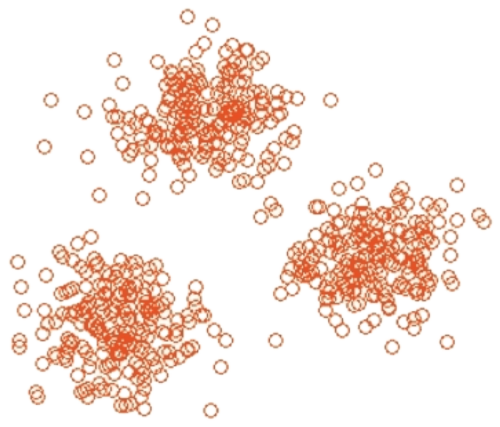
Clustering Algorithms

- Data mining is sorting through data to identify patterns and establish relationships.
- Extract useful knowledge from large data sets!
- Clustering is a data-mining (machine learning) technique used to place data elements into related groups without advance knowledge of the group definitions.

Hard vs. Soft Clustering

- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do.
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies.
  - You may want to put a pair of sneakers in two clusters: (1) sports apparel and (2) shoes.
  - You can only do that with a soft-clustering approach.
- See book IIR Section 16.5, 18.

Clustering Algorithms



- Clustering is a data-mining (machine learning) algorithm used to cluster observations into groups of related observations without any prior knowledge of those relationships.
- Try to locate homogeneous groups of observations.
- Clustering algorithms try to formalize this.

Distance Metric Is More Important Than the Clustering Algorithm

- Attach a label to each observation or data point in a set.
- You can say this is "unsupervised classification."
- Intuitively, you would want to assign the same label to data points that are "close" to each other.
- Thus, clustering algorithms rely on a distance metric between data points.
- Sometimes it is said that, for clustering, the distance metric is more important than the clustering algorithm.

Distances: Quantitative Variables

Data point:

$$x_i = [x_{i1} \dots x_{ip}]^T$$

- Identity (absolute) error

$$d_j(x_{ij}, x_{i'j}) = I(x_{ij} \neq x_{i'j})$$

- Squared distance

$$d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$$

- $L_q$norms

$$L_{qii'} = [\sum | x_{ij} - x_{i'j} |^q]^{1/q}$$

- Canberra distance

$$d_{ii'} = \sum \frac{| x_{ij} - x_{i'j} |}{| x_{ij} + x_{i'j} |}$$

- Correlation

$$\rho(x_i, x_{i'}) = \frac{\sum_j (x_{ij} - \bar{x}_i)(x_{i'j} - \bar{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{i'j} - \bar{x}_{i'})^2}}$$

Distances: Ordinal and Categorical Variables

- Ordinal variables can be forced to lie within (0, 1), and then a quantitative metric can be applied:

$$\frac{k-1/2}{M},\ k=1,2,\ldots,M$$

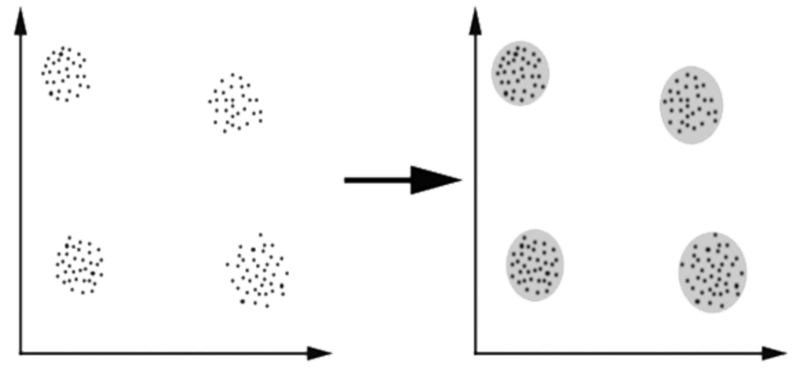
- For categorical variables, distances must be specified by user between each pair of categories:
  - E.g., distance in a hierarchy of categories.

Combining Distances

- Often weighted sum is used

$$D(x_i,x_j)=\sum_{l=1}^{p} w_l d(x_{il},x_{jl}),\quad \sum_{l=1}^{p} w_l=1,\ w_l>0.$$

Clustering Algorithms: Example



- Here we have four clusters, and the similarity criterion is distance.

