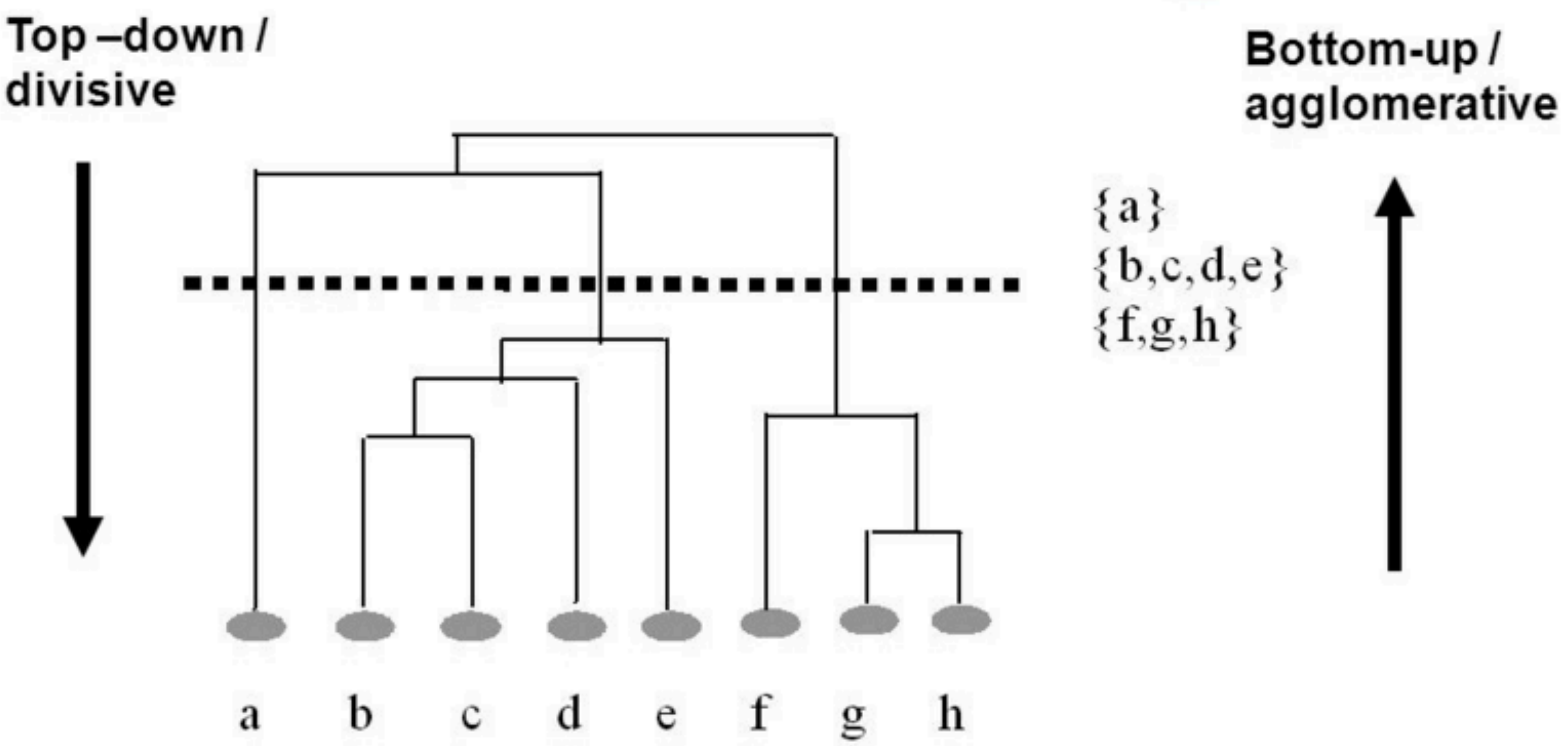


HOW DOES HIERARCHICAL CLUSTERING WORK?

- In hierarchical clustering, instead of clustering in one step, the clusters are determined in a varying number of partitions. At each step, it makes the best choice based on the surrounding data points, with the ultimate goal that these best choices will lead to the best choice of clusters overall. Because hierarchical clusters make the best choice *at each step*, we say that hierarchical clustering is a **greedy algorithm.**

HOW DOES HIERARCHICAL CLUSTERING WORK?

