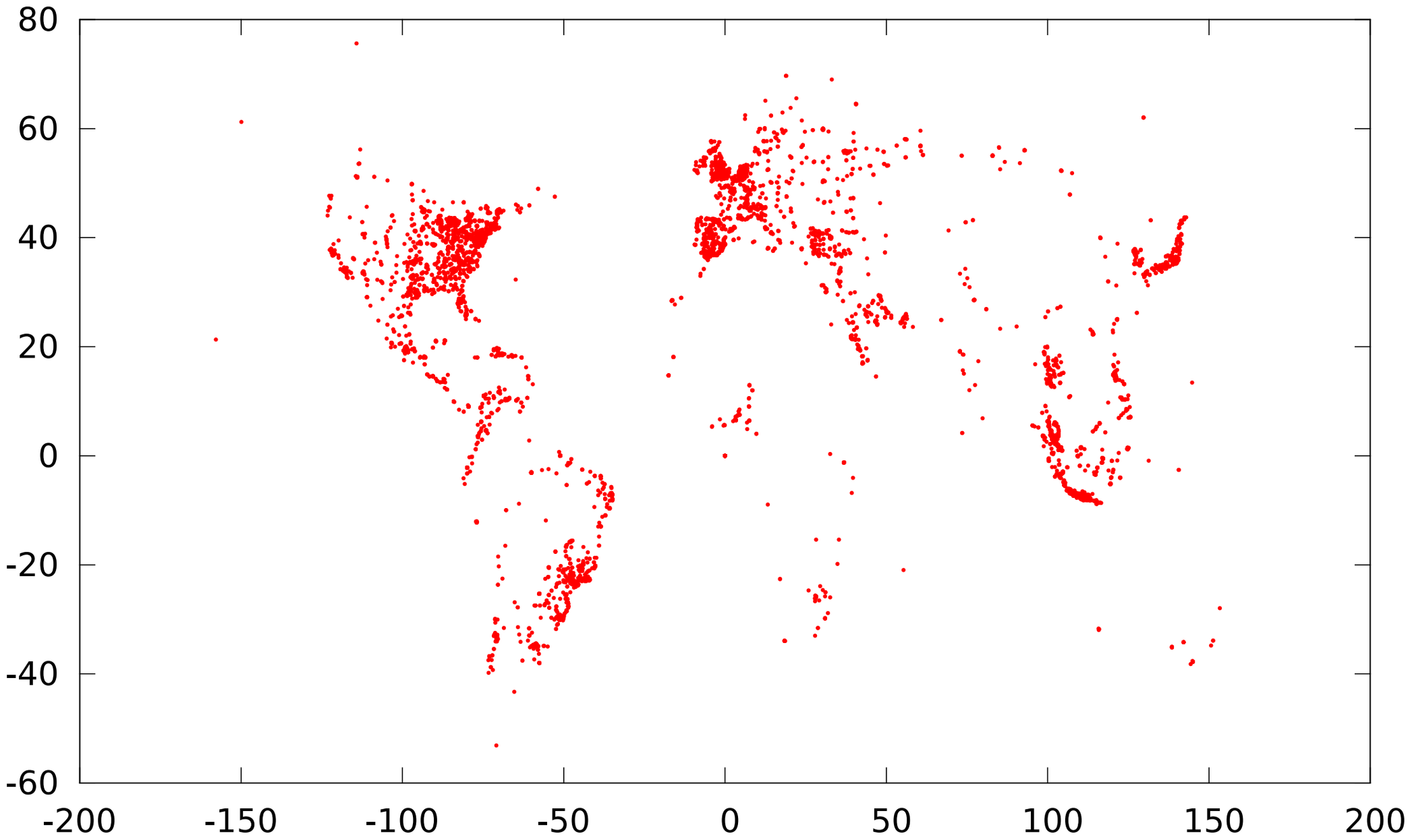
Lecture 24: Clustering

COSC 480 Data Science, Spring 2017

Michael Hay

Geo-tags of tweets



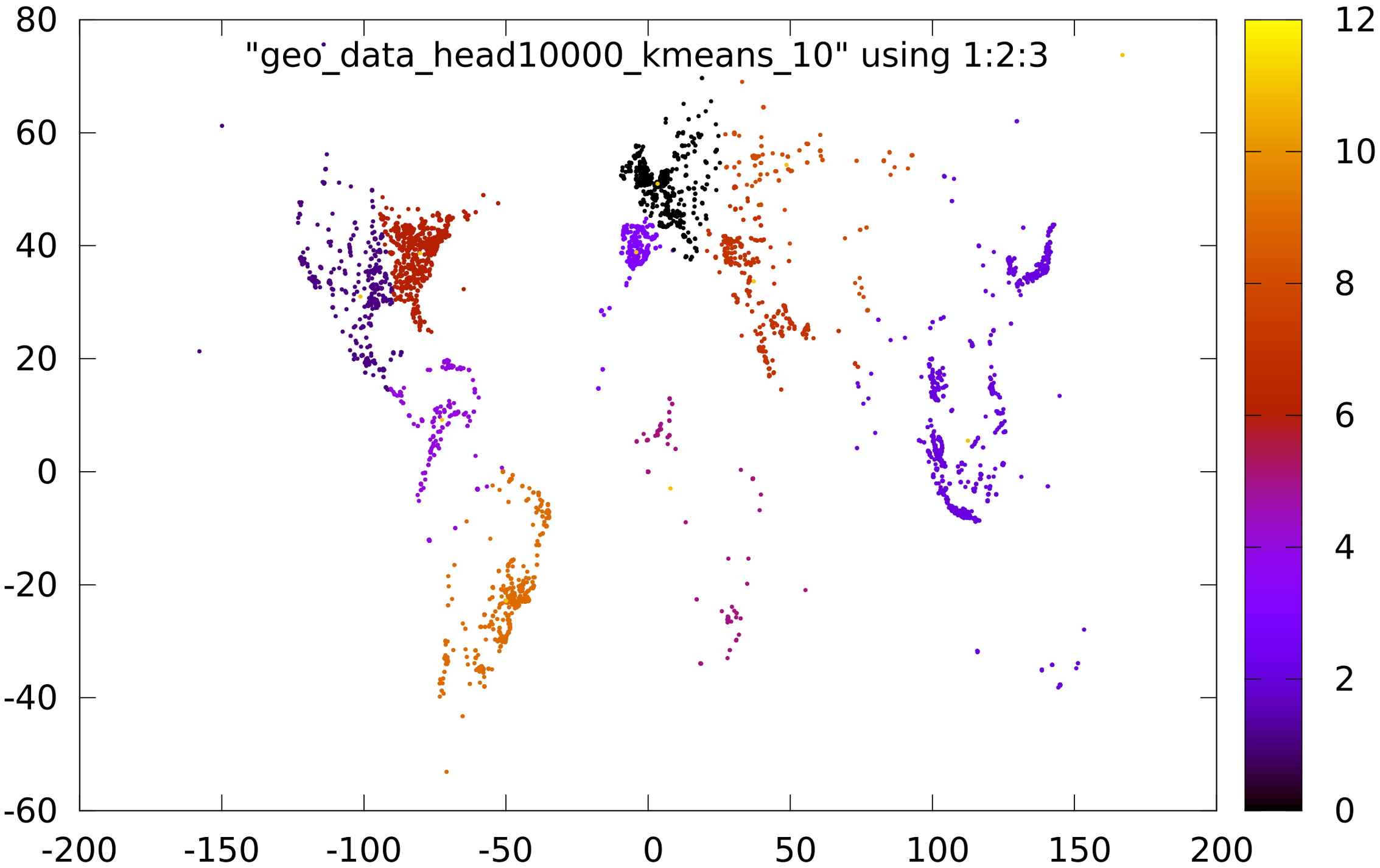
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Trending topics

- How would you compute trending topics?
  - Most frequent hashtags
  - Frequent keywords or phrases (which are not stopwords)
  - …
- But interesting trends in one region may not represent interesting trends in another.

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Idea: Cluster tweets by geography



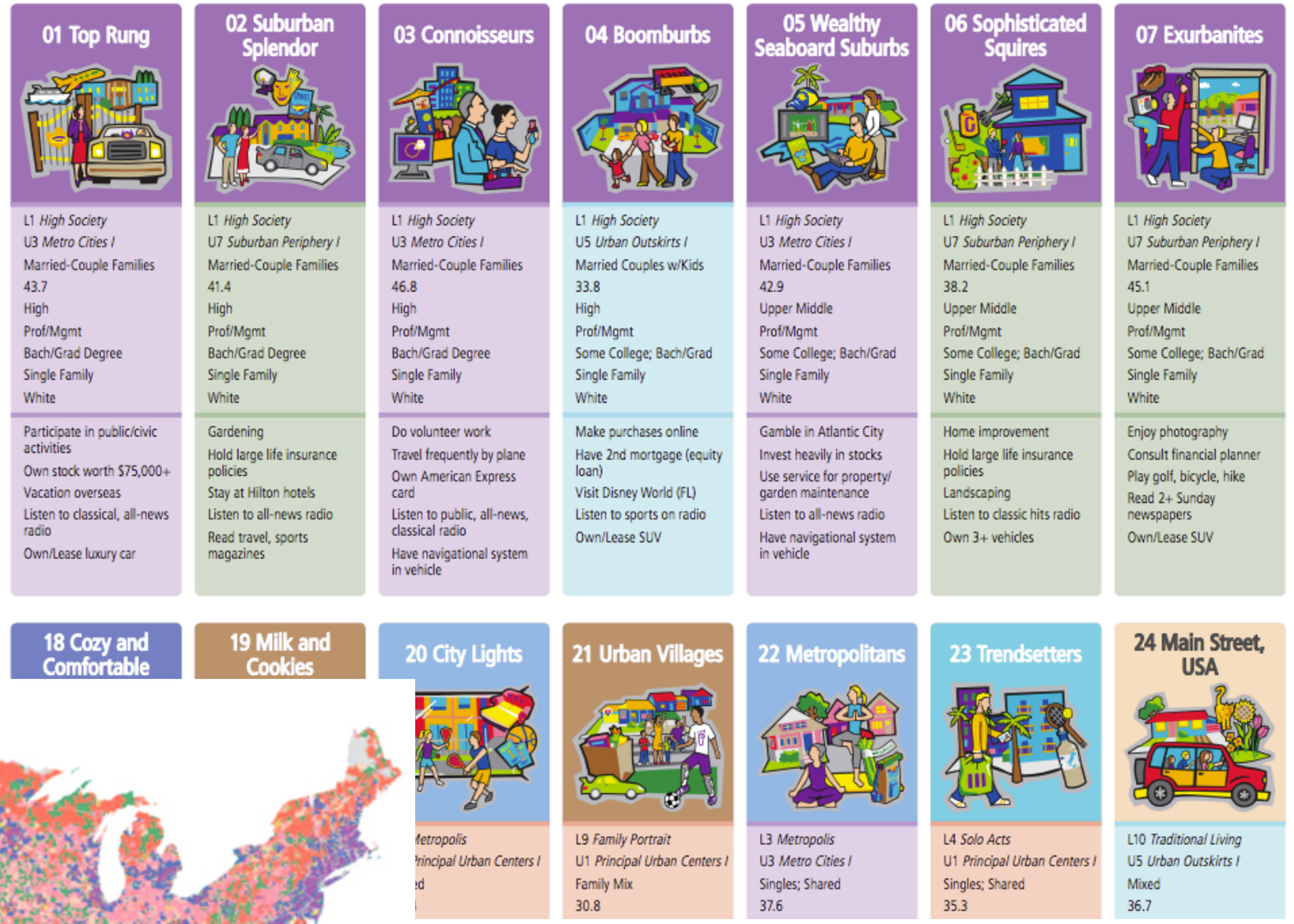
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

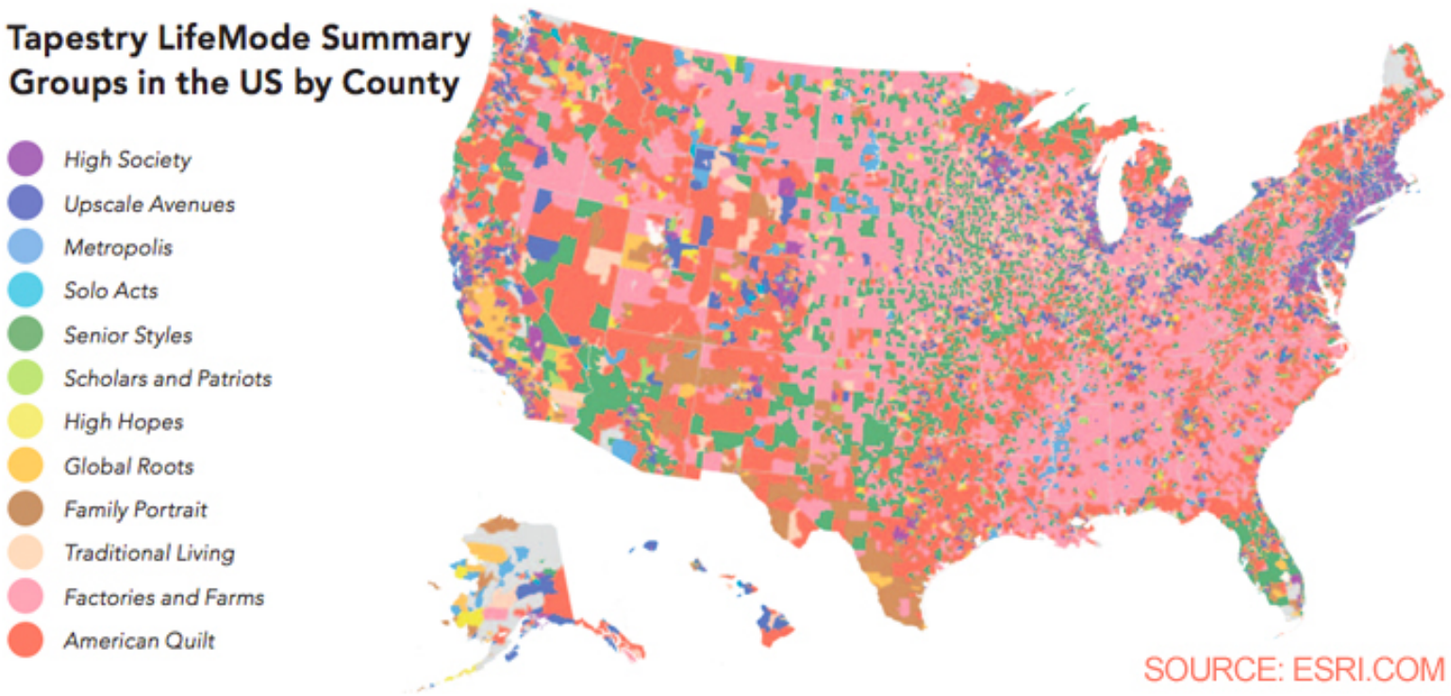
Trending topics by geography

- We can now compute trending topics within each cluster (region).