Clustering Algorithms: Euclidian Distance

- Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

- Where *n* is the number of dimensions (attributes) and *pk* and *qk* are, respectively, the *k*th attributes (components) or data objects *p* and *q*.

Clustering Algorithms: Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance.

$$dist = (\sum_{k=1}^{n} | p_k - q_k |^r)^{\frac{1}{r}}$$

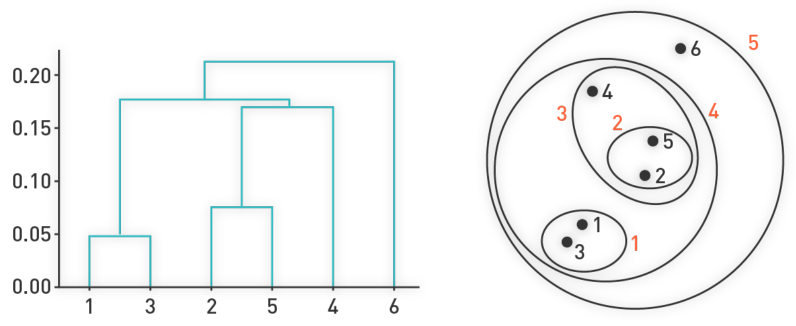
- Where *r* is a parameter, *n* is the number of dimensions (attributes) and *pk* and *qk* are, respectively, the *k*th attributes (components) or data objects *p* and *q*.

Clustering Algorithms: Nonhierarchical and Hierarchical

- Nonhierarchical clustering
  - Exclusive clustering (k-means) [in R, cluster package]
  - Overlapping clustering (fuzzy c means) [in R, e1071]
  - Probabilistic clustering (mixture of Gaussians)
- Hierarchical clustering
  - Agglomerative approach (bottom up)
    - In R: hclust() and agnes()
  - Divisive approach (top down)
    - In R: diana()

Question: Can both types of algorithm be effectively implemented in Hadoop?

Clustering Algorithms: Review

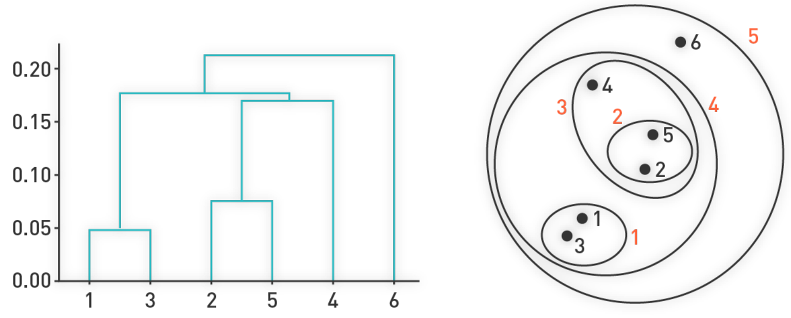


- Hierarchical clustering:
  - Produces a set of nested clusters organized as a hierarchical tree
  - Can be visualized as a dendrogram
    - A treelike diagram that records the sequences of merges or splits

Clustering Algorithms: Hierarchical

- Two main types of hierarchical clustering:
    - Agglomerative:
        - Start with the points as individual clusters.
        - At each step, merge the closest pair of clusters until only one cluster (or *k* clusters) is left.
    - Divisive:
        - Start with one all-inclusive cluster.
        - At each step, split a cluster until each cluster contains a point (or there are *k* clusters).
- Traditional hierarchical algorithms use a similarity or distance matrix.
    - Merge or split one cluster at a time.