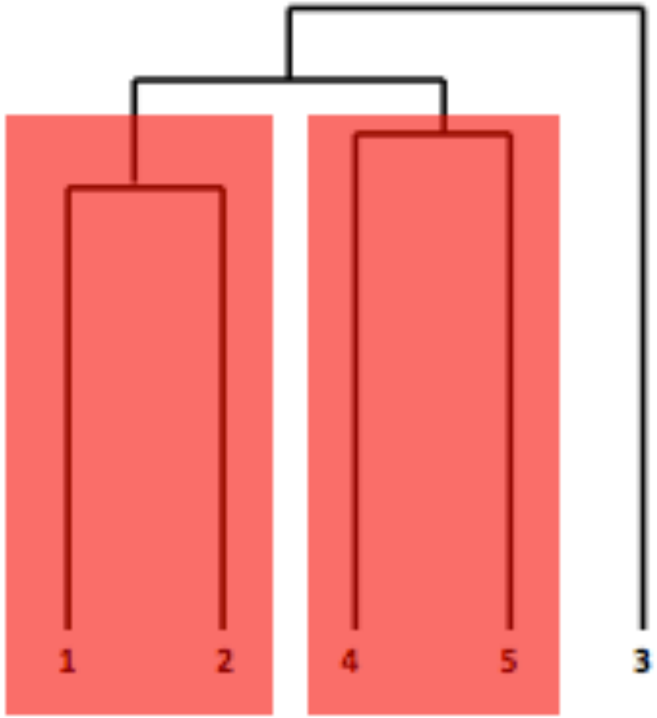
- There are two forms of hierarchical clustering; **agglomerative hierarchical** clustering and **divisive hierarchica**l clustering.



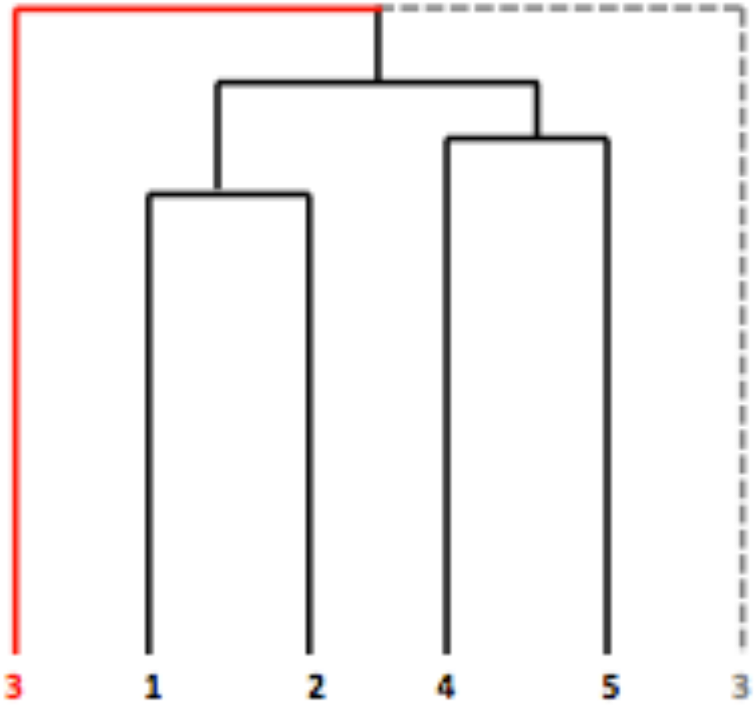
HOW DOES HIERARCHICAL CLUSTERING WORK?

- There are two forms of hierarchical clustering; **agglomerative hierarchical** clustering and **divisive hierarchica**l clustering.

Cluster from Bottom Up



Horizontal Reading



HOW DOES HIERARCHICAL CLUSTERING WORK?

- To think about the difference between agglomerative vs divisive, with the former we start with the leaves of the tree and build the trunk, and with the latter we start with the trunk of the tree and build the leaves. Both methods are applicable when using hierarchical clustering, but agglomerative (bottom up) is more well-known.

- Today we're going to look at agglomerative hierarchical clustering, also known as **linkage clustering**. Linkage clustering iterates through data points and computes the dissimilarity between groups by computing the distances between points within those two groups. It then combines two clusters into one, based on which ever pair of clusters has the **lowest** dissimilarity (ie. whichever two clusters are most similar)

LINKAGE

- We use linkage to compute distance between two clusters, but there are actually different linkages and, therefore, different ways of computing distance. How we decide to compute distance can affect which clusters form.

