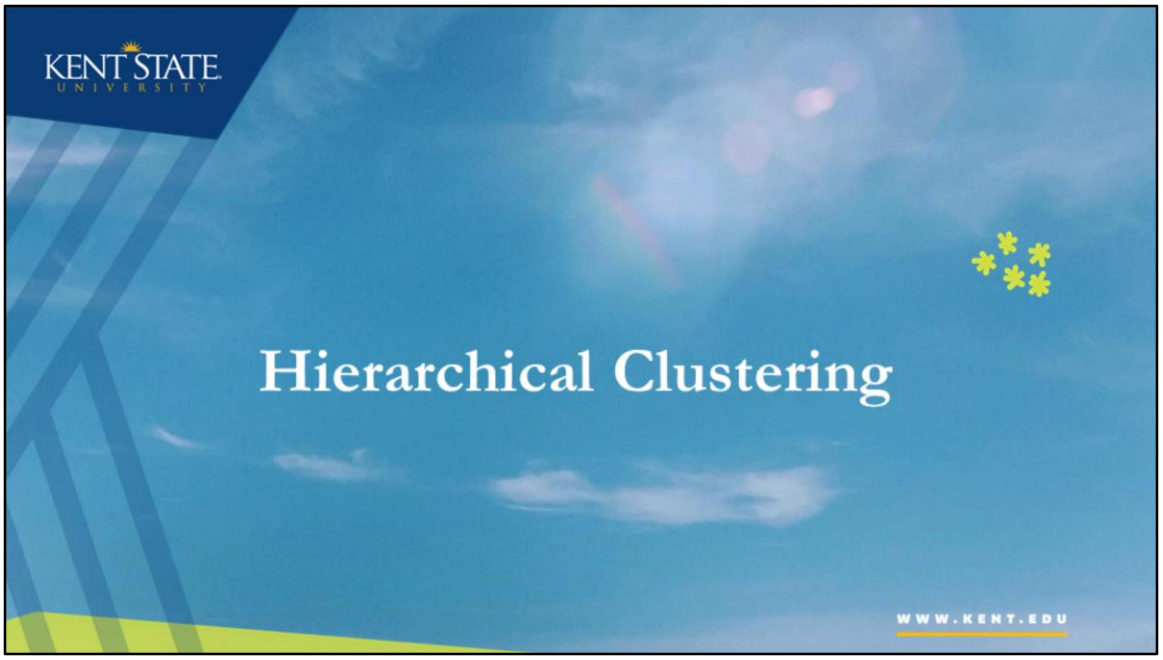
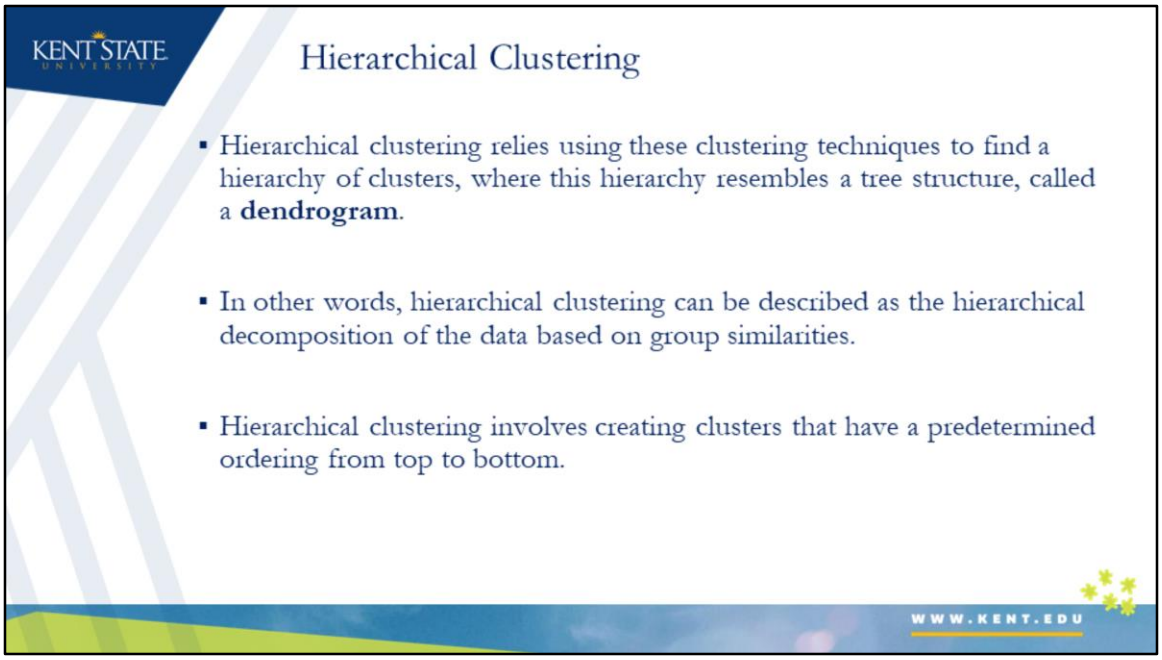