Clustering Algorithms: Hierarchical Strengths



- No assumptions on the number of clusters
  - Any desired number of clusters can be obtained by "cutting" the dendrogram at the proper level
- Hierarchical clusterings may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., phylogeny reconstruction), Web (e.g., product catalogs)

Clustering Algorithms

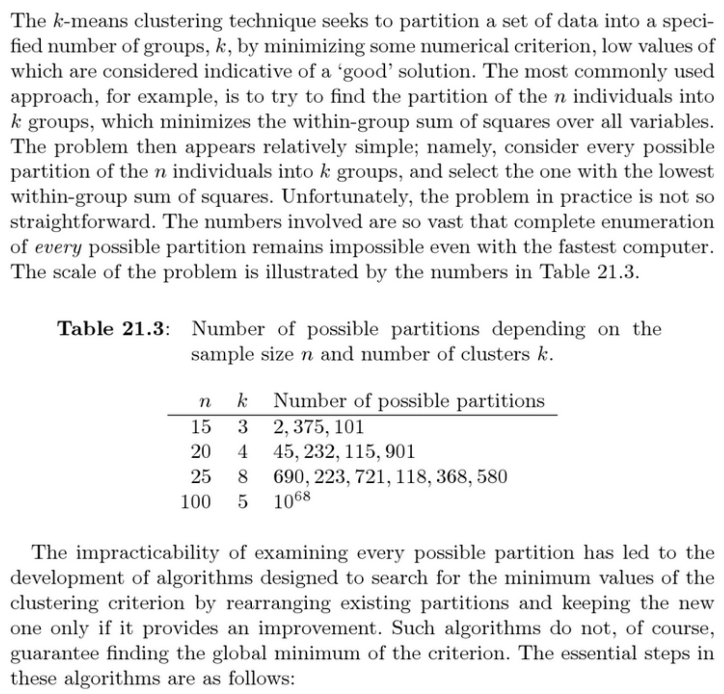
- Nonhierarchical Clustering
  - Exclusive Clustering (k-means) [in R, cluster package]
  - Overlapping Clustering (fuzzy c means) [in R, e1071]
  - Probabilistic Clustering (mixture of Gaussians)
- Hierarchical Clustering
  - Agglomerative approach (bottom up)
    - In R: hclust() and agnes()
  - Divisive approach (top down)
    - In R: diana()

K-Means Algorithm

- K-means clustering aims to partition *n* observations into *k*clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.
  - This results in a partitioning of the data space into Voronoi cells.
- The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum.
- These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both algorithms.

K-Means: Partitioning

Partition data so that it minimizes the within group sum of squares over all variables



The k-means clustering technique seeks to partition a set of data into a specified number of groups, k, by minimizing some numerical criterion, low values of which are considered indicative of a 'good' solution. The most commonly used approach, for example, is to try to find the partition of the n individuals into k groups, which minimizes the within-group sum of squares over all variables. The problem then appears relatively simple; namely, consider every possible partition of the n individuals into k groups, and select the one with the lowest within-group sum of squares. Unfortunately, the problem in practice is not so straightforward. The numbers involved are so vast that complete enumeration of *every* possible partition remains impossible even with the fastest computer. The scale of the problem is illustrated by the numbers in Table 21.3.

N examples with *k* clusters

Table 21.3: Number of possible partitions depending on the sample size n and number of clusters k.

n	k	Number of possible partitions
15	3	$2,375,101$
20	4	$45,232,115,901$
25	8	$690,223,721,118,368,580$
100	5	10^{68}

Impractical to explore all possible partitions

The impracticability of examining every possible partition has led to the development of algorithms designed to search for the minimum values of the clustering criterion by rearranging existing partitions and keeping the new one only if it provides an improvement. Such algorithms do not, of course, guarantee finding the global minimum of the criterion. The essential steps in these algorithms are as follows:

K-Means Loop

- E-step: The "assignment" step is also referred to as the expectation step.
- M-step: The "update step" as maximization step, making this algorithm a variant of the generalized expectation-maximization algorithm.
- Until convergence