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Example: Market Segmentation

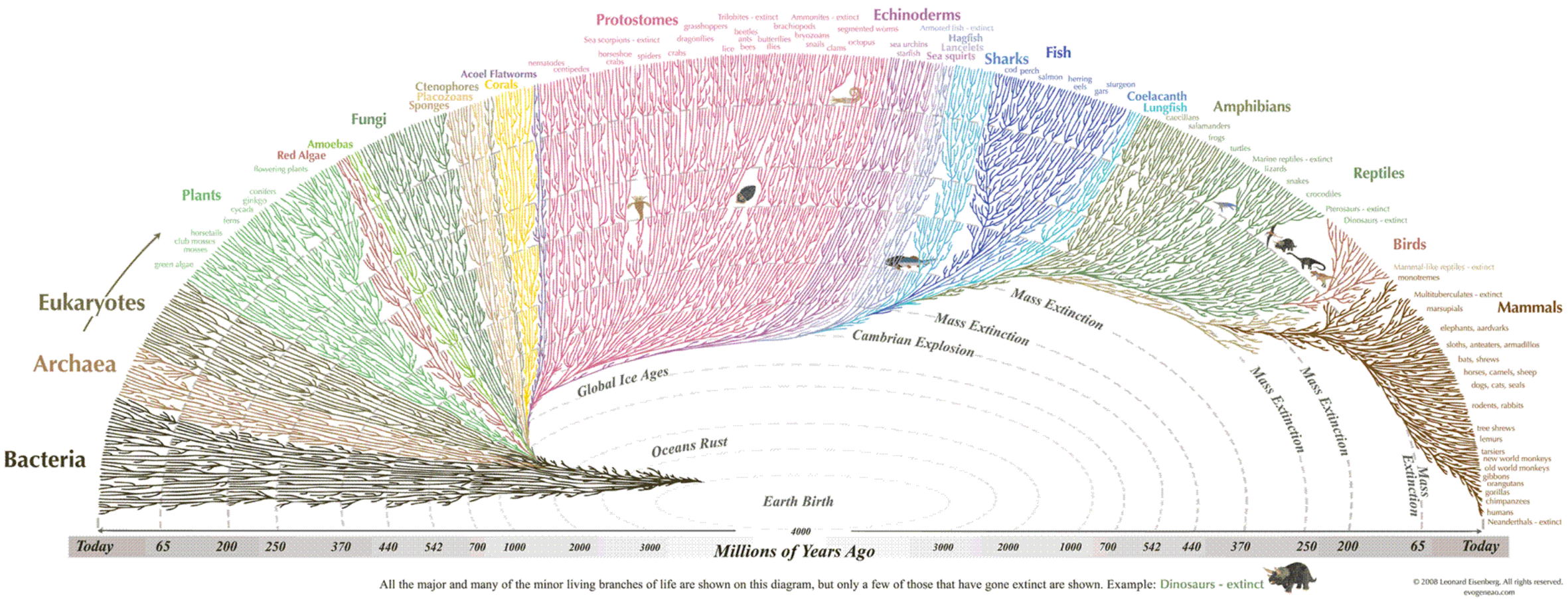




http://www.esriro.ro/library/fliers/pdfs/tapestry_segmentation.pdf#page=2

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Example: Phylogenetic Trees



Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Other Examples

- Image segmentation
- Document clustering
- De-duplication …

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Today

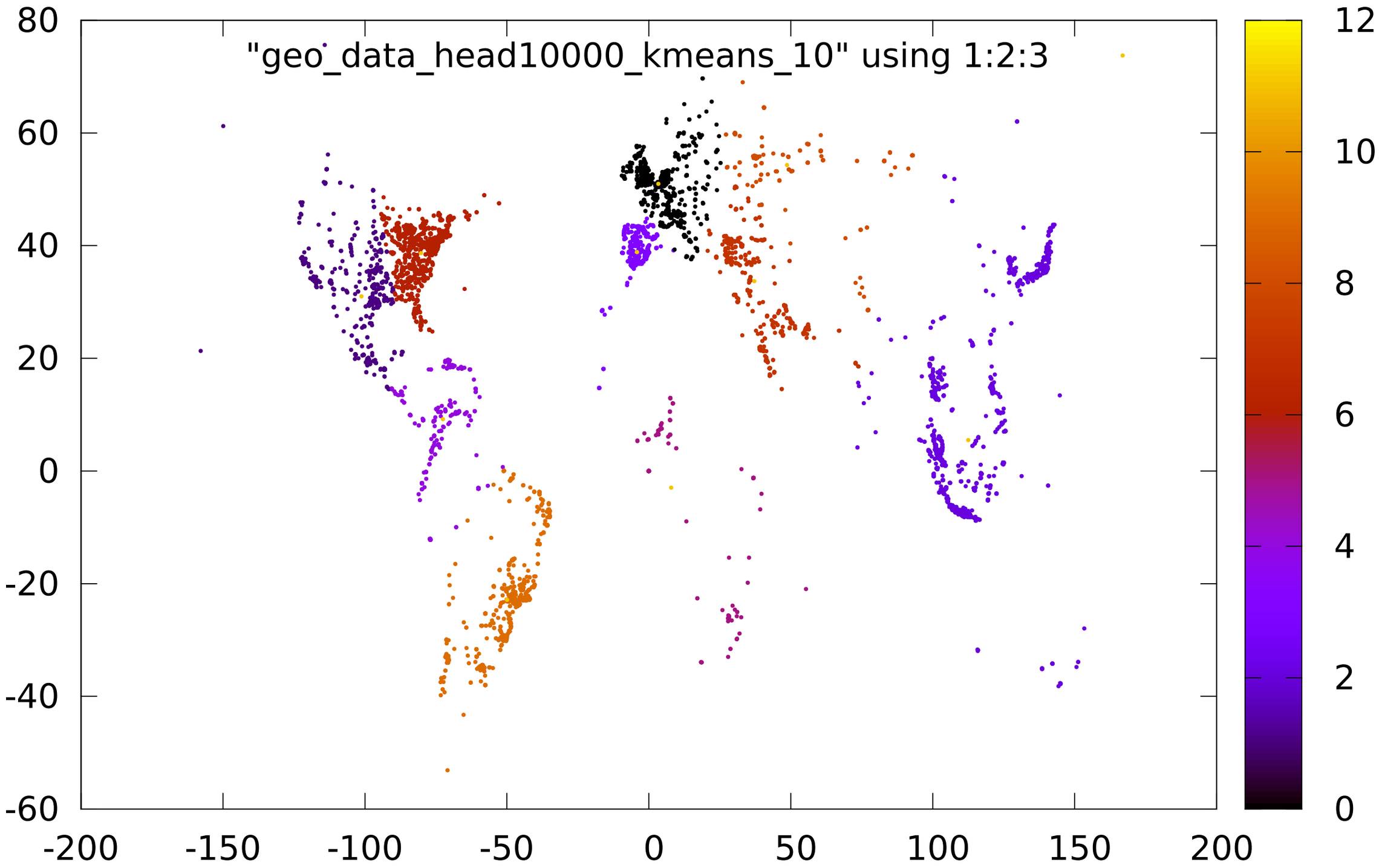
- K-means Clustering

- Distance Metrics

- Using distance metrics for clustering
  - K-medoids
  - Hierarchical Clustering

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

How did we create 10 clusters?



Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Can compare apples vs oranges …

- … if they are in the same ***feature space***.

- $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ is a dataset

- Each $\boldsymbol{x}_i$ is assumed to be a point in some d-dimensional space
  - $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]$
  - Each dimension represents a feature

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Euclidean Distance

$$\|\boldsymbol{x} - \boldsymbol{y}\|_2 = \sqrt{\left(\sum_i (x_i - y_i)^2\right)}$$

- Straight line distance between two points $\mathbf{x} = [x_1, x_2, \ldots, x_d]$ and $\mathbf{y} = [y_1, y_2, \ldots, y_d]$

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-means

- Partition a set of points $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ into k partitions $\boldsymbol{C} = \{C_1, C_2, \ldots, C_k\}$ that minimizes

$$RSS(\boldsymbol{C}) = \sum_{i=1}^{k} \sum_{j=1}^{n} a_{ij} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|_2^2$$

a_{ij} is 1 if
x_j is assigned to cluster C_i

Assignment Function

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-means

- Partition a set of points $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ into k partitions $\boldsymbol{C} = \{C_1, C_2, \ldots, C_k\}$ that minimizes

$$RSS(\boldsymbol{C}) = \sum_{i=1}^{k} \sum_{j=1}^{n} a_{ij} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|_2^2$$

$$\boldsymbol{\mu}_i = [\mu_{i1}, \mu_{i2}, \ldots, \mu_{id}]$$
$$\mu_{ij} = \sum_{\boldsymbol{x} \in C_i} \frac{x_j}{|C_i|}$$

μ_i is the mean of points in cluster C_i.

Cluster Representative

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-means

- Partition a set of points $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ into k partitions $\boldsymbol{C} = \{C_1, C_2, \ldots, C_k\}$ that minimizes

$$RSS(\boldsymbol{C}) = \sum_{i=1}^{k} \sum_{j=1}^{n} a_{ij} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|_2^2$$

Square of the straight line distance between x_j and its center μ_i.

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Chicken-and-Egg problem

- How do we minimize *RSS(**C**)* ?
  - If we know the cluster representatives (or the means), then it is easy to find the assignment function (which minimizes *RSS(**C**)*)
    - *Assign point to the closest cluster representative*
  - If we know the assignment function, computing the cluster representatives is easy
    - *Compute the mean of the points in the cluster*

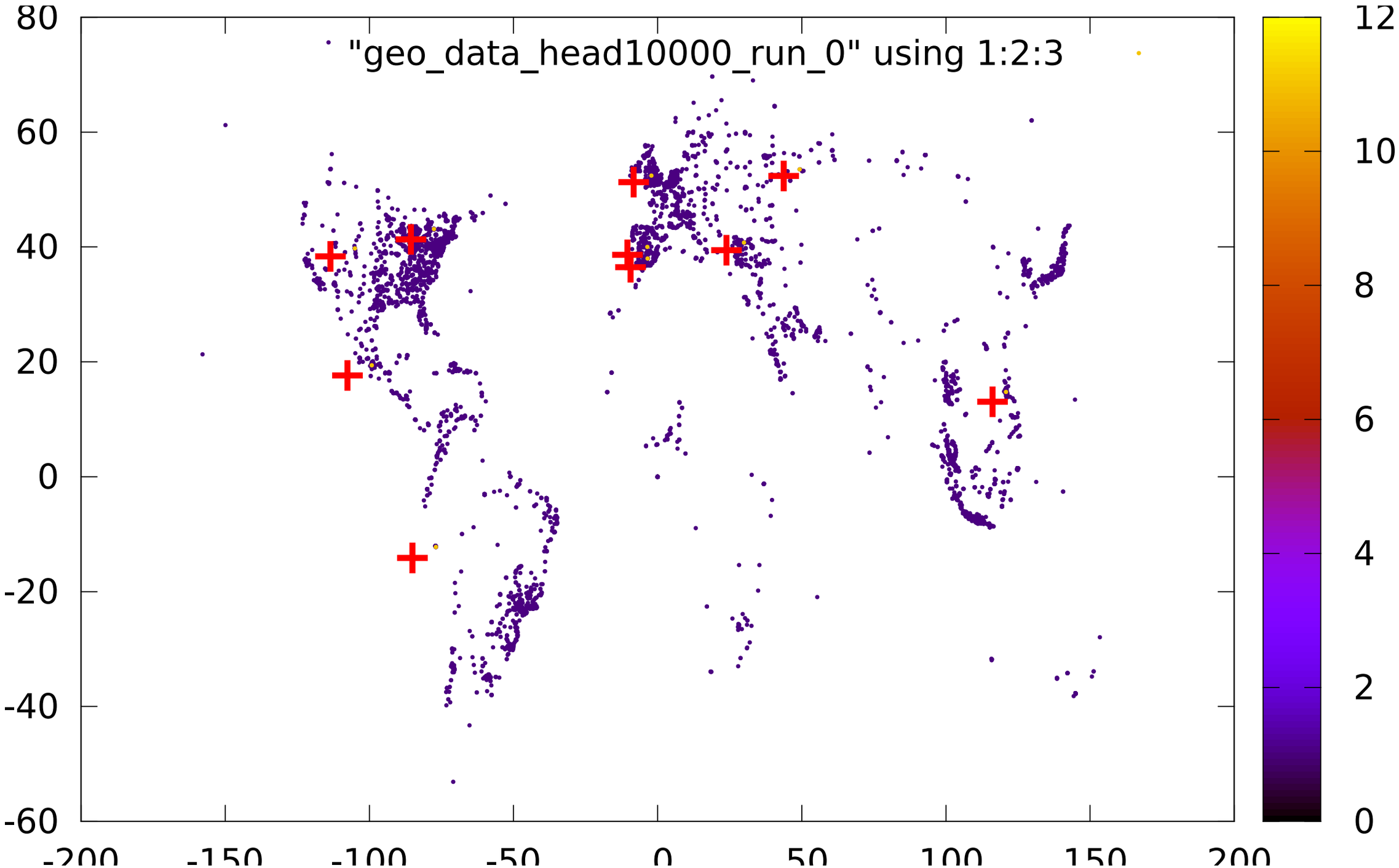
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-means Algorithm

- Idea: Alternate these two steps.
  - Pick some initialization for cluster representatives $\boldsymbol{\mu}^0$.
  - **E-step:**
    Assign points to the closest representative in $\boldsymbol{\mu}^i$.
  - **M-step:**
    Recompute the representatives $\boldsymbol{\mu}^{i+1}$ as means of the current clusters.

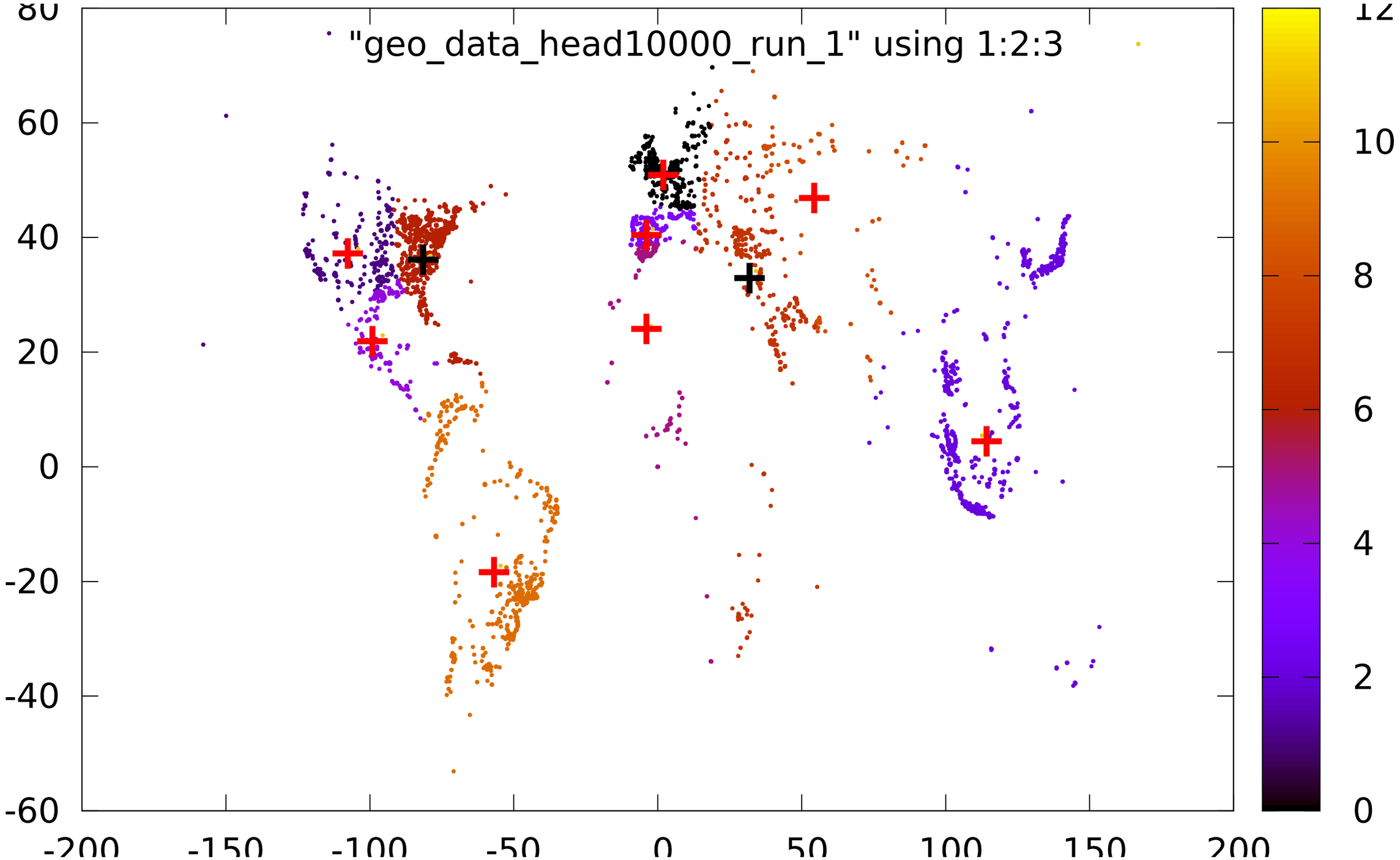
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-means: Initialization



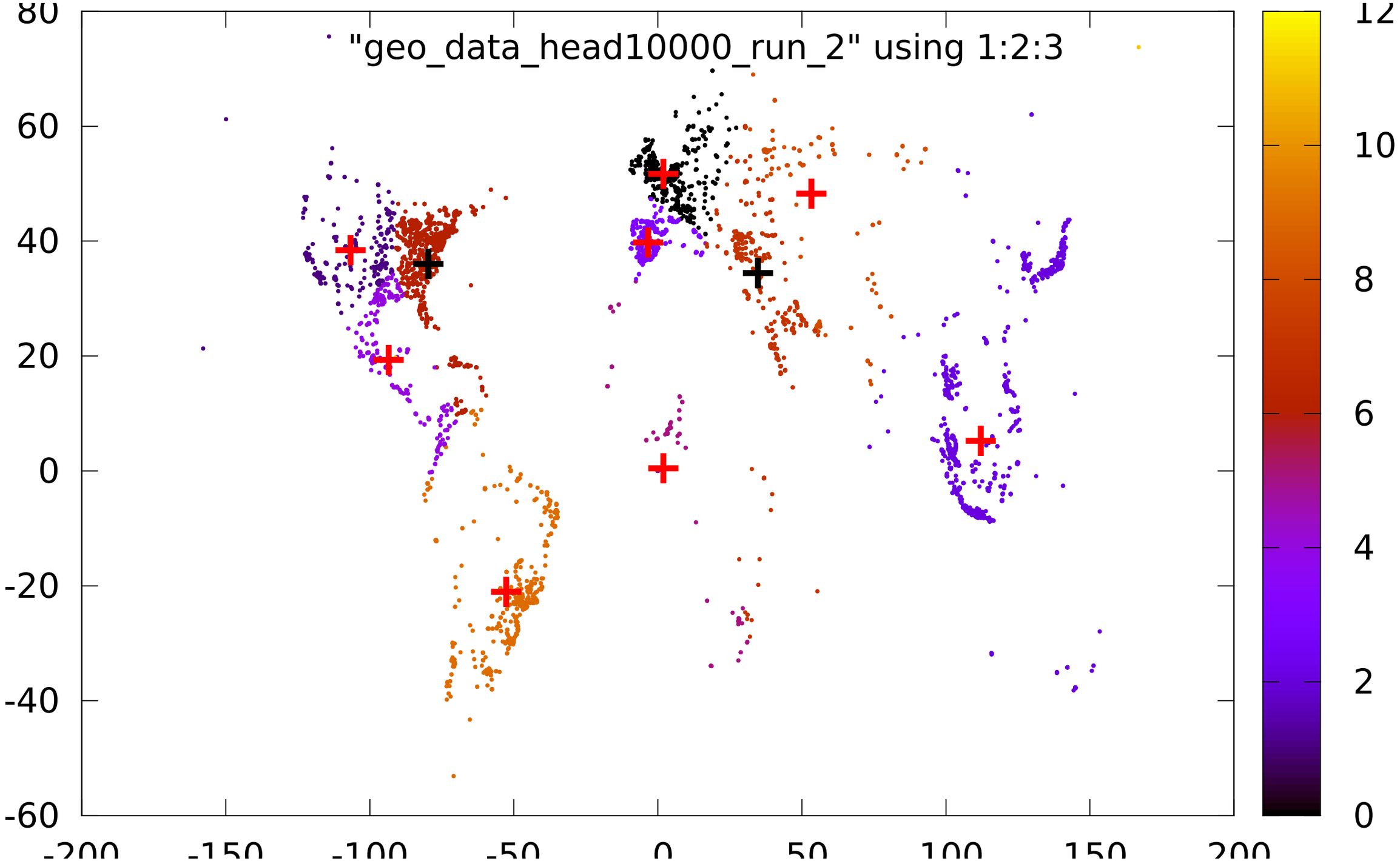
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-means: Iteration 1



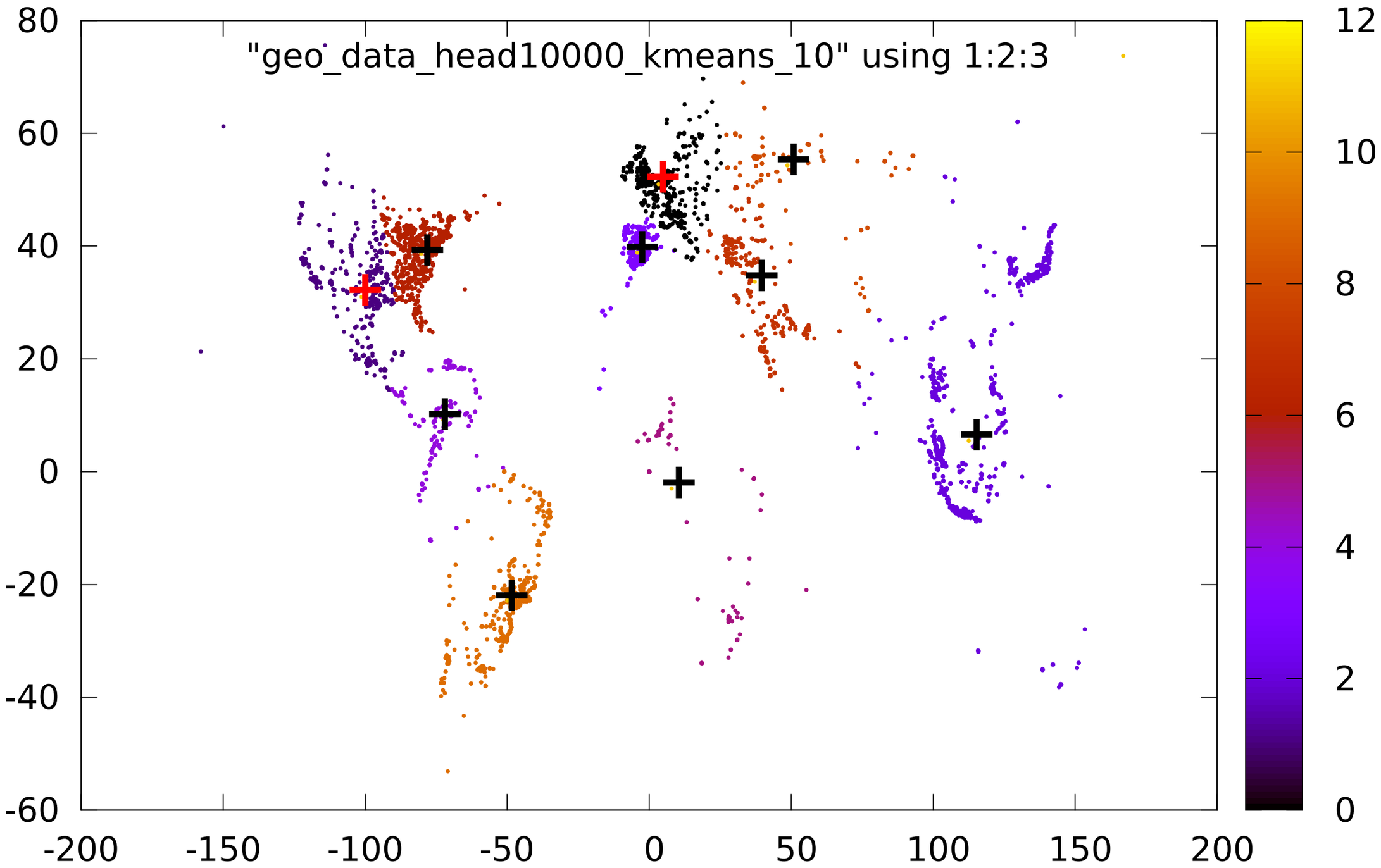
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-means: Iteration 2



Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-means: Iteration 10



Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Initialization

- Many heuristics
  - *Random*: K random points in the dataset
  - *Farthest First*:
    - Pick the first center at random
    - Pick the i^{th} center as the point "farthest away" from the last (i-1) centers
  - *K-means++*: (supplemental reference: paper)
    - Nice theoretical guarantees on quality of clustering

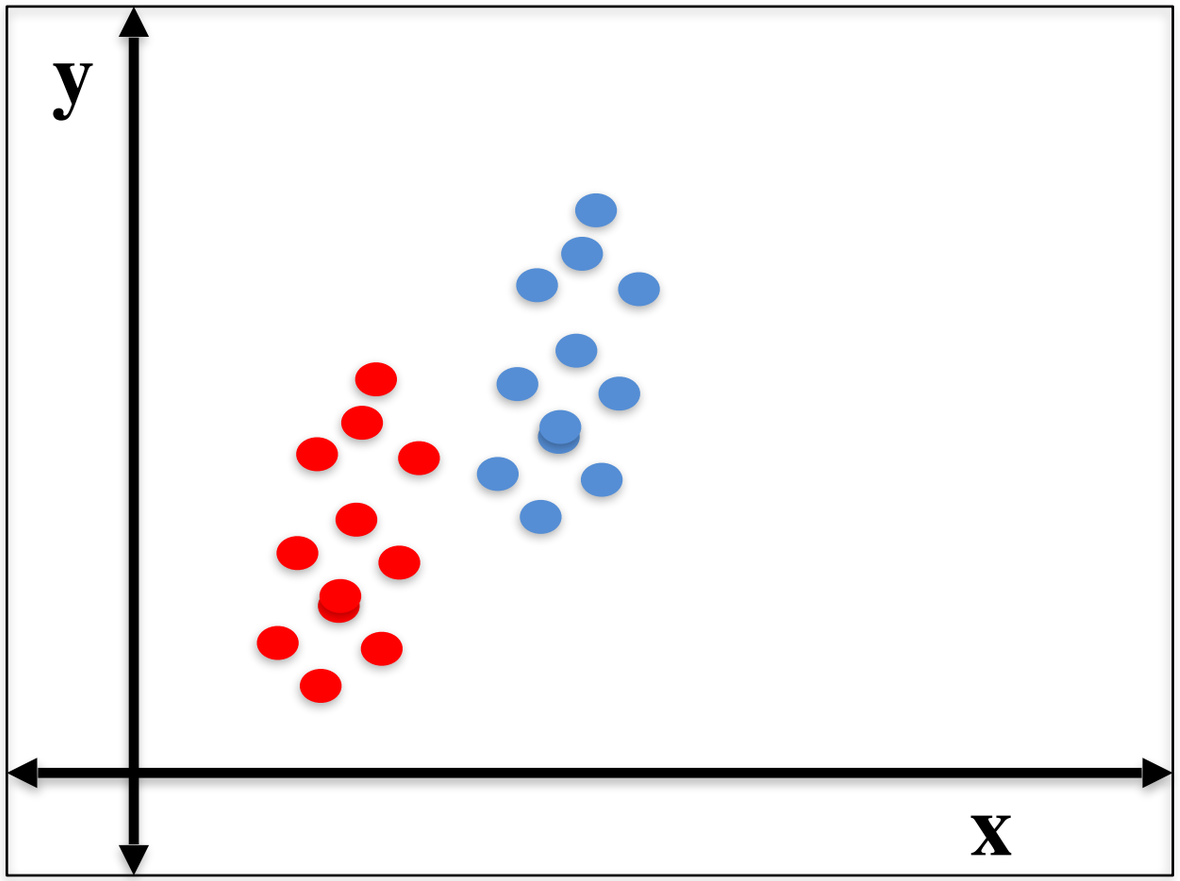
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

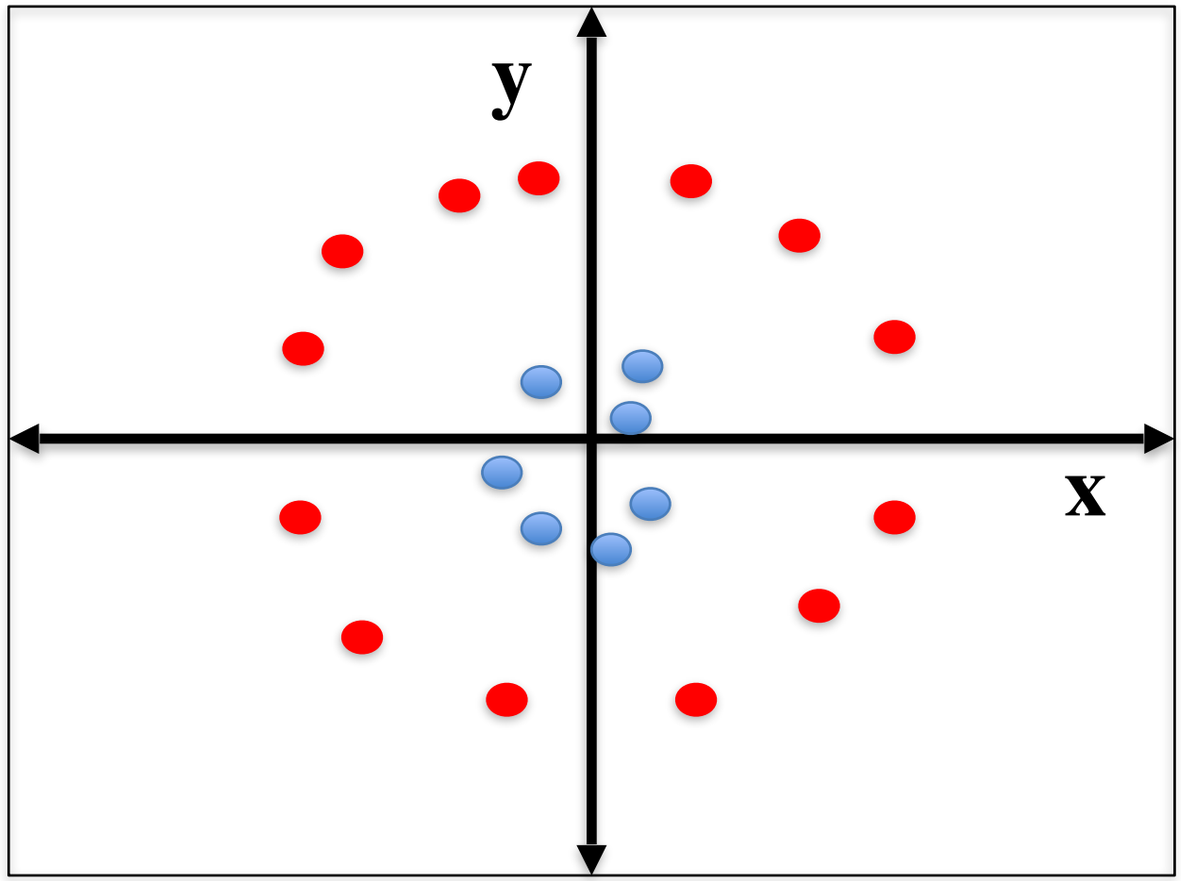
Stopping

- Alternate E and M steps till the cluster representatives do not change.
- Guaranteed to converge
  - To a **local optima** …
  - … but **not** necessarily to a **global optima**
    - *Finding the optimal solution (with least RSS(**C**)) is NP-hard, even for 2 clusters.*

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

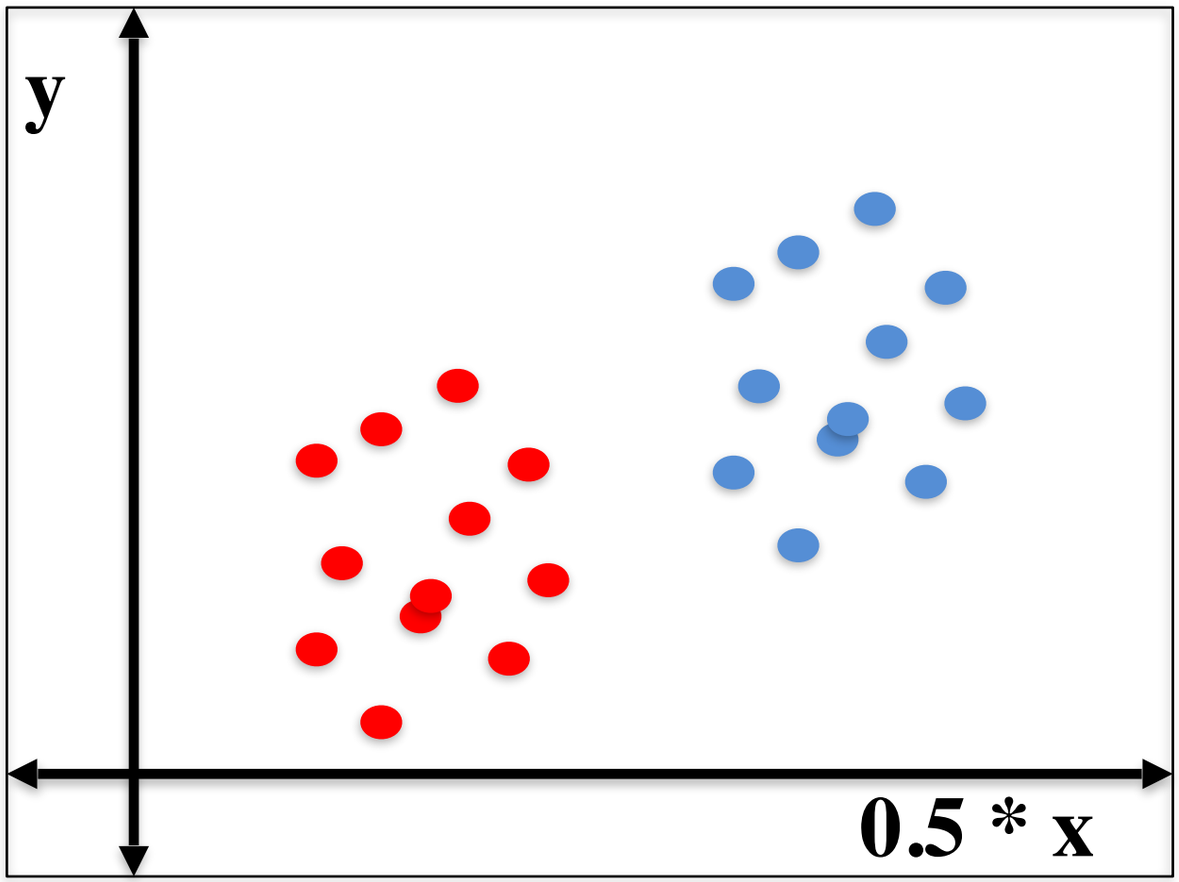
Where k-means fails …

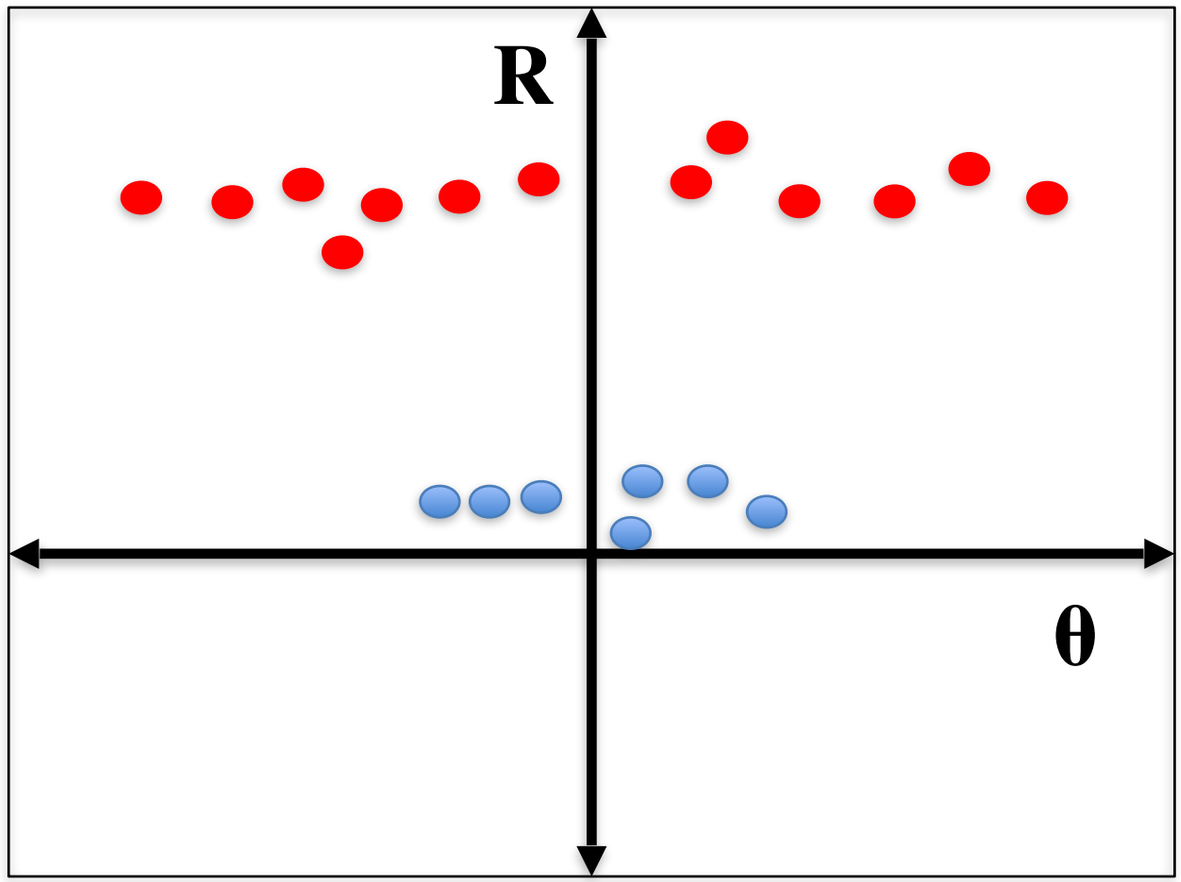




Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Scaling / changing features can help





Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Question

Instructions: ~1 minute to think/answer on your own; then discuss with neighbors; then I will call on one of you

Suppose we our dataset consists of records about individuals. Each individual is described by two attributes: age (in years) and salary (in dollars). We run k-means on the data with 10 clusters. What will happen?

A. The data will be clustered by age (clusters will contain points with similar age but possibly vastly different salaries)

B. The data will be clustered by salary (clusters will contain points with similar salary but possibly vastly different ages)

C. The data will be clustered by both salary and age (clusters will contain points with similar salaries and ages)

D. Not enough information given.

http://mdsr-book.github.io/exercises.html

Limitations of k-means

- Scaling/changing the feature space can change the solution.
- Cluster points into spherical regions.

- Number of clusters should be known apriori

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Outline

- K-means Clustering

- Distance Metrics

- Using distance metrics for clustering
  - K-medoids
  - Hierarchical Clustering

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Distance Metrics

- Function *d* that maps pairs of points $\boldsymbol{x}$, $\boldsymbol{y}$ to real numbers (usually between 0, 1)

- *Symmetric:* $d(\boldsymbol{x},\boldsymbol{y}) = d(\boldsymbol{y},\boldsymbol{x})$
- *Triangle Inequality:* $d(\boldsymbol{x},\boldsymbol{y}) + d(\boldsymbol{y},\boldsymbol{z}) \geq d(\boldsymbol{x},\boldsymbol{z})$

- Choice of distance metric is usually application dependent

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Euclidean Distance

$$\|\boldsymbol{x} - \boldsymbol{y}\|_2 = \sqrt{\left(\sum_i (x_i - y_i)^2\right)}$$

- Straight line distance between two points $\mathbf{x} = [x_1, x_2, \ldots, x_d]$ and $\mathbf{y} = [y_1, y_2, \ldots, y_d]$

- *K-means minimizes the sum of the Euclidean distances between the points and the centers*
  - *We use the mean as a center*

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Set-based Distances

- Let A and B be two sets.

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

  - *A measure of similarity (inverse of distance). Can subtract from 1 to get a distance.*

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Scaling / Changing features …

- … can be thought of as using a different distance function.
- How do we cluster for general distance functions?

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Outline

- K-means Clustering
- Distance Metrics
- Using distance metrics for clustering
  - K-medoids
  - Hierarchical Clustering

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-means for general distance functions?

- Mean of a set of points does not always make sense.
  - *Mean of a set of movies or a set of documents?*

- Mean *m* of a set of points *P* minimizes the sum of Euclidean distances between *m* and every point in *P*
  - *Best cluster representative under Euclidean Distance*

  - The above is not true for a general distance metric.

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-medoids

- Allows a general distance metric *d(x,y)*.

- Same algorithm as K-means …
- … but we don't pick the new centers using mean of the cluster.

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-medoids

- Pick some initialization for cluster representatives $\boldsymbol{\mu}^0$.
- **E-step:**
  Assign points to the closest representative in $\boldsymbol{\mu}^i$.
- **M-step:**
  Recompute the representatives $\boldsymbol{\mu}^{i+1}$ as the *medoid*, or **one of the points in the cluster with the minimum distance from all the other points.**

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Medoid

- m is the medoid of a set of points P if

$$m = \underset{x \in P}{\operatorname{argmin}} \left(\sum_{y \in P} d(x, y) \right)$$

Point that minimizes the sum of distances to all other points in the set.

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Computing the medoid

$$m = \underset{x \in P}{\operatorname{argmin}} \left(\sum_{y \in P} d(x, y) \right)$$

- Need to compute all $|P|^2$ distances.

- In comparison, computing the mean in k-means only requires computing d averages involving $|P|$ numbers each.

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

K-medoids summary

- Same algorithm as K-means, but uses medoids instead of means

- Centers are always points that appear in the original dataset

- Can use any distance measure for clustering.

- *Still need to know the number of clusters a priori…*

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Outline

- K-means Clustering
- Distance Metrics
- Using distance metrics for clustering
  - K-mediods
  - Hierarchical Clustering

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Hierarchical Clustering

- Rather than compute a single clustering, compute a family of clusterings.
- Can choose the clusters *a posteriori*.

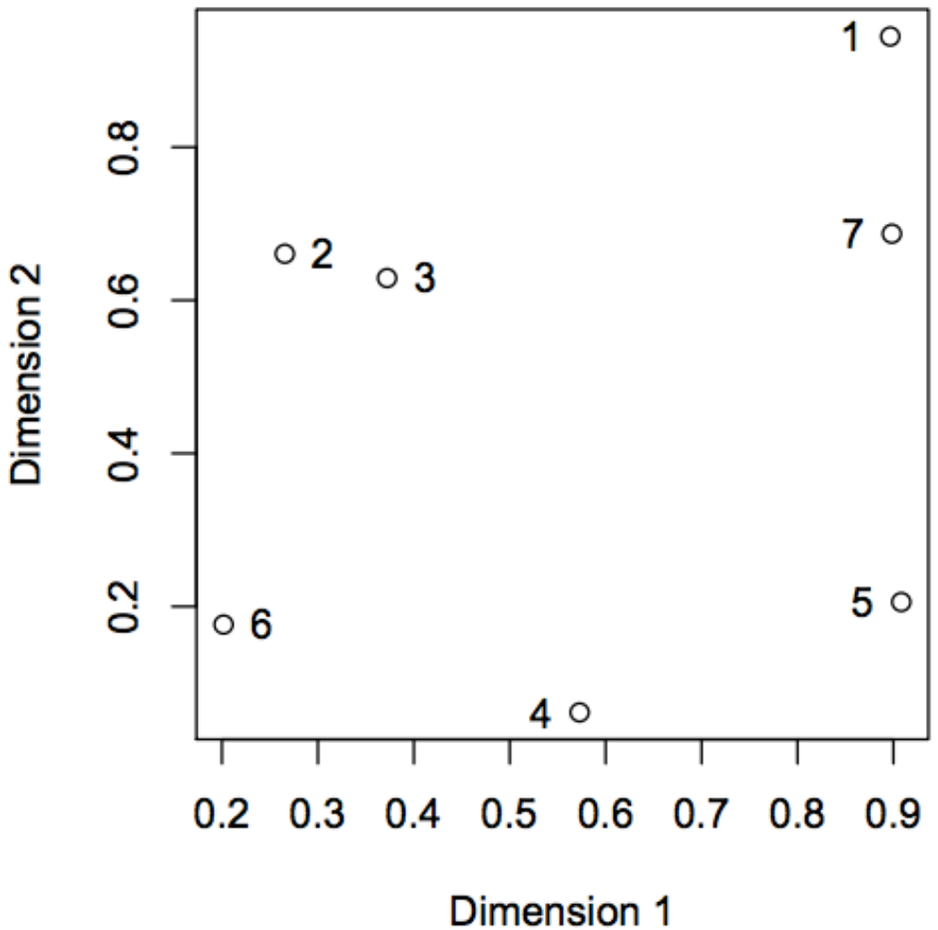
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Agglomerative Clustering

- Initialize each point to its own cluster
- Repeat:
  - Pick the two clusters that are *closest*
  - Merge them into one cluster
  - Stop when there is only one cluster left

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Example



Step 1: {1} {2} {3} {4} {5} {6} {7}

Step 2: {1} {2, 3} {4} {5} {6} {7}

Step 3: {1, 7} {2, 3} {4} {5} {6}

Step 4: {1, 7} {2, 3} {4, 5} {6}

Step 5: {1, 7} {2, 3, 6} {4, 5}

Step 6: {1, 7} {2, 3, 4, 5, 6}

Step 7: {1, 2, 3, 4, 5, 6, 7}

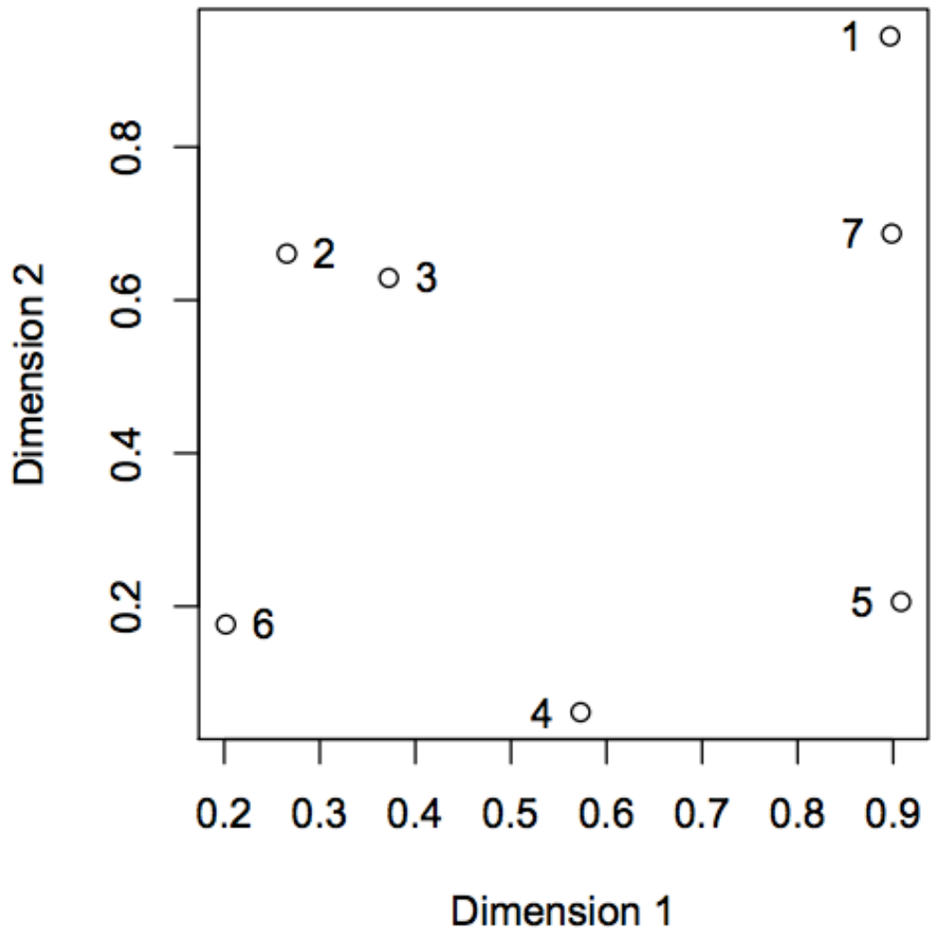
Example based on Ryan Tibshirani's slides

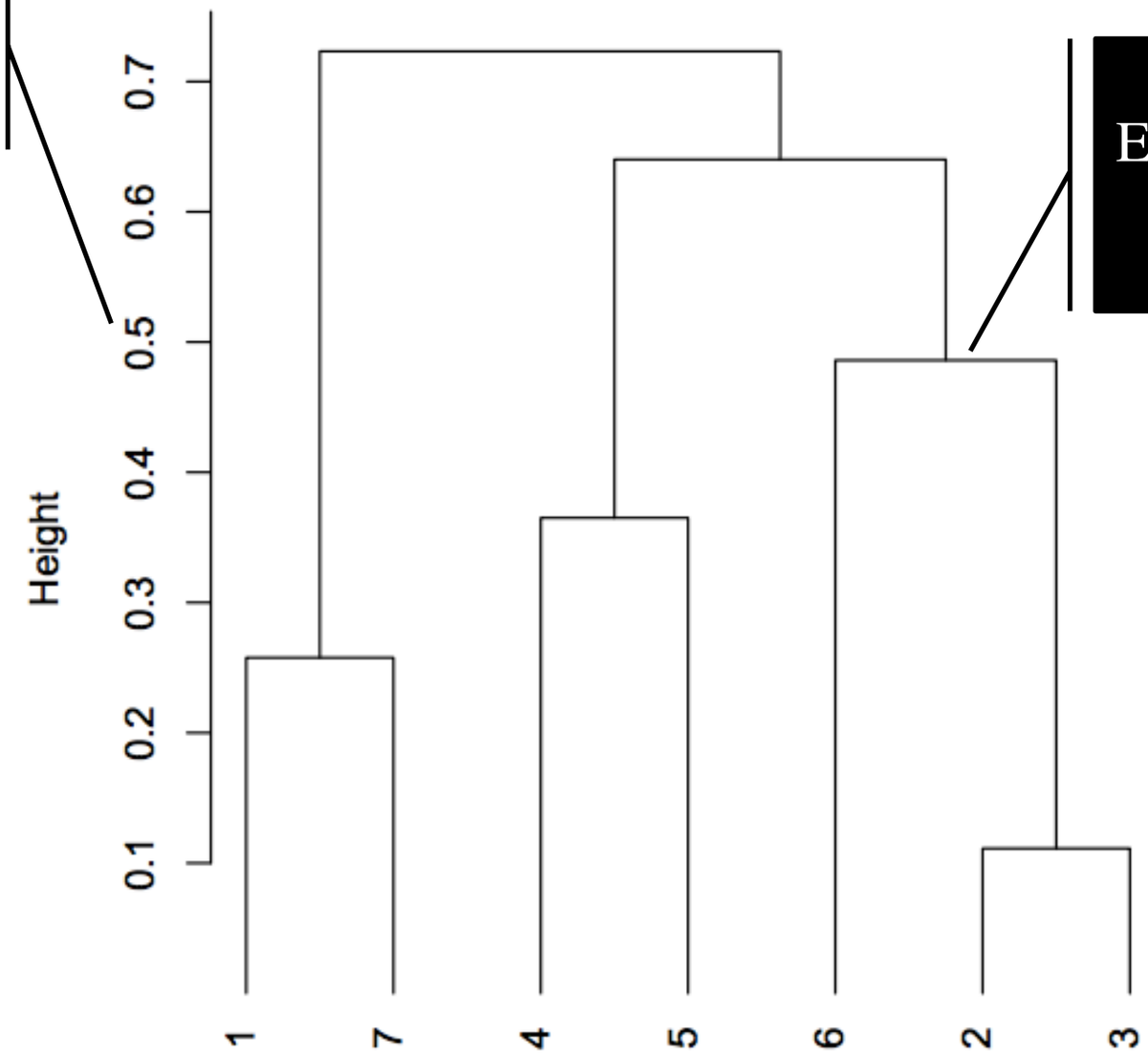
Dendrogram

Entire dataset

Height of a node is proportional to distance between children clusters

Each node is a cluster

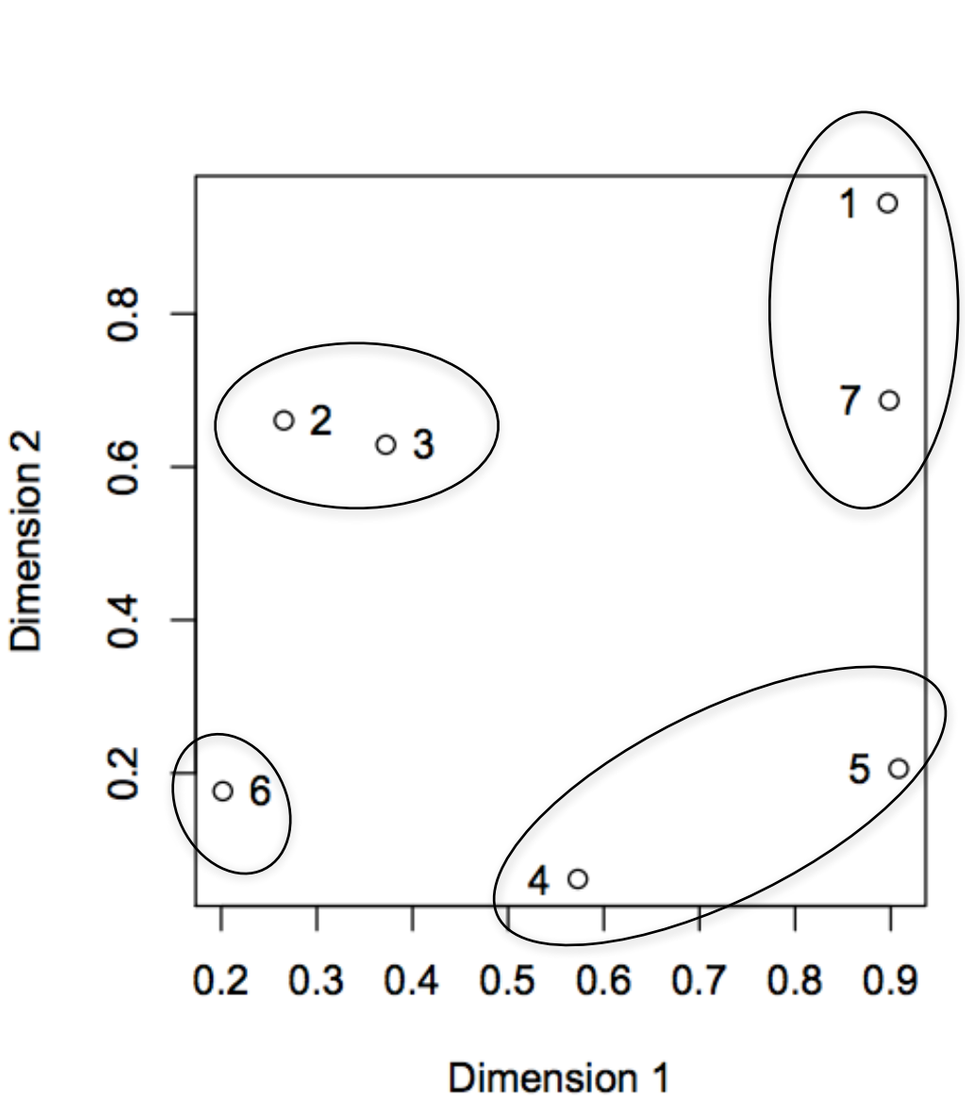


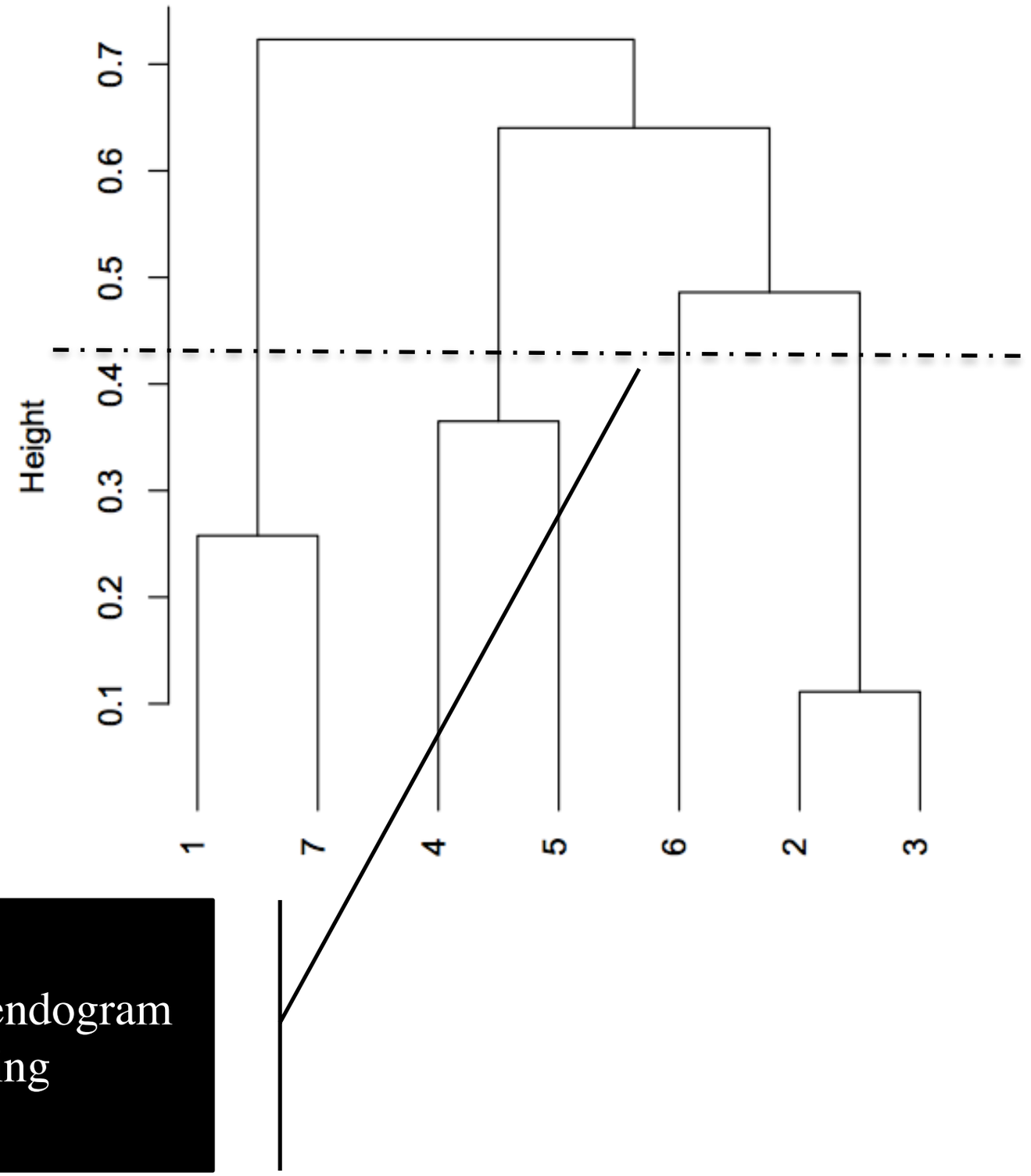


Individual points in the dataset

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Dendrogram

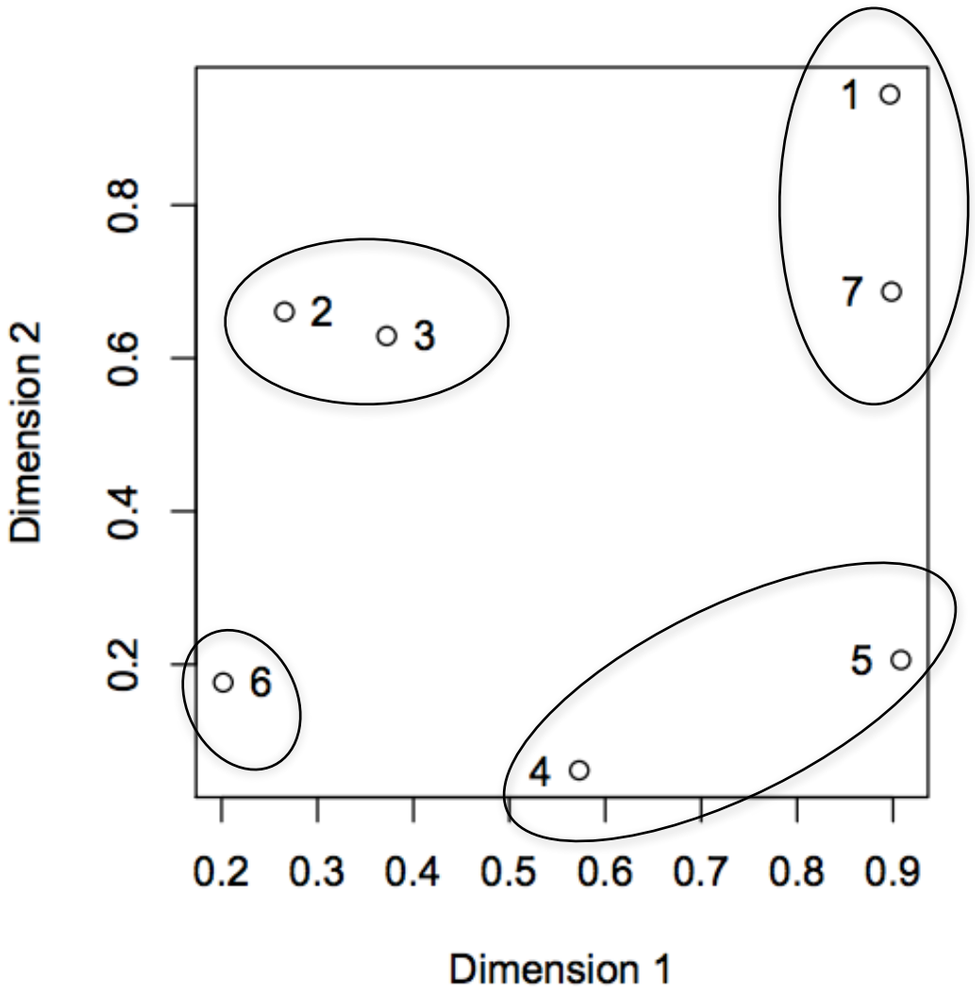




A horizontal cut in the dendogram results in a clustering

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Distance between clusters



Step 1: {1} {2} {3} {4} {5} {6} {7}

Step 2: {1} {2, 3} {4} {5} {6} {7}

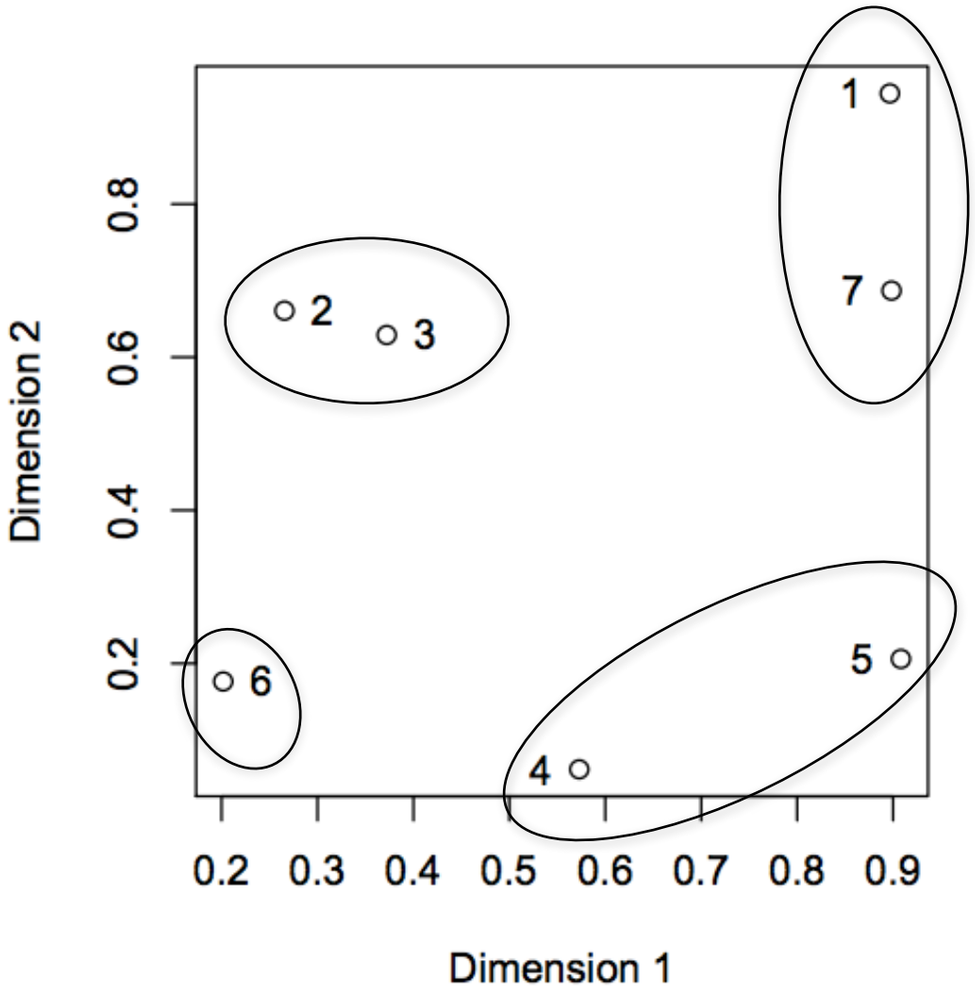
Step 3: {1, 7} {2, 3} {4} {5} {6}

Step 4: {1, 7} {2, 3} {4, 5} {6}

What are the next two closest clusters?

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Single Linkage



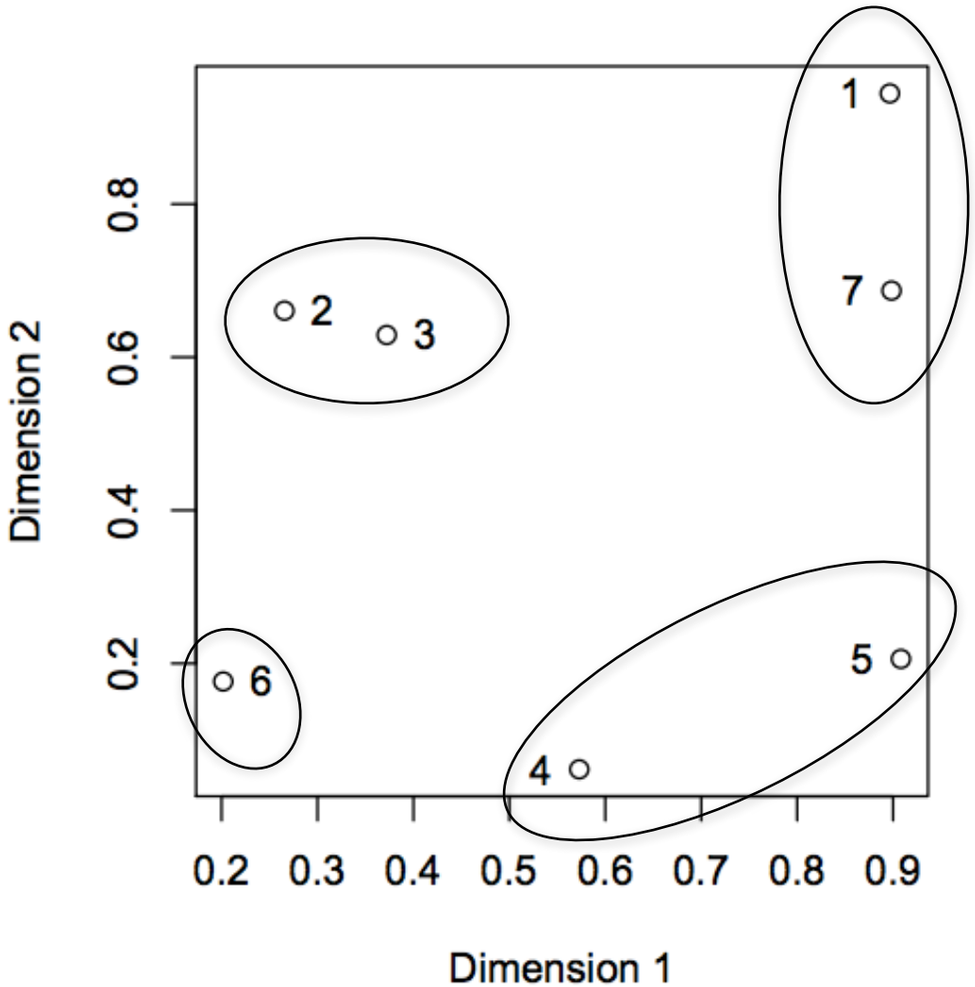
$$d_{single}(C_1, C_2) = \min_{x \in C_1, y \in C_2} d(x, y)$$

Distance between two clusters is the distance between the two ***closest*** *points in the clusters.*

{6} is closer to {4,5} than {2,3} according to single linkage

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Complete Linkage



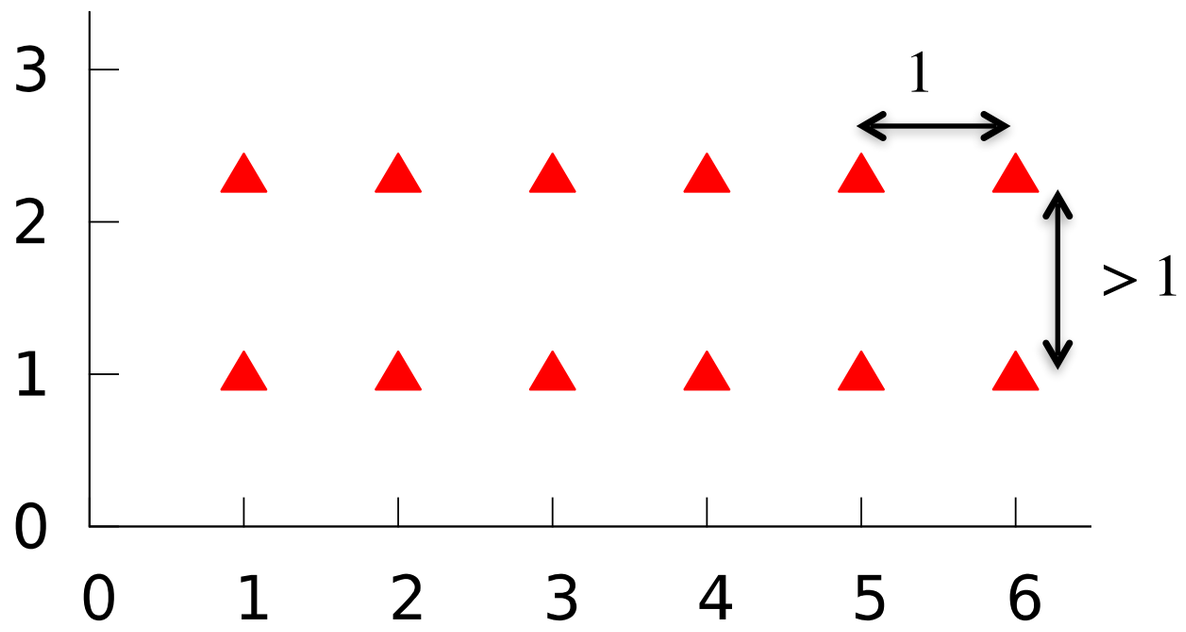
$$d_{complete}(C_1, C_2) = \max_{x \in C_1, y \in C_2} d(x, y)$$

Distance between two clusters is the distance between the two ***farthest*** *points in the clusters.*

{6} is closer to {2,3} than {4,5} according to complete linkage

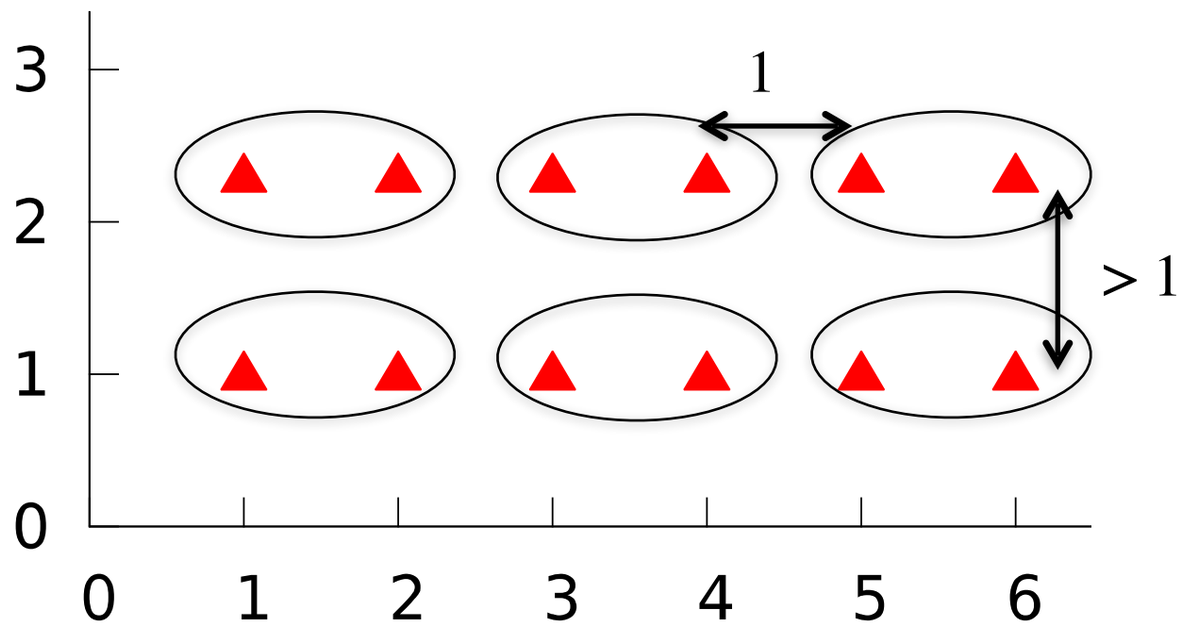
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Single vs Complete Linkage



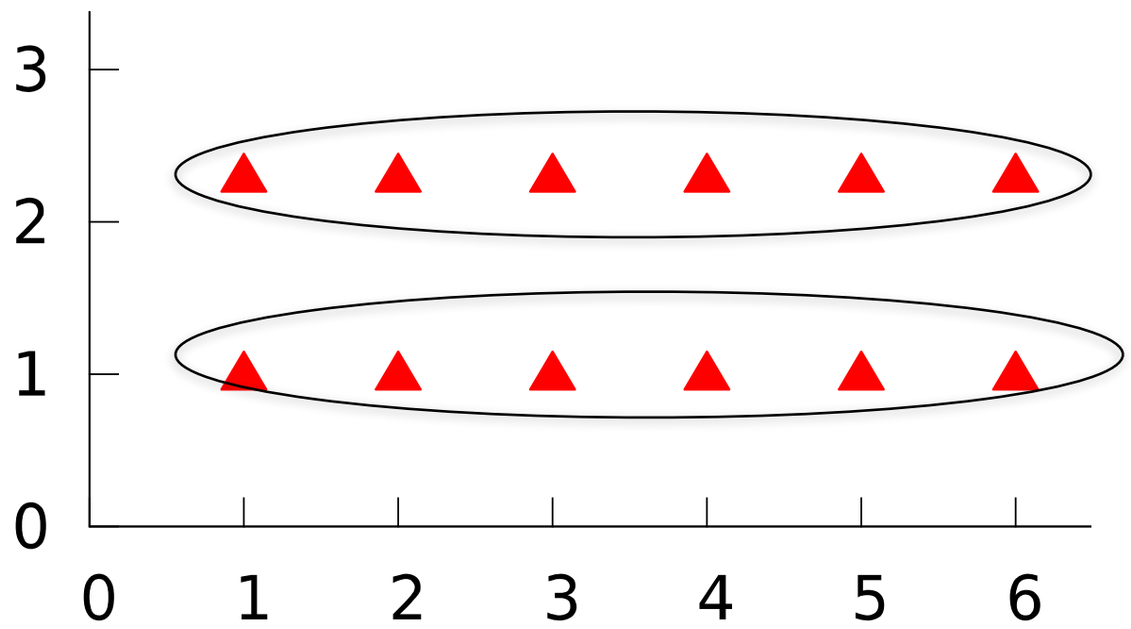
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Single Linkage



Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Single Linkage



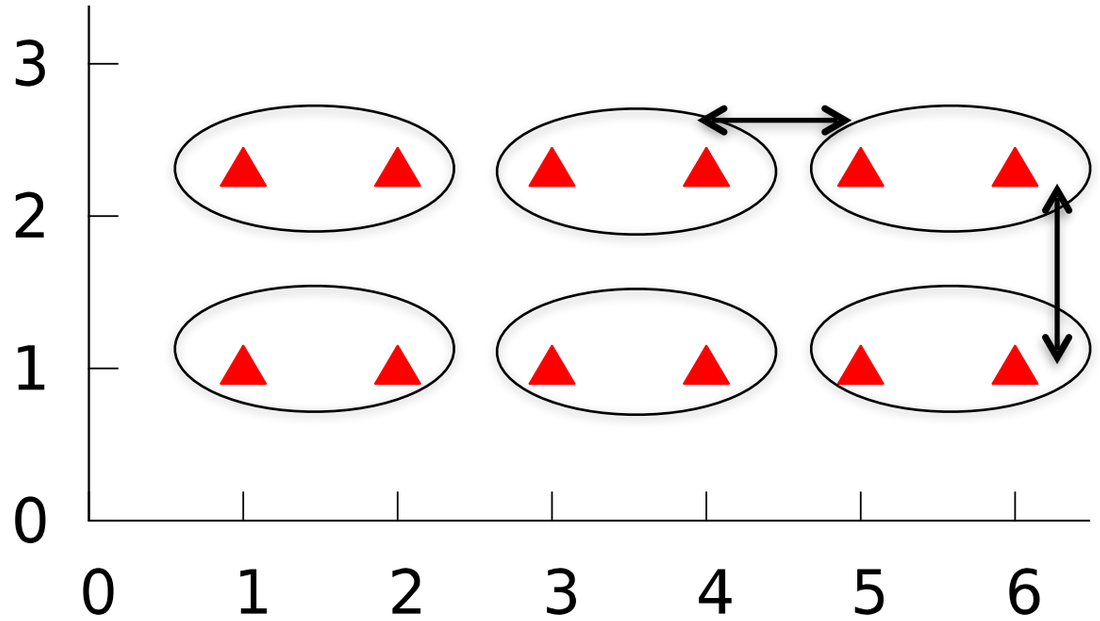
Chaining: *Single linkage can result in clusters that are spread out and not compact*

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Question

Instructions: ~1 minute to think/answer on your own; then discuss with neighbors; then I will call on one of you

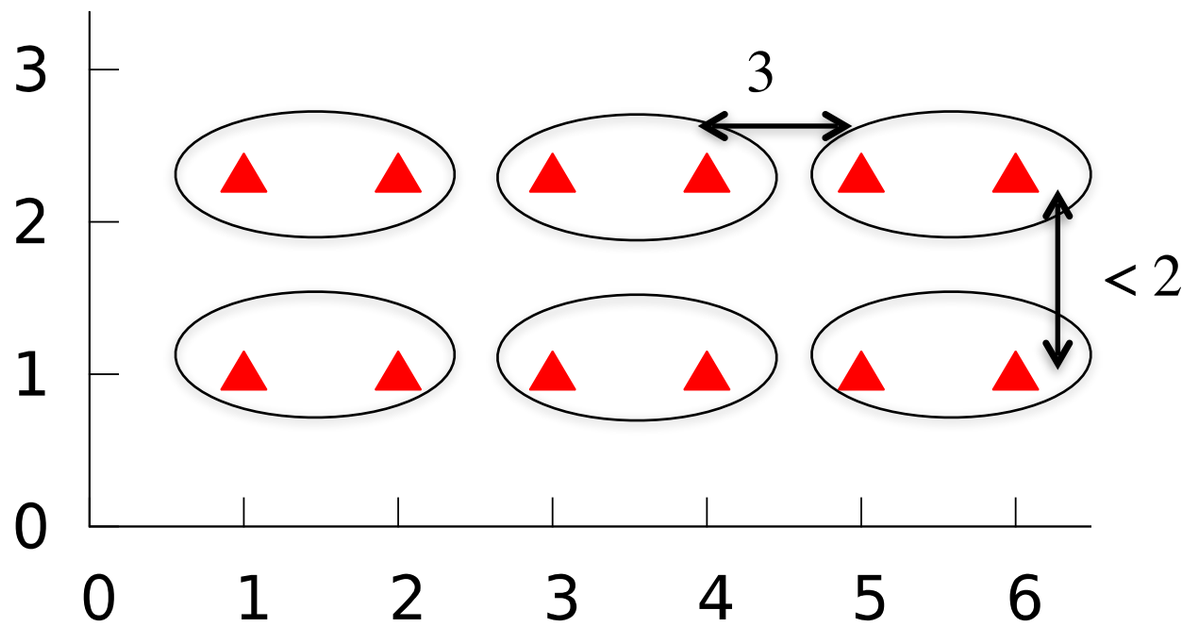
What would complete linkage do next?
Hint: compare the two cluster distances, indicated by arrows.



$$d_{complete}(C_1, C_2) = \max_{x \in C_1, y \in C_2} d(x, y)$$

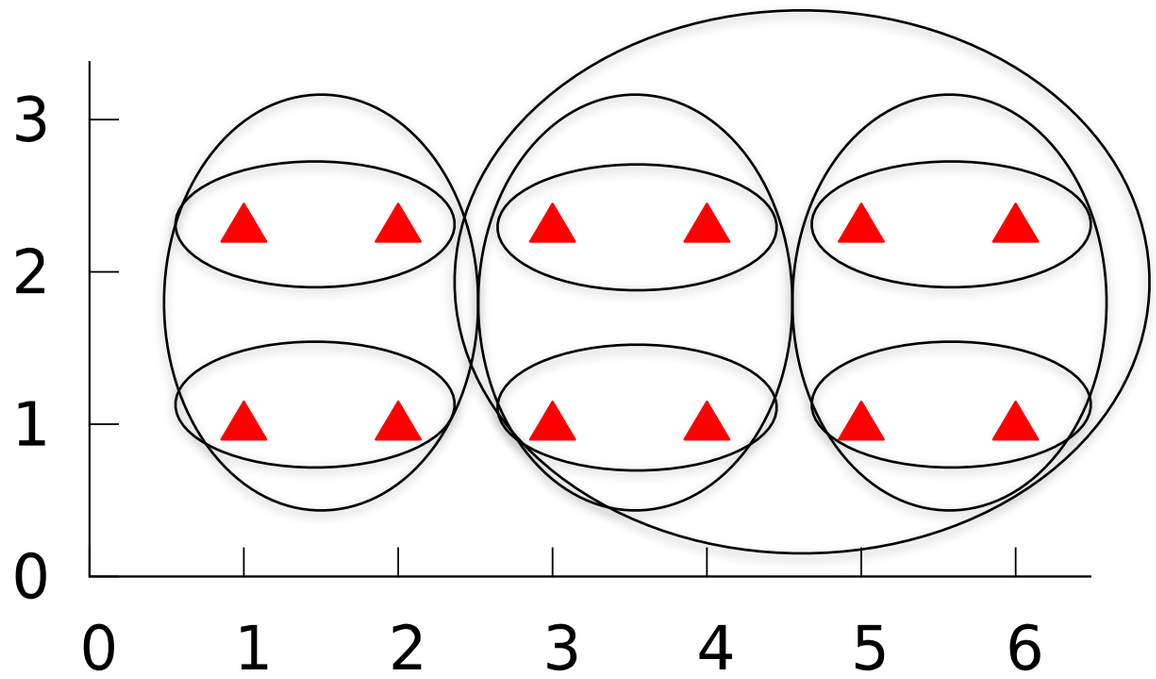
http://mdsr-book.github.io/exercises.html

Complete Linkage



Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

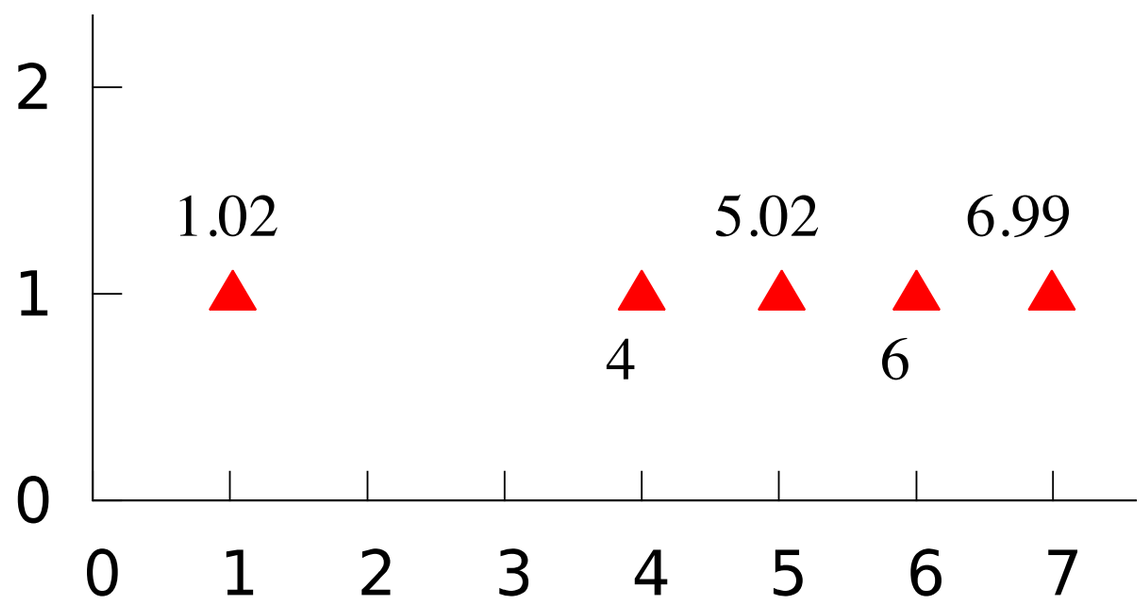
Complete Linkage



Complete linkage returns more compact clusters in this case.

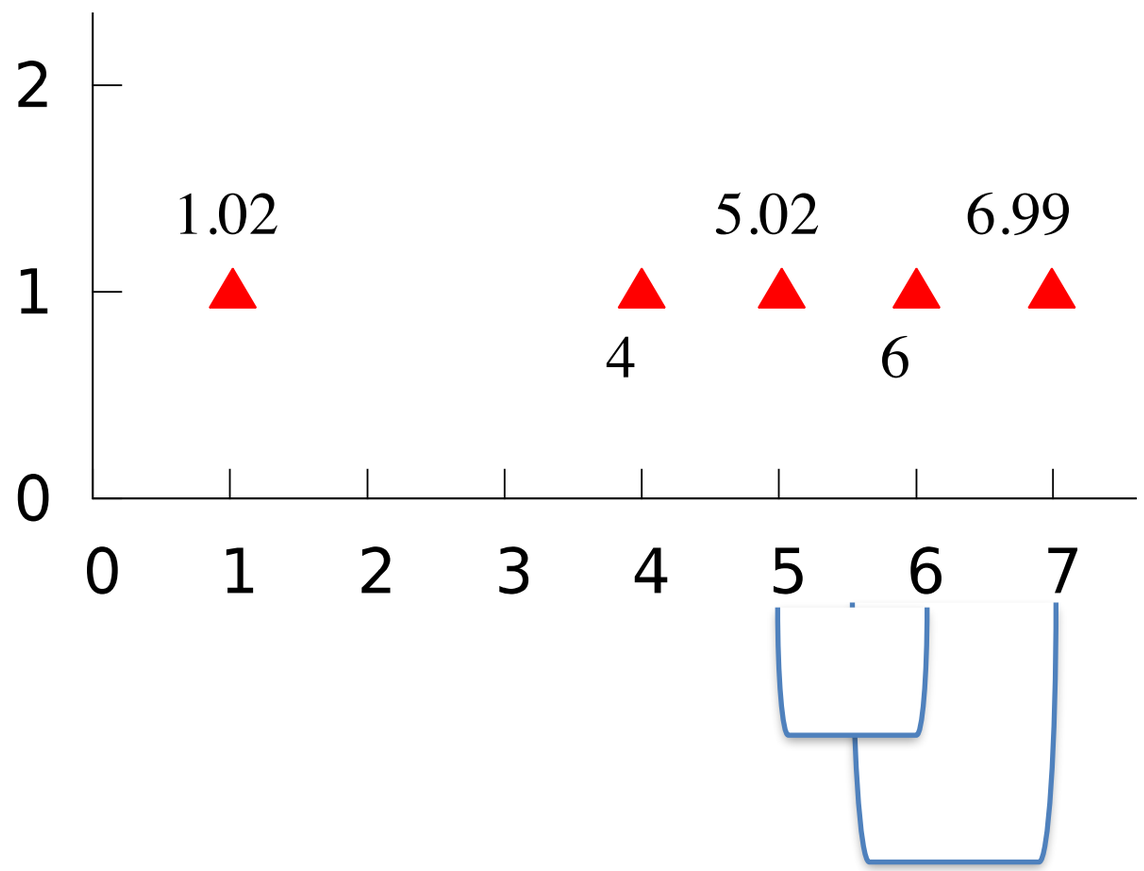
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Single vs Complete Linkage



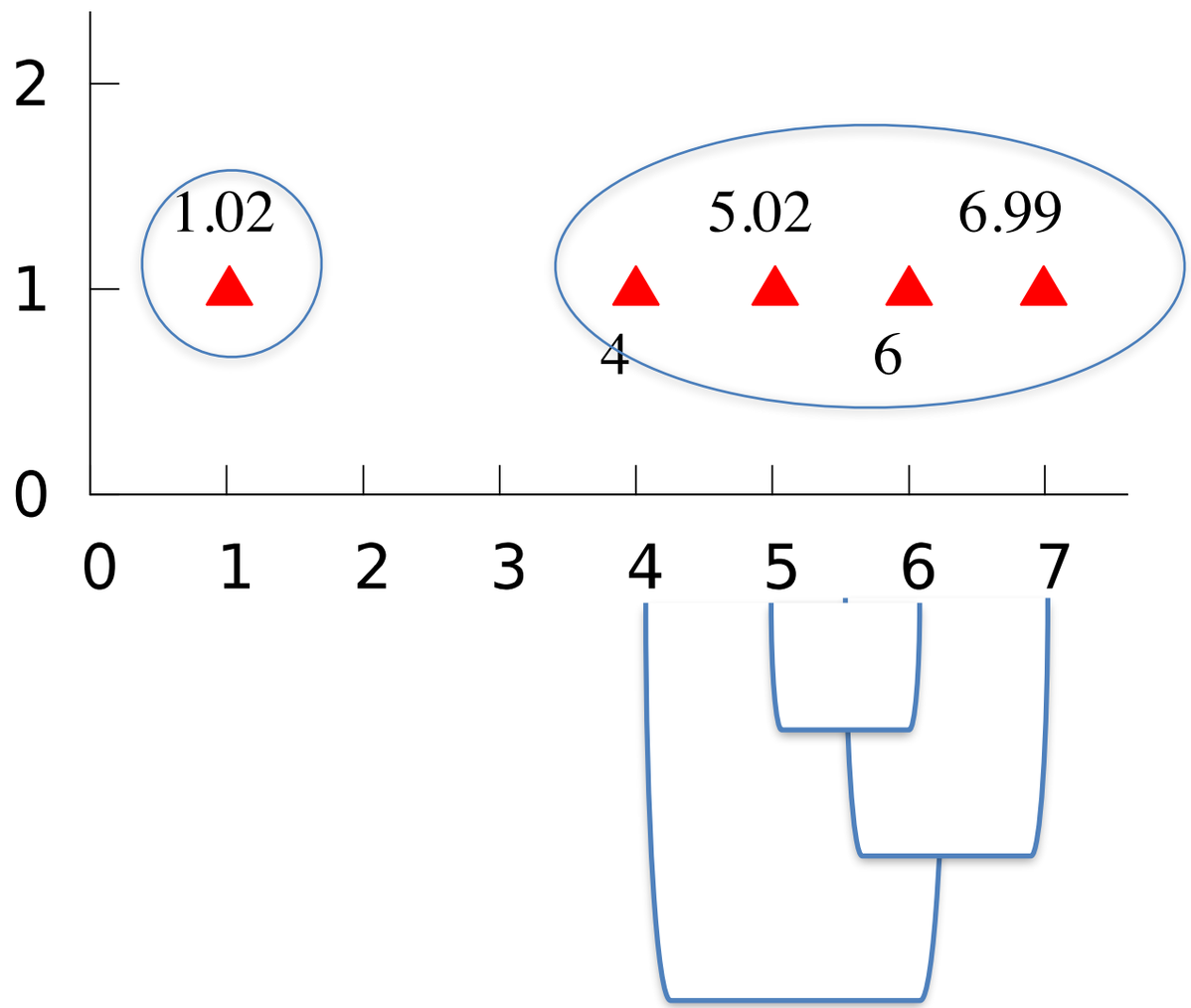
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

In both cases …



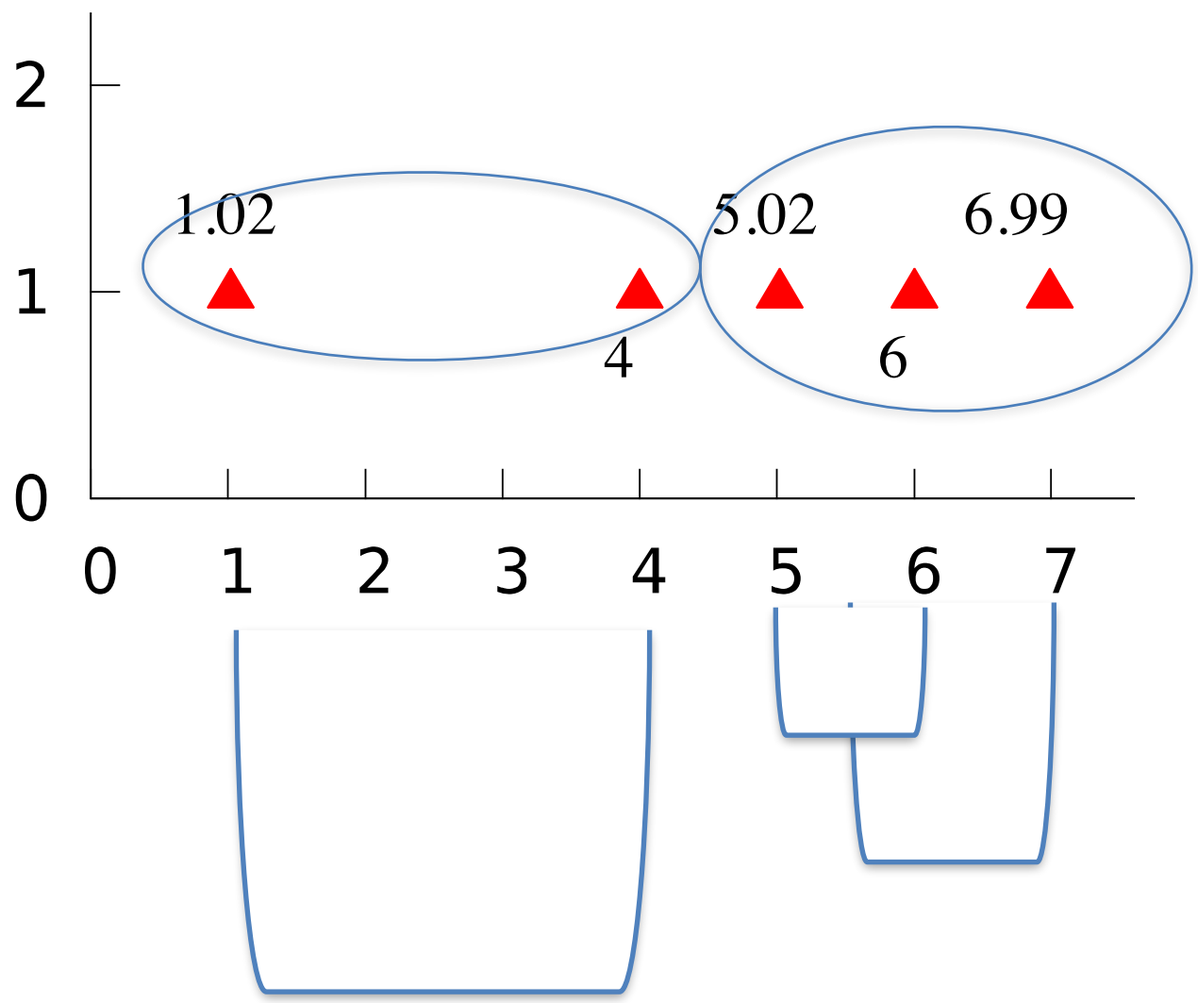
Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Single Linkage



Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

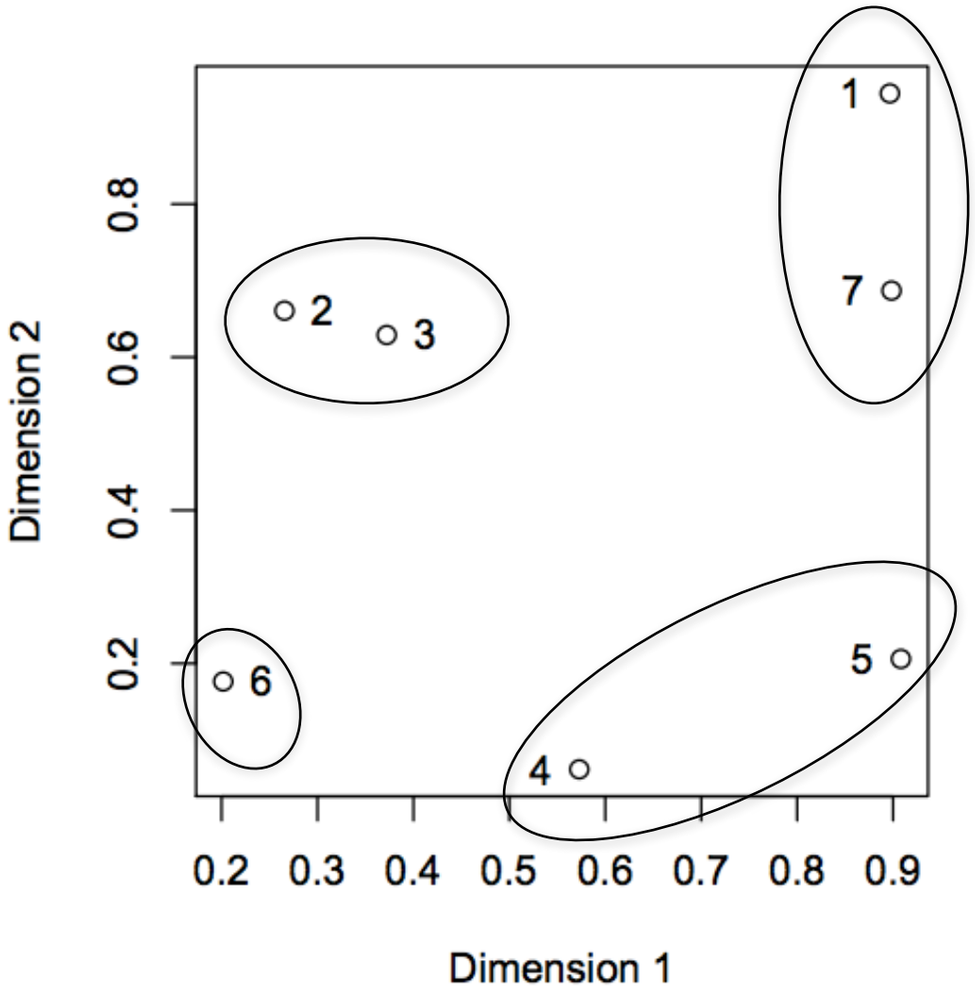
Complete Linkage



*Complete Linkage is **sensitive to outliers.***

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Average Linkage



$$d_{avg}(C_1, C_2) = \frac{\sum_{x \epsilon C_1, y \epsilon C_2} d(x, y)}{|C_1| \cdot |C_2|}$$

*Distance between two clusters is the **average** distance between every pair of points in the clusters.*

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University

Hierarchical Clustering summary

- Create a family of hierarchical clusterings
  - Visualized using a dendrograms
  - Users can choose number of clusters *after* clustering is done.
- Can use any distance function
- Different choices for measuring distance between clusters.

Credit: slide adapted with permission from Ashwin Machanavajjhalla and Jun Yang, Duke University