In this module, we discuss another type of clustering algorithm called Hierarchical Clustering.



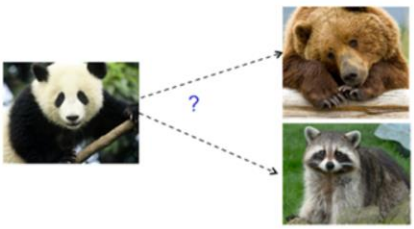
There are two general approaches to clustering: Hierarchical and non-hierarchical clustering. An example of the latter was k-means clustering. Here, we discuss hierarchical clustering.

Hierarchical methods can be either agglomerative or divisive. Agglomerative methods begin with n clusters and sequentially merge similar clusters until a single cluster is obtained. Divisive methods work in the opposite direction, starting with one cluster that includes all records. Hierarchical methods are especially useful when the goal is to arrange the clusters into a natural hierarchy, as is illustrated by the next example.

KENT STATE UNIVERSITY

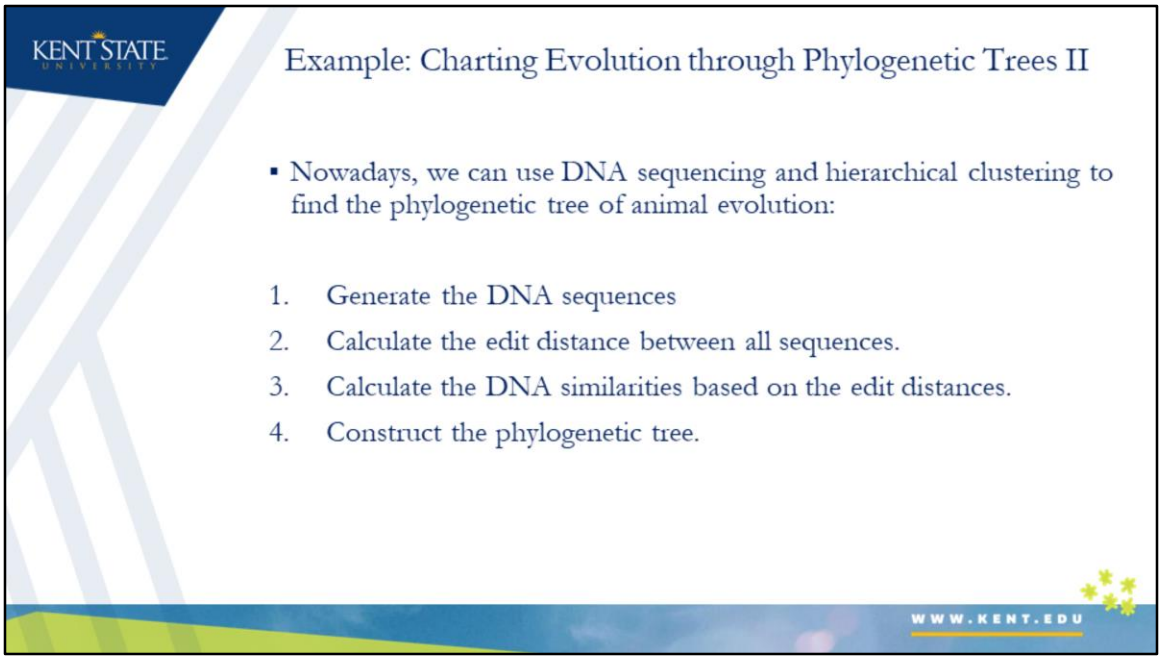
Example: Charting Evolution through Phylogenetic Trees

- How can we relate different species together?
- In the decades before DNA sequencing was reliable, the scientists struggled to answer a seemingly simple question: Are giant pandas closer to bears or racoons?



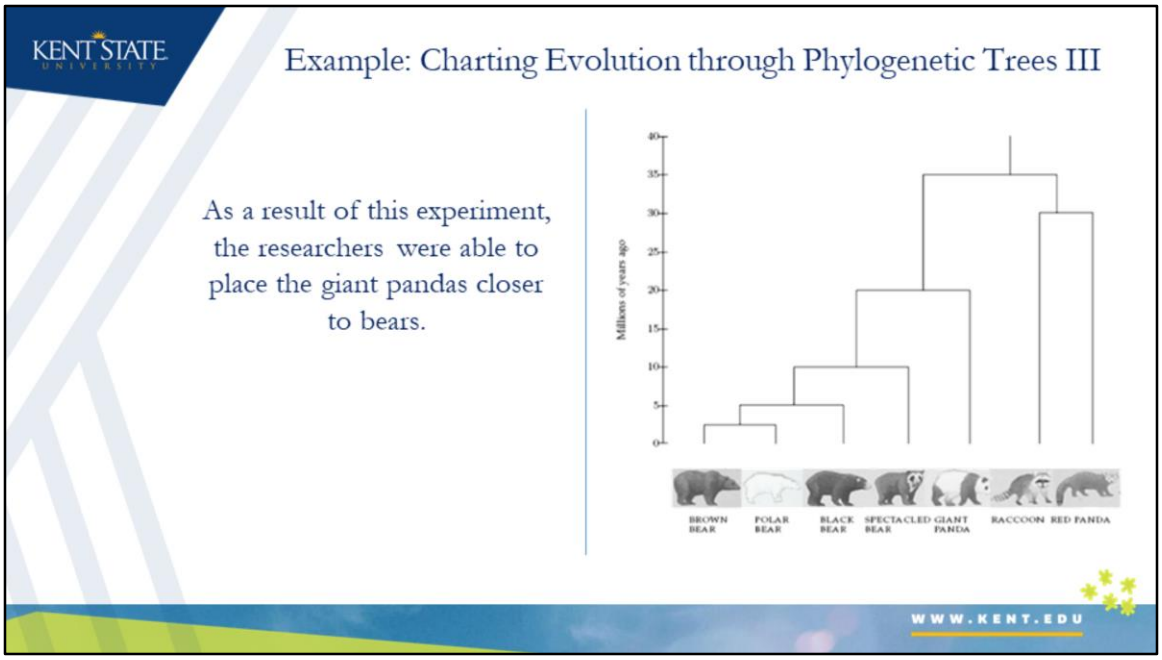
WWW.KENT.EDU

This is an important application of clustering, i.e., to find a natural grouping of species with respect to some known evolutionary attributes.



Deriving the known attributes through DNA sequencing, and then applying hierarchical clustering provides a natural grouping of species. Question: Why couldn't we use K-Means clustering here? Note that HC still uses the concept of distance, which can again be defined in several ways.

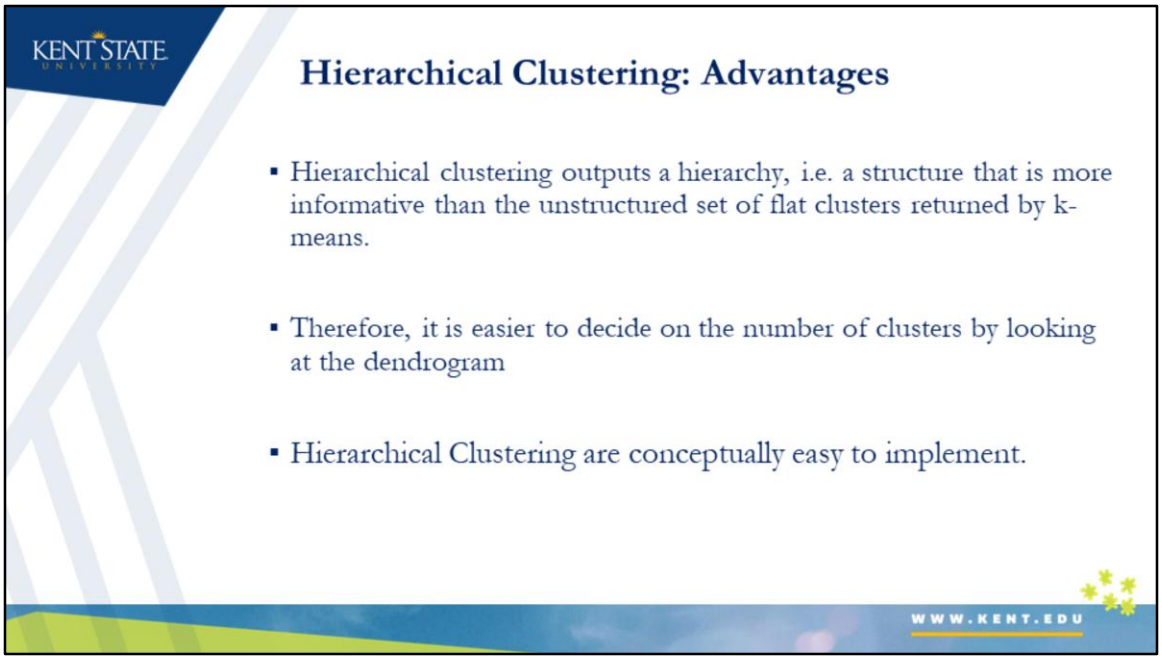
So, why HC and not k-means? The reason being that we don't really have an idea of how many groupings are needed, so determining optimal K in k-means might be difficult. In such cases, HC provides us with a better approach.



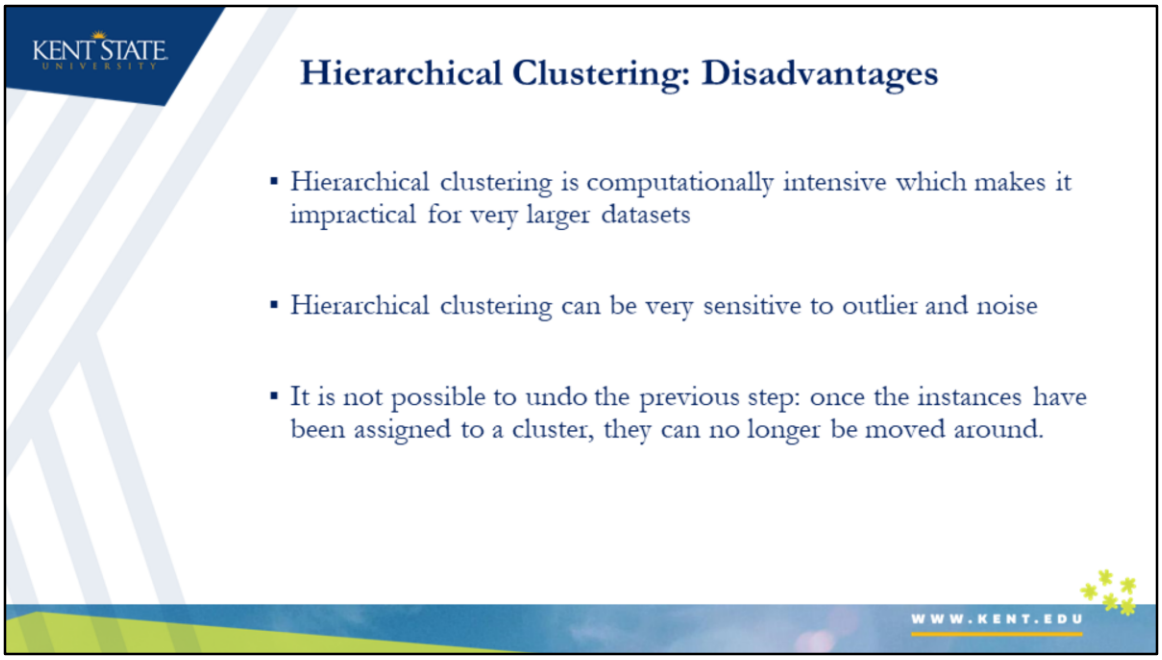
A dendrogram of the clustering algorithm is shown here.

A dendrogram is a treelike diagram that summarizes the process of clustering. On the x-axis are the records. Similar records are joined by lines whose vertical length reflects the distance between the records. The above figure shows the dendrogram for our example. Note that the figure is affected by the choice of distance, e.g., Euclidean versus City, and the type of linkage, centroid, single, etc.

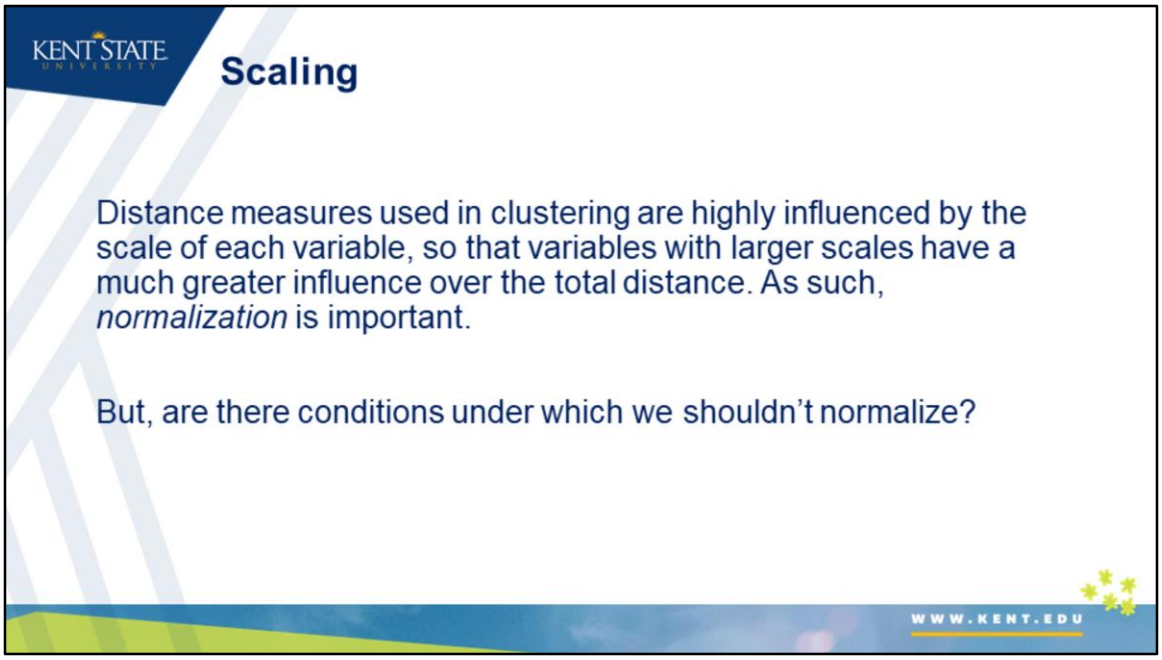
By choosing a cutoff distance on the y-axis, a set of clusters is created. Visually, this means drawing a horizontal line on a dendrogram. Records with connections below the horizontal line (that is, their distance is smaller than the cutoff distance) belong to the same cluster. For example, setting the cutoff distance to 22 on the above graph results in three clusters: The bears and Giant Panda in one cluster, while the Racoon and Red Panda in different clusters.



Hierarchical clustering is very appealing in that it does not require specification of the number of clusters, and in that sense is purely data-driven. The ability to represent the clustering process and results through dendrograms is also an advantage of this method, as it is easier to understand and interpret.



1. Hierarchical clustering requires the computation and storage of an n × n distance matrix. For very large datasets, this can be expensive and slow
2. The hierarchical algorithm makes only one pass through the data. This means that records that are allocated incorrectly early in the process cannot be reallocated subsequently.
3. Hierarchical clustering also tends to have low stability. Reordering data or dropping a few records can lead to a different solution.
4. With respect to the choice of distance between clusters, single and complete linkage are robust to changes in the distance metric (e.g., Euclidean, statistical distance) as long as the relative ordering is kept. In contrast, average linkage is more influenced by the choice of distance metric, and might lead to completely different clusters when the metric is changed.
5. Hierarchical clustering is sensitive to outliers.



As with most algorithms using distance measures, HC is sensitive to the scale of data. Always normalize. But, are there circumstances when we shouldn't?