K-Means Algorithm

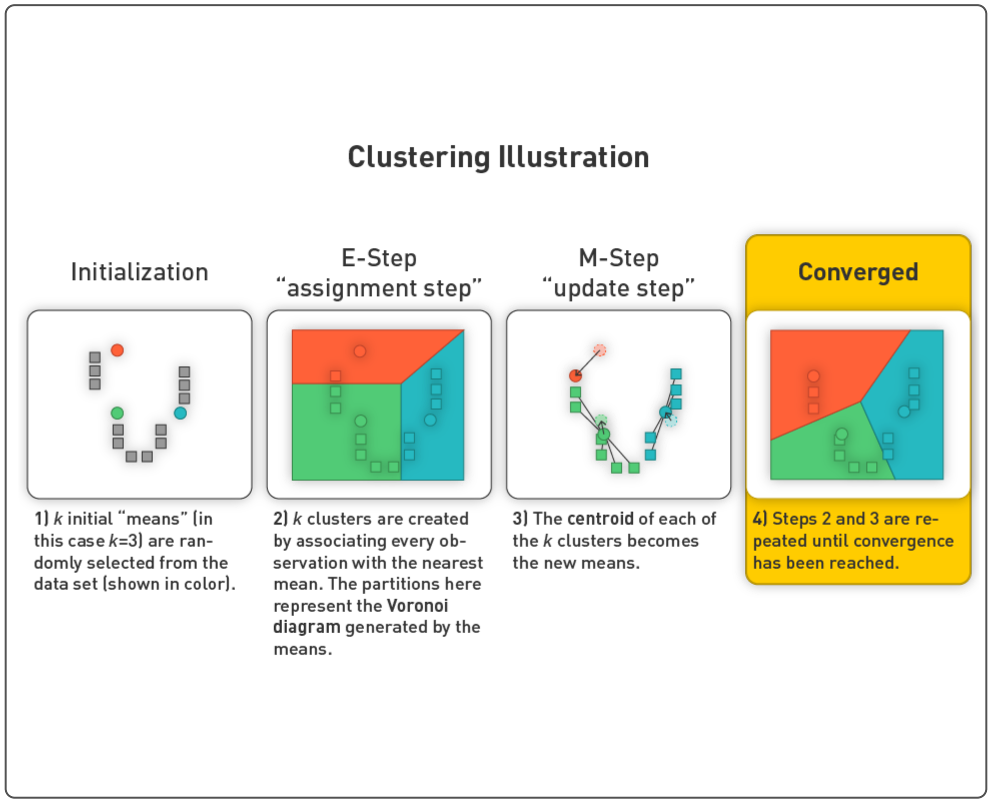
- For a current set of cluster means, assign each observation as:

$$C(i) = \arg\min_{1 \le k \le K} \|x_i - m_k\|^2,\ i = 1, \ldots, N$$

- For a given cluster assignment C of the data points, compute the cluster means m_k:

$$m_k = \frac{\sum_{i:C(i)=k} x_i}{N_k},\ k = 1, \ldots, K.$$

- Iterate above two steps until convergence



1) *k* initial “means” (in this case *k*=3) are randomly selected from the data set (shown in color).

2) *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the **Voronoi diagram** generated by the means.

3) The **centroid** of each of the *k* clusters becomes the new means.

4) Steps 2 and 3 are repeated until convergence has been reached.

Question: Clustering and Hadoop

Question: Can both types of algorithm be effectively implemented in Map Reduce?

Nonhierarchical Clustering

- Exclusive Clustering (k-means) [in R, cluster package]
- Overlapping Clustering (fuzzy c means) [in R, e1071]
- Probabilistic Clustering (mixture of Gaussians)

Hierarchical Clustering

- Agglomerative approach (bottom up)
  - In R: hclust() and agnes()
- Divisive approach (top down)
  - In R: diana()

Hierarchical clustering is tough to deploy. Distributing a bottom-up algorithm is tricky because each distributed process needs the entire data set to make choices about appropriate clusters. It also needs a list of clusters at its current level so it doesn't add a data point to more than one cluster at the same level.

Week 5 Summary

We have covered the following topics:

- ~~MRJob~~ (This is no longer covered in the course)
  - ~~Installation~~
  - ~~Fundamentals and concepts~~
  - ~~Writing MRJob code~~
- Logfile processing
- Serializable, JSON, and ~~MRJob~~ benchmarks
- Clustering algorithms
  - Clustering overview
  - *k*-means algorithm

Week 5 Summary (cont.)

During our live session for Week 5, we will discuss:

- Distributed *k*-means in ~~MRJob~~ (with a notebook)
- Data and distributions
- Model-based clustering
- EM algorithm for a mixture of Bernoulli distributions
- Distributed EM clustering algorithm in ~~MRJob~~