LINKAGE:

- Linkage name: **Single Linkage**
- In single linkage (aka nearest-neighbor linkage), the dissimilarity between two clusters G and H is calculated as the minimum distance (or dissimilarity) between any two points in the opposite groups

$$d_{\text{single}}(G, H) = \min_{i \in G,\, j \in H} d_{ij}$$

LINKAGE:

- Linkage name: **Complete linkage**
- In complete linkage (aka furthest-neighbor linkage), the dissimilarity between two clusters G and H is calculated as the maximum distance (or dissimilarity) between any two points in the opposite groups

$$d_{\text{single}}(G,H) = \min_{i \in G,\, j \in H} d_{ij}$$

LINKAGE:

- Linkage name: **Average linkage**
- In average linkage, the dissimilarity between two clusters G and H is calculated as the mean distance (or dissimilarity) between over all points in opposite groups.

$$d_{\mathsf{single}}(G, H) = \min_{i \in G,\, j \in H} d_{ij}$$

LINKAGE SUMMARY

- We learned about three types of linkage (ways of calculating dissimilarity):
- **Single –** minimal inter-cluster dissimilarity
- **Complete –** maximal inter-cluster dissimilarity
- **Average –** mean inter-cluster dissimilarity

- There are other types too:
- Centroid – dissimilarity between centroids of clusters
- Ward’s method – minimizes variance when forming clusters
- More…

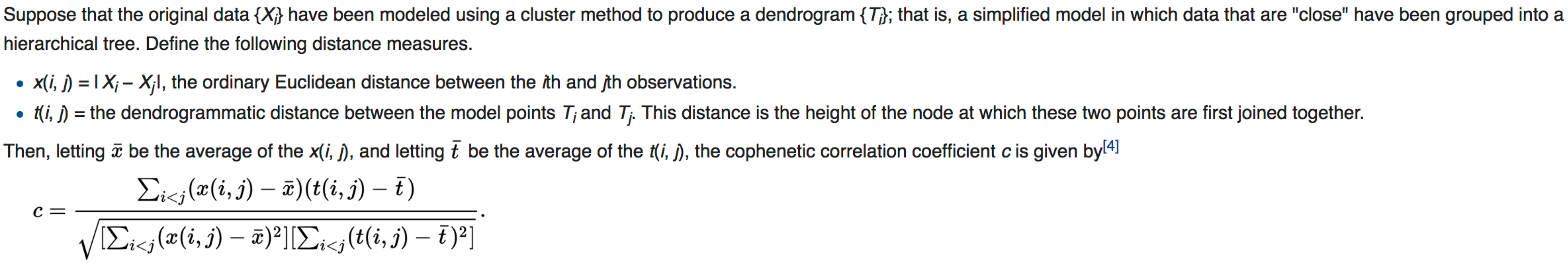
IMPLEMENTATION

- Implementing hierarchical clustering in python is as simple as calling a function from the SciPy toolbox:
- `Z = linkage(X, ’single')`
- Here, "X" represents the matrix of data that we are clustering, and “single” tells our algorithm which method to use to calculate distance between our newly formed clusters - in this case **single,** which we talked about earlier. When calculating distance, the default is **Euclidean distance,** which is what we’ve been using so far.

IMPLEMENTATION

- After we cluster, we can calculate the dendrogram using a simple dendrogram() function from SciPy, which we can then draw using our handy plt from matplotlib.
- To check how well our algorithm has measured distance, we can calculate the **cophenetic correlation coefficient.** This metric, which measures the height of the dendrogram at the point where two branches merge, can tell us how well the dendrogram has measured the distance between data points in the original dataset and is a helpful measure to see how well our clustering test has run.
- Basically, it’s the correlation between two data points’ actual distances and the distance at which they joined the same cluster.
- `c, coph_dists = cophenet(Z, pdist(X))`

COPHENETIC CORRELATION COEFFICIENT



Suppose that the original data $\{X_i\}$ have been modeled using a cluster method to produce a dendrogram $\{T_i\}$; that is, a simplified model in which data that are "close" have been grouped into a hierarchical tree. Define the following distance measures.

- $x(i, j) = |X_i - X_j|$, the ordinary Euclidean distance between the *i*th and *j*th observations.
- $t(i, j)$ = the dendrogrammatic distance between the model points T_i and T_j. This distance is the height of the node at which these two points are first joined together.

Then, letting $\bar{x}$ be the average of the $x(i, j)$, and letting $\bar{t}$ be the average of the $t(i, j)$, the cophenetic correlation coefficient c is given by[4]

$$c = \frac{\sum_{i<j}(x(i,j) - \bar{x})(t(i,j) - \bar{t})}{\sqrt{[\sum_{i<j}(x(i,j) - \bar{x})^2][\sum_{i<j}(t(i,j) - \bar{t})^2]}}.$$

- Anyway…

REFERENCES:

- http://www.stat.cmu.edu/~ryantibs/datamining/lectures/05-clus2-marked.pdf
- https://en.wikipedia.org/wiki/Cophenetic_correlation#Calculating_the_cophenetic_correlation_coefficient

CODING IMPLEMENTATION

- To the